MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANALOG TO TEST CHART No. 2)

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

AVIS À L'ÉTUDIANT

NOTICE TO STUDENT

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Prediction of the Effects of Spanwise Mixing for Turbines with Damaged Blades

by

Jeffrey R. Verhiel

B.A.Sc. (Mechanical Engineering)

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements
for the degree of

Master of Engineering

in

Aerospace Engineering

**Ottawa-Carleton Institute for Mechanical
and Aerospace Engineering**

**Department of Mechanical and Aerospace Engineering
Carleton University
Ottawa, Ontario
January, 1994**

© 1994

Jeffrey R. Verhiel

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Exemplaire - Exemplaire

Exemplaire - Exemplaire

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-92999-5

Canada

Name JEFFREY RICHARD JEFFREY

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

ENGINEERING - AEROSPACE

SUBJECT TERM

0538

SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 413
Speech Communication 0459
Theater 0465

Psychology 0525
Reading 0535
Religious Sciences 0527
Secondary 0714
Social Sciences 0533
Sociology of 0534
Special 0340
Teacher Training 0529
Technology 0530
Tests and Measurements 0710
Vocational 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language 0679
General 0289
Ancient 0290
Linguistics 0291
Modern 0401
Literature 0294
General 0295
Classical 0297
Comparative 0298
Medieval 0316
Modern 0591
African 0305
American 0352
Asian 0355
Canadian (English) 0593
Canadian (French) 0311
English 0312
Germanic 0315
Latin American 0313
Middle Eastern 0314
Romance 0501
Slavic and East European 0503

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy 0422
Religion 0318
General 0321
Biblical Studies 0319
Clergy 0320
History of 0322
Philosophy of 0469
Theology 0323

SOCIAL SCIENCES

American Studies 0323
Anthropology 0324
Archaeology 0326
Cultural 0327
Physical 0310
Business Administration 0272
General 0770
Accounting 0454
Banking 0338
Management 0385
Marketing 0501
Canadian Studies 0503
Economics 0505
General 0508
Agricultural 0509
Commerce Business 0510
Finance 0511
History 0358
Labor 0366
Theory 0351
Folklore 0578
Geography 0351
Gerontology 0351
History 0578
General 0578

Ancient 0579
Medieval 0581
Modern 0582
Black 0328
African 0331
Asia, Australia and Oceania 0332
Canadian 0334
European 0335
Latin American 0336
Middle Eastern 0333
United States 0337
History of Science 0585
Law 0398
Political Science 0615
General 0616
International Law and Relations 0617
Public Administration 0814
Recreation 0452
Social Work 0626
Sociology 0627
General 0938
Criminology and Penology 0631
Demography 0628
Ethnic and Racial Studies 0629
Individual and Family Studies 0630
Industrial and Labor Relations 0700
Public and Social Welfare 0344
Social Structure and Development 0709
Theory and Methods 0999
Transportation 0453
Urban and Regional Planning 0453
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture 0473
General 0285
Agronomy 0475
Animal Culture and Nutrition 0476
Animal Pathology 0359
Food Science and Technology 0478
Forestry and Wildlife 0479
Plant Culture 0480
Plant Pathology 0817
Plant Physiology 0777
Range Management 0746
Wood Technology 0306

Biology 0306
General 0287
Anatomy 0308
Biostatistics 0309
Botany 0379
Cell 0329
Ecology 0353
Entomology 0369
Genetics 0793
Limnology 0410
Microbiology 0307
Molecular 0311
Neuroscience 0416
Oceanography 0433
Physiology 0821
Radiation 0778
Veterinary Science 0472
Zoology 0786
Biophysics 0760
General 0786
Medical 0760

EARTH SCIENCES

Biogeochemistry 0425
Geochemistry 0696

Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences 0768
Health Sciences 0566
General 0300
Audiology 0992
Chemotherapy 0567
Dentistry 0350
Education 0769
Hospital Management 0758
Human Development 0982
Immunology 0564
Medicine and Surgery 0347
Mental Health 0569
Nursing 0570
Nutrition 0380
Obstetrics and Gynecology 0354
Occupational Health and Therapy 0381
Ophthalmology 0571
Pathology 0411
Pharmacology 0572
Pharmacy 0382
Physical Therapy 0573
Public Health 0574
Radiology 0575
Recreation 0575

Speech Pathology 0460
Toxicology 0383
Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences 0485
Chemistry 0749
General 0486
Agricultural 0487
Analytical 0488
Biochemistry 0738
Inorganic 0490
Nuclear 0491
Organic 0494
Pharmaceutical 0495
Physical 0754
Polymer 0405
Radiation 0605
Mathematics 0986
Physics 0606
General 0608
Acoustics 0748
Astronomy and Astrophysics 0607
Atmospheric Science 0798
Atomic 0759
Electronics and Electricity 0609
Elementary Particles and High Energy 0610
Fluid and Plasma 0752
Molecular 0756
Nuclear 0611
Optics 0463
Radiation 0346
Solid State 0984
Statistics 0984

Applied Sciences

Applied Mechanics 0346
Computer Science 0984

Engineering 0537
General 0538
Aerospace 0539
Agricultural 0540
Automotive 0541
Biomedical 0542
Chemical 0543
Civil 0544
Electronics and Electrical 0348
Heat and Thermodynamics 0545
Hydraulic 0546
Industrial 0547
Marine 0794
Materials Science 0548
Mechanical 0743
Metallurgy 0551
Mining 0552
Nuclear 0549
Packaging 0765
Petroleum 0554
Sanitary and Municipal 0790
System Science 0428
Geotechnology 0796
Operations Research 0795
Plastics Technology 0994
Textile Technology 0994

PSYCHOLOGY

General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

The undersigned recommend to the Faculty of Graduate Studies
and Research acceptance of the thesis

**Prediction of the Effects of Spanwise Mixing
for Turbines with Damaged Blades**

submitted by Jeffrey R. Verhiel, B.A.Sc. (Mechanical Engineering)
in partial fulfilment of the requirements for
the degree of Master of Engineering
in Aerospace Engineering

A handwritten signature in black ink, appearing to read 'S.A. Sjolander', with a long horizontal line extending to the right.

Thesis Supervisor, S.A. Sjolander

A handwritten signature in black ink, appearing to read 'Robert Bell', with a long horizontal line extending to the right.

Chair, Department of Mechanical and Aerospace Engineering

Carleton University

to my parents

Abstract

Experiments have revealed that spanwise mixing of fluid properties can occur in axial compressors due to turbulent diffusion and secondary flow. It is assumed here that spanwise convection due to secondary flow will be greater in turbines due to the higher flow turning compared to compressors. It has also been shown that including spanwise mixing in turbomachinery flow computations of the streamline curvature type greatly improves the ability of such codes to accurately predict radial flow variations throughout the machine. Thus, a predictive method for damaged turbines must model spanwise mixing in order to diffuse radially, the concentrated losses expected in such machines. The present study was conducted to investigate, computationally, the effects of spanwise mixing in turbine duct flows.

A method of including spanwise mixing in a streamline curvature program for steady compressible duct flow is introduced. The spanwise mixing model includes the effects of both turbulent diffusion and secondary flow. A separate inviscid, incompressible, secondary flow computation is also developed in order to predict secondary flow velocities at the exit of a stationary blade passage. These results are then used to model spanwise convection in the duct flow program.

The secondary flow code is tested using published experimental results from two linear turbine cascades. The code is shown to predict the secondary flowfield at the blade passage exit plane with reasonable accuracy. The duct flow program, including both types of mixing, is tested using published experimental results from an annular diffuser downstream of a single stage turbine. The inclusion of spanwise mixing is shown to improve the agreement between the computed diffuser exit flow and that measured. Any discrepancies between computed and measured results for both codes are, in general, consistent with the simplifications made in developing the flow models.

Acknowledgements

I wish to express my profound gratitude to my thesis supervisor, Dr. Steen A. Sjolander, for his invaluable advice, encouragement, and direction during the course of this research. The generous funding which he imparted to me is greatly appreciated. Financial support for this study provided by GasTOPS Ltd., Ottawa, Ontario, Canada and the Government of Ontario (under the University Research Incentives Fund) is gratefully acknowledged.

I would also like to extend many thanks to the staff of the Aeronautics and Astronautics Library of the National Research Council of Canada for assisting me in locating those hard-to-find references which I always seemed to encounter.

The funding of the Natural Sciences and Engineering Research Council of Canada through the Postgraduate Scholarship Program is duly acknowledged.

Finally, I would like to extend my sincere thanks to Karen K. Campbell for her patience and understanding in dealing with me during the more stressful times and Dale W. Ladoon for his constant interest in my work (and the beers!).

Table of Contents

	page
Abstract	i
Acknowledgements	ii
Table of Contents	iii
List of Figures	vii
List of Tables	xiii
Nomenclature	xiv
1 INTRODUCTION	1
2 LITERATURE REVIEW	5
2.1 Secondary Flow in Axial Turbomachinery	5
2.1.1 Experimental Investigations	6
2.1.2 Predictive Methods for Cascades	8
2.2 Turbomachinery Computation Methods	11
2.3 Spanwise Mixing in Axial Turbomachinery	17
2.3.1 Experimental Investigations	17
2.3.2 Predictive Methods	25
2.3.3 Mixing Coefficients	32
3 SECONDARY FLOW COMPUTATION: THEORY	42
3.1 Flow Model	42
3.2 Secondary Stream Function Equation	46
3.3 Distributed Secondary Vorticity	51
3.3.1 Inlet Vorticity Components	55
3.4 Convection of Streamlines	57
3.5 Solution Procedure	60

4	SECONDARY FLOW COMPUTATION: NUMERICAL PROCEDURES	66
4.1	Solution of the Secondary Stream Function Equation	66
4.1.1	Estimating the Optimum Relaxation Factor	68
4.1.2	Boundary and Initial Conditions	69
4.2	Bivariate Interpolation on a Rectangular Grid	70
4.3	Bivariate Interpolation on an Unstructured Grid	72
4.3.1	Triangulation Methods	73
4.3.2	Inverse-Distance Methods	75
4.3.3	Kriging	78
4.3.4	Present Method	78
5	ANNULAR DUCT FLOW COMPUTATION: THEORY	82
5.1	Flow Model	82
5.2	Streamline Curvature Method	85
5.2.1	Governing Equations	86
5.2.2	Solution Procedure	91
5.2.2.1	Boundary Conditions	94
5.2.3	Stability Considerations	95
5.2.4	Mach Number Limitations	96
6	SPANWISE MIXING	98
6.1	Mixing Due to Turbulent Diffusion	98
6.1.1	Turbulence Model	99
6.1.2	Mixing Equations	101
6.1.3	Method of Implementation	104
6.1.4	Boundary Conditions	104
6.2	Mixing Due to Secondary Flows	105
6.3	Mixing Coefficients	111
6.3.1	Turbulent Mixing Coefficient	112
6.3.2	Convective Mixing Coefficient	112

7	ANNULAR DUCT FLOW COMPUTATION: NUMERICAL PROCEDURES	115
7.1	Initial Conditions	115
7.2	Estimating Streamline Slope and Curvature	119
7.2.1	Rational Cubic Spline	120
7.2.2	Slope and Curvature Using Finite Differences	123
7.3	Solution of the Radial Equilibrium Equation	125
7.4	Adjustment of Streamlines	128
7.5	Solution of the Mixing Equations	130
7.5.1	Applying the Boundary Conditions	133
8	RESULTS AND DISCUSSION	134
8.1	Verification of the Secondary Flow Code (BLADE)	134
8.1.1	Convective Mixing Coefficients	154
8.1.2	Parametric Study for Program BLADE	156
8.2	Results from Annular Duct Flow Computation (DUCT)	160
8.2.1	Test-Case Description	161
8.2.2	Diffuser Inlet Conditions	165
8.2.3	Determination of Mixing Coefficients	169
8.2.4	Comparison of Diffuser Exit Conditions	171
9	CONCLUSIONS AND RECOMMENDATIONS	182
9.1	Conclusions	182
9.2	Recommendations for Further Study	183
	References	186
	Appendices	193
	A: Contributions to Experimental Research	194
	B: Method of Spanwise Grid Clustering	203
	C: Derivation of the Mixing Equations	208
	D: Numerical Approximation of Derivatives	215

E: Numerical Integration Using Non-Uniform Step-Size	219
F: Program Documentation	224
G: Program Listings	225

List of Figures

	page
Figure 2.1	
Main secondary flow features found in a turbomachinery blade passage (reproduced from De Ruyck and Hirsch, 1988).	7
Figure 2.2	
Horseshoe vortex (V1) formed near the junction of a cylinder and a flat surface (reproduced from Jabbari and Goldstein, 1993).	9
Figure 2.3	
Passage and horseshoe vortices formed by a boundary layer entering a turbine cascade (reproduced from Langston, 1980).	9
Figure 2.4	
A meridional section of a two-stage axial-flow compressor (reproduced from Hirsch and Denton, 1981).	13
Figure 2.5	
Computational grid, for streamline curvature method, formed by the intersection of computational planes and the projection of streamlines (reproduced from Hirsch and Denton, 1981).	13
Figure 2.6	
Grid nodes for finite-difference computation (reproduced from Hirsch and Denton, 1981).	15
Figure 2.7	
Flow passage of two-stage compressor divided into finite elements (dashed lines) (reproduced from Hirsch and Denton, 1981).	15
Figure 2.8	
Blade-to-blade (S1) surface and hub-to-casing (S2) surface for a blade passage (reproduced from Hirsch and Denton, 1981).	16
Figure 2.9	
Blade passage subdivided into finite volume elements (reproduced from Hirsch and Denton, 1981).	16
Figure 2.10.a	
Ethylene concentration contours showing extensive mixing by turbulent diffusion in the later stages of a compressor (reproduced from Gallimore and Cumpsty, 1986).	19
Figure 2.10.b	
Ethylene concentration contours measured four stages downstream of injection point (reproduced from Gallimore and Cumpsty, 1986).	19
Figure 2.11	
Computed secondary flow velocity vectors, at 95% chord of stator passage, showing significant radial velocities (reproduced from Leylek and Wisler, 1991).	22

Figure 2.12	Tracer-gas concentration contours in the (a) first stage; (b) third stage (with injection at endwalls); and (c) third stage (with injection away from endwalls) of an axial turbine (reproduced from Lewis, 1993(a)).	24
Figure 2.13	(a) Computed radial flow angle aft of rotor. (b) Stagnation pressure profiles at exit of three-stage compressor. (c) Stagnation temperature profiles in the rear stages of a nine-stage compressor (reproduced from Adkins and Smith, 1982).	27
Figure 2.14	Total temperature profiles from three-stage axial compressors with (a) low aspect ratio and (b) higher aspect ratio (reproduced from Gallimore, 1986).	30
Figure 2.15	Normalized mixing coefficients, aft of the third stage of a four-stage compressor, calculated using three different models (reproduced from Wisler et al., 1987).	35
Figure 2.16	Dimensionless mixing coefficients calculated at various points in three compressors (reproduced from Li and Cumpsty, 1991(b)).	37
Figure 2.17	Dimensionless mixing coefficients in a single-stage compressor with different tip clearances (note: z is axial direction) (reproduced from Goto, 1991).	37
Figure 2.18	Normalized mixing coefficients from a transonic compressor stage compared to those calculated in previous studies (reproduced from Kotidis and Epstein, 1991).	38
Figure 2.19	Axial variation of midspan effective mixing coefficients (dimensionless) for two turbines (reproduced from Lewis, 1993(b)).	41
Figure 3.1	Classical secondary flow model of Hawthorne (adapted from Sieverding, 1985).	43
Figure 3.2	Distortion of a Bernoulli surface, by convective secondary flow, as it passes through a blade passage (adapted from Bardon et al., 1975).	45
Figure 3.3	Blade passage profile divided into computational regions and illustrating local primary streamline coordinate system.	47
Figure 3.4	Intrinsic coordinate system (adapted from Lakshminarayana and Horlock, 1972).	52
Figure 3.5	Streamline convection between intermediate planes (ie. across a sub-region).	58

Figure 3.6	Blade passage terminology for secondary flow computation.	61
Figure 3.7	Iterating to find the streamline intersection point on intermediate plane 2 (k is the iteration number).	63
Figure 4.1	Grid points employed in the Point Successive Over Relaxation iteration formula.	67
Figure 4.2	Grid points used in bilinear interpolation of $P(y,z)$.	71
Figure 4.3.a	Thiessen polygon (shaded) around point A . All locations within polygon are closer to A than to any other point (reproduced from Davis, 1986).	74
Figure 4.3.b	Delaunay triangle network around point A . All connected points are Thiessen neighbours of A (reproduced from Davis, 1986).	74
Figure 4.4	(a) Search technique that locates n nearest neighbours around point (y,z) . (b) Quadrant search pattern used to insure equitable distribution of neighbouring points. (c) Octant search pattern around point (y,z) (adapted from Davis, 1986).	77
Figure 4.5	Closest point in each quadrant surrounding point to be interpolated; $P(y,z)$.	80
Figure 5.1	The meridional plane for a hypothetical diverging annulus.	84
Figure 5.2	Notation for meridional plane.	87
Figure 5.3	Physical and computational domains for a diverging annulus.	92
Figure 6.1	Idealized flow channel downstream of blade passage.	106
Figure 7.1	Node indices for computational grid of a hypothetical annulus.	116
Figure 7.2	Damping effects of the rational cubic spline (adapted from Spaeth, 1974).	121
Figure 8.1	Blade profile and nomenclature for nozzle cascade of Marchal and Sieverding (1977).	138
Figure 8.2	Measured inlet boundary layer profiles of Marchal and Sieverding (1977).	139
Figure 8.3.a	Measured secondary flow velocity vectors near trailing edge plane of Marchal and Sieverding cascade (reproduced from Marchal and Sieverding, 1977).	141

Figure 8.3.b	Computed secondary flow velocity vectors at the blade passage exit plane for the cascade of Marchal and Sieverding ($V_i \approx 12.2$ m/s).	141
Figure 8.4.a	Comparison of exit flow angle deviation for Marchal and Sieverding cascade (thin inlet boundary layer).	143
Figure 8.4.b	Comparison of exit flow angle deviation for Marchal and Sieverding cascade (thick inlet boundary layer).	143
Figure 8.5	Blade profile and nomenclature for rotor cascade of Walsh and Gregory-Smith (1990).	146
Figure 8.6	Inlet boundary layer profile based on power-law (as indicated by Walsh and Gregory-Smith, 1990).	147
Figure 8.7.a	Measured secondary flow velocity vectors at trailing edge plane of Walsh and Gregory-Smith cascade (reproduced from Walsh and Gregory-Smith, 1990).	149
Figure 8.7.b	Computed secondary velocity vectors at the blade passage exit plane for the cascade of Walsh and Gregory-Smith ($V_i \approx 13.4$ m/s).	149
Figure 8.8	Comparison of exit flow angle deviation for Walsh and Gregory-Smith cascade.	150
Figure 8.9	Comparison of pitchwise mass-averaged dimensionless streamwise vorticity at blade passage exit plane for Walsh and Gregory-Smith cascade.	150
Figure 8.10.a	Contours of dimensionless streamwise vorticity, ξ_r^* , near the blade passage exit plane of the Walsh and Gregory-Smith cascade (reproduced from Walsh and Gregory-Smith, 1990).	152
Figure 8.10.b	Computed contours of dimensionless streamwise vorticity, ξ_r^* , at the blade passage exit plane for the cascade of Walsh and Gregory-Smith.	152
Figure 8.11.a	Comparison of dimensionless convective mixing coefficients for the two cascade test cases (linear axes).	155
Figure 8.11.b	Comparison of dimensionless convective mixing coefficients for the two cascade test cases (semi-log axes).	155
Figure 8.12	Computation time versus number of computational regions (test case 1).	158

Figure 8.13	The effect of the number of CRs used on exit flow angle deviation (test case 1).	158
Figure 8.14	Computation time versus number of grid points on intermediate plane (test case 1).	159
Figure 8.15	The effect of grid point density on exit flow angle deviation (test case 1).	159
Figure 8.16	Cold-air turbine and diffuser test rig (reproduced from Quest, 1990).	162
Figure 8.17	Diffuser geometry and measuring planes (reproduced from Quest, 1990).	162
Figure 8.18	Total pressure coefficient at diffuser inlet.	167
Figure 8.19	Static pressure coefficient at diffuser inlet.	167
Figure 8.20	Axial velocity coefficient at diffuser inlet.	168
Figure 8.21	Swirl angle at diffuser inlet.	168
Figure 8.22	Swirl angle at diffuser inlet measured by Quest (1990) compared with that calculated by best data fit using the secondary flow computation of the present study.	170
Figure 8.23	Dimensionless convective mixing coefficient measured by Lewis (1993(a)) compared with that calculated for the turbine of Quest (1990).	170
Figure 8.24	Calculated total pressure coefficient at diffuser exit (turbulent mixing only, $\theta=0^\circ$).	172
Figure 8.25	Calculated axial velocity coefficient at diffuser exit (turbulent mixing only, $\theta=0^\circ$).	172
Figure 8.26	Calculated swirl angle at diffuser exit (turbulent mixing only, $\theta=0^\circ$).	173
Figure 8.27	Calculated total pressure coefficient at diffuser exit (convective mixing only, $\theta=0^\circ$, $d=0.75$).	173
Figure 8.28	Calculated axial velocity coefficient at diffuser exit (convective mixing only, $\theta=0^\circ$, $d=0.75$).	175
Figure 8.29	Calculated swirl angle at diffuser exit (convective mixing only, $\theta=0^\circ$, $d=0.75$).	175

Figure 8.30	Calculated static pressure coefficient at diffuser exit (turbulent mixing only, $\theta=7^\circ$).	177
Figure 8.31	Calculated total pressure coefficient at diffuser exit (turbulent mixing only, $\theta=7^\circ$).	177
Figure 8.32	Calculated axial velocity coefficient at diffuser exit (turbulent mixing only, $\theta=7^\circ$).	178
Figure 8.33	Calculated swirl angle at diffuser exit (turbulent mixing only, $\theta=7^\circ$).	178
Figure 8.34	Calculated static pressure coefficient at diffuser exit (convective mixing only, $\theta=7^\circ$, $d=0.75$).	180
Figure 8.35	Calculated total pressure coefficient at diffuser exit (convective mixing only, $\theta=7^\circ$, $d=0.75$).	180
Figure 8.36	Calculated axial velocity coefficient at diffuser exit (convective mixing only, $\theta=7^\circ$, $d=0.75$).	181
Figure 8.37	Calculated swirl angle at diffuser exit (convective mixing only, $\theta=7^\circ$, $d=0.75$).	181
Figure A.1	Circuit diagram for Scanivalve interface.	195
Figure B.1	Notation for spanwise grid clustering.	204
Figure B.2	Examples of spanwise grid clustering for $n=11$.	207
Figure D.1	Using second-order polynomial approximations to develop: a) central-; b) forward-; and c) backward-difference expressions.	216
Figure E.1	Area in two strips of unequal width under a quadratic.	220
Figure E.2	Area under a curve divided into an even number of strips of unequal width.	222

List of Tables

	page
Table 2.1	Dimensionless mixing coefficients calculated by Gallimore and Cumpsty (1986). 35
Table 2.2	Dimensionless diffusive and secondary flow mixing coefficients for turbines investigated by Lewis, 1993(a). 41
Table 8.1	Turbine nozzle cascade geometry of Marchal and Sieverding (1977). 135
Table 8.2	Inlet boundary layer and test conditions of Marchal and Sieverding (1977). 137
Table 8.3	Cascade geometry of Walsh and Gregory-Smith (1990). 145
Table 8.4	Inlet boundary layer data of Walsh and Gregory-Smith (1990). 145
Table 8.5	Turbine-stage operating conditions for test case of Quest (1990). 164

English Symbols

a	blade pitch; parameter, Eqn. (4.2)
A	grid aspect ratio Eqn. (5.9); parameter Eqn. (2.7)
A_i	coefficient in spline curve expression, Eqn. (7.1)
b	blockage factor Eqn. (5.5); binormal coordinate (Sect. 3.3)
B_i	coefficient in spline curve expression, Eqn. (7.1)
c	blade chord
c_p, c_u	spanwise grid spacing ratios for lower and upper halves of grid respectively, c
c_p	specific heat at constant pressure
c_x	blade axial chord
C_i	coefficient in spline curve expression, Eqn. (7.1)
d	decay power-law exponent, Eqn. (6.15); power-law exponent, Eqn. (8.1)
D_i	coefficient in spline curve expression, Eqn. (7.1); distance in weighting function (Sect. 4.3.2)
E	Reynolds stress
e	internal energy; exponent in weighting function (Sect. 4.3.2)
f	general scalar function; spline function
F	body force
g	general vector function (Appendix C)
g_r, g_θ, g_z	components of vector function g (Appendix C)
h	enthalpy
h_o	stagnation (total) enthalpy
i, j, k	unit vectors in Cartesian coordinates
k	thermal conductivity; parameter Eqn. (5.9); iteration number
k_1, k_2	Runge-Kutta coefficients (Sects. 7.3 and 7.5)

l	blade span
L_s	stage length
m	coordinate in the meridional direction (Fig. 5.2)
$\dot{m}$	mass flow rate
M	Mach number
n	normal coordinate; number of points
o	blade passage throat dimension (Fig. 8.1)
p	pressure
p_o	stagnation (total) pressure
P	general fluid property
q	quasi-orthogonal direction (Fig. 5.2); heat transfer rate per unit area (Sect. 6.1.1); damping coefficient for rational cubic spline (Sect. 7.2.1)
Q	general fluid property, Eqn. (2.5)
r, θ, x	cylindrical polar coordinates in the radial, tangential and axial directions respectively
r'	first derivative of r with respect to x
r''	second derivative of r with respect to x
R_c	meridional radius of curvature (Fig. 5.2); principle radius of curvature (Fig. 3.4)
R	gas constant
s	streamwise coordinate (Figs. 3.3 and 3.4)
S	entropy
t	time; ratio of differences in x -coordinates, Eqn. (7.1); blade thickness, Eqn. (2.7)
T	temperature
T_o	stagnation (total) temperature
u, v, w	velocity components in the x , y , and z directions respectively
$\mathbf{u}, \mathbf{v}, \mathbf{w}$	orthogonal unit vectors in the r , θ , and x -directions respectively (Appendix C)
u'	velocity component in the primary flow direction (Sect. 6.2); turbulent velocity fluctuation in the x -direction (Sect. 6.1.1)
U	free-stream fluid speed
U_m	mean blade speed
v'	secondary velocity component in the y' -direction (Chap. 3); turbulent velocity fluctuation in the y -direction (Chap. 6)
v_m	meridional velocity component

v_r, v_θ, v_z	velocity components in the r , θ , and z -directions respectively
v_q	velocity component in the q -direction
v_n'	secondary velocity component normal to primary and spanwise directions
V	fluid velocity
V	fluid speed
w'	turbulent velocity fluctuation in the z -direction
w_i	weighting function (Sect. 4.3.2)
x, y, z	Cartesian coordinates in axial, tangential, and spanwise directions respectively
x'	coordinate in primary flow direction
y'	coordinate in direction normal to both the x' and z directions

Greek Symbols

α	primary flow angle (Fig. 3.3)
α'	blade metal angle
β	mixing coefficient, Eqns. (2.3) and (6.11); grid spacing ratio (Chap. 4)
γ	specific heat ratio, c_p/c_v ; blade stagger angle
Γ	angle between binormal and stagnation pressure gradient vectors (Fig. 3.4)
$\delta r, \delta \theta$	streamline deflection components in the radial and tangential directions respectively, Eqns. (3.19.a) and (3.19.b)
$\delta x, \delta y$	streamline deflection components in the cross-passage and spanwise directions respectively, Eqns. (3.20.a) and (3.20.b)
$\delta_{99\%}$	boundary layer thickness based on 99% free-stream fluid speed (Chap. 8)
δ^*	boundary layer displacement thickness (Chap. 8)
	$= \int_0^{\delta} \left(1 - \frac{u}{U} \right) dy$
Δx	axial distance between intermediate planes (Sect 3.4); x -direction spacing of data points for cubic spline (Sect 7.2.1)
Δy	pitchwise grid spacing (Chap. 4)
Δz	spanwise grid spacing (Chap. 4)

$\Delta\alpha$	exit flow angle deviation (Chap. 8)
ϵ	mixing coefficient, Eqn. (6.13)
ζ	streamline slope angle (angle between meridional and axial directions) (Fig. 5.2)
η	isentropic efficiency
κ	quasi-orthogonal lean angle (Fig. 5.2)
θ	diffuser casing divergence angle (Fig. 8.17); boundary layer momentum thickness (Chap. 8) = $\int_0^\delta \frac{u}{U} \left(1 - \frac{u}{U}\right) dy$
λ	second coefficient of viscosity
μ	dynamic viscosity
ν	kinematic viscosity
ξ	vorticity
π	3.14159...
ρ	density
τ_x	shear stress in the x -direction
υ	specific volume
Φ	dissipation function, Eqn. (6.3)
ϕ	angle between meridional and quasi-orthogonal directions (Fig. 5.2); flow coefficient, Eqn. (2.7)
χ	substitution variable for Runge-Kutta integration, $= r v_0$ (Sect. 7.5)
ψ	secondary stream function
ω	relaxation factor (Sect 4.1); loss coefficient, Eqn. (2.7)
Ω	inlet swirl angle

Dimensionless Groups

h^*	dimensionless channel height (Sect. 8.2.1)
H	boundary layer shape factor $= \delta^*/\theta$ (Chap. 8)
p^*	static pressure coefficient (Sect. 8.2.1)
p_o^*	total pressure coefficient (Sect. 8.2.1)

Pr_t	turbulent Prandtl number, Eqn. (6.1)
Sc_t	turbulent Schmidt number, Eqn. (6.2)
V_x^*	axial velocity coefficient (Sect. 8.2.1)
ξ_s^*	dimensionless streamwise vorticity (Sect. 8.1)

Subscripts / Superscripts

a	atmospheric
b	component in the binormal direction
c	convective
cv	convective, Eqn. (2.8)
$casing$	casing surface
d	diffusive
e	empirical; blade passage exit
hub	hub surface
i	column index; blade passage inlet; data point number for cubic spline
j	row index
k	iteration number
m	component in the m -direction
n	component in the n -direction; normal
o	stagnation (total) conditions
opt	optimum value
q	component in the quasi-orthogonal direction
r, θ, x	components in the r , θ , and x directions respectively
rel	relative
s	component in the s -direction; streamwise
sec	secondary
sp	user specified
st	streamtube
t	turbulent

<i>tip</i>	blade tip
<i>wall</i>	boundary value (Sect. 6.1.4)
<i>x,y,z</i>	components in the <i>x</i> , <i>y</i> , and <i>z</i> directions respectively
1	upstream station
2	downstream station
—	time mean; arithmetic mean; area-average
— ^m	mass-average

Abbreviations

CFD	computational fluid dynamics
EHM	engine health monitoring
QO	quasi-orthogonal
REE	radial equilibrium equation

Engine health monitoring (EHM) systems are routinely used by essentially all civilian and military users of gas turbine engines. These systems inform users of gradual or abrupt deterioration of engine performance so that appropriate actions may be taken before the problem leads to severe engine damage or an unscheduled shutdown. Since the gas turbine engine has a low work ratio (engine power output/turbine power output) a small decrease in turbine efficiency can result in a significant decrease in the efficiency, and thus power output, of the engine. A common cause of reduced turbine efficiency is in-service damage to the blades. Rotor and stator blade damage can take a number of forms: blade surface roughening due to fouling, erosion or corrosion; damage to blade tips, leading or trailing edges due to foreign-object ingestion, over-temperatures, severe vibration, or lack of sufficient blade cooling. Current EHM systems cannot distinguish between these types of damage, they can only detect a problem and identify the general area of the engine where the problem exists. Thus a significant portion of maintenance time is spent determining the specific nature of the problem.

Development of an EHM system which is capable of distinguishing different types of turbine blade damage was the fundamental motivation for the present study. This will first require knowledge of the aerothermodynamic characteristics which may be specific to each type of damage. Such knowledge could be gained by performing cascade experiments to study how a particular mode of damage affects blade loss characteristics and flow-turning ability. With this information, it should be possible to predict how different types of damage will affect turbine stage performance. This was the specific motivation behind developing the predictive methods of the present study.

One commonly used computational method for turbomachinery flows, referred to as the throughflow method, assumes that the flow remains in concentric, axisymmetric streamtubes

between which no heat or momentum may be exchanged. Such a computational method was developed many years ago at Carleton University, by Ngo and Millar (1973), in order to predict turbine performance. Their code distributed all losses uniformly along the span of the blade. However, certain types of losses, such as tip leakage and secondary losses, are actually concentrated in particular regions of the annulus. Furthermore, the additional losses associated with in-service damage will often appear as small regions of very high losses. Therefore, to simulate damaged turbines the code must allow for concentrated losses. Recently, Klein (1993) altered the original code of Ngo and Millar to allow for concentrated tip leakage losses. Modelled as a parabolic distribution at the casing, the concentrated tip leakage loss was found to have a significant effect on the predicted turbine performance.

Recent experimental investigations in axial compressors have revealed that a significant amount of spanwise mixing of fluid properties can occur. Turbulent diffusion as well as secondary flow have been identified as mechanisms which contribute to the spanwise mixing process; the relative importance of each mechanism being dependent on compressor configuration and blade loading. Recent efforts to include the effects of spanwise mixing in turbomachinery throughflow computations have been presented by several authors and show that the inclusion of the mixing process can greatly improve the accuracy of results. In view of this, it was felt that a throughflow code used to predict the behaviour of damaged turbines would also have to include spanwise mixing. The effect of concentrated regions of high-loss fluid and flow misdirection due to blade damage must be accurately predicted downstream for their impact on the performance of subsequent blade rows.

The original objective of the present study was to alter the code of Ngo and Millar (1973) by including the effects of spanwise mixing. However, since the code was somewhat outdated and relatively unstructured, the author opted instead to develop a new code capable of utilizing a spanwise mixing model. This new throughflow computation was based on the same technique as that of Ngo and Millar, namely the streamline curvature method. It was hypothesised from the outset that the mixing effects of secondary flow would be much more prominent in turbines than in compressors due to the high flow turning seen in turbines. For this reason a separate secondary flow computation was developed. Results from the three-dimensional secondary flow computation would then be input to the throughflow computation to be utilized by the spanwise mixing model.

(which also included the effects of turbulent mixing). It should be noted that the throughflow code currently only has the capability of predicting turbine duct flow. In order to predict damaged turbine performance, the addition of blade rows is necessary. This future enhancement will be relatively straight-forward.

The original scope of the present work was also to have included an experimental investigation of turbine blades with tip damage. However, due to time constraints, this portion of the study was withdrawn. The author's contributions towards the intended experiments are given in Appendix A.

A review of the many recent studies concerning spanwise mixing is given in Chapter 2. Nearly all of the work reviewed involves compressor flows. Only very recently (after the present study was completed) has work concerning spanwise mixing in turbines been published. This very recent work is also discussed in the text. As well, Chapter 2 contains some brief material regarding secondary flow in turbines and computational methods for turbomachinery.

Chapters 3 and 4 describe the secondary flow computation developed in the present study. In Chapter 3 the secondary flow theory is developed, including the derivation of the fundamental equations. Several of the more important numerical considerations involved with the secondary flow code are detailed in Chapter 4.

In Chapter 5 the fundamental theory of the streamline curvature method is presented in the context of an annular duct flow computation. This chapter describes only the basic throughflow computation method in which spanwise mixing is not included.

Chapter 6 introduces the spanwise mixing model developed in the present study. The chapter describes a new way in which the mixing effects of turbulent diffusion and secondary flow can be included in the streamline curvature method. Chapter 6 thus clarifies how the theory of Chapter 3 is used to incorporate spanwise mixing, due to secondary flow, into the annular duct flow theory of Chapter 5. Following this, some of the numerical methods involved with the streamline curvature method and the spanwise mixing model are discussed in Chapter 7.

Results from the secondary flow code are presented in Chapter 8 and compared to the experimental results of two test cases. As well, some preliminary results obtained from the annular duct flow computation are presented. These computations include the spanwise mixing effects of both turbulent diffusion and secondary flows.

From the outset of the present study, it was known that the code developed herein would undergo future expansion and alteration. This is reflected in the recommendations of Chapter 9. With this in mind, all code developed by the author has been well documented, both in the source code and in the text, and designed to be well structured and modular. As well the author has compiled an extensive list of references which deal with both the theoretical and numerical aspects of the code.

The main purpose of this chapter is to familiarize the reader with previous research on spanwise mixing in axial turbomachinery. However, some fundamental concepts concerning secondary flows in turbomachines and computational methods for turbomachinery are also presented because of their relevance to the present study. The nomenclature of those authors reviewed has been altered in order to be as consistent as possible with that used in the present study. Any departures from these conventions have been indicated.

2.1 Secondary Flow in Axial Turbomachinery

This section presents some fundamental aspects of secondary flow as it applies to turbomachinery blading. Some experimental investigations which have revealed the main secondary flow features are reviewed as well as some of the efforts to predict secondary flow in cascades.

Vorticity is fundamental to the understanding of secondary flow. The vorticity of a fluid "particle" (an infinitesimally small mass of fluid) is defined as being equal to twice the angular velocity of the particle. More useful mathematically, however, is that the vorticity vector, ξ , at any point in a fluid is equal to the curl of the velocity vector, V , such that

$$\xi = \text{curl } V = \begin{vmatrix} i & j & k \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ u & v & w \end{vmatrix},$$

in Cartesian coordinates. Flows in which vorticity is present are called *rotational*.

Sir William Hawthorne has defined secondary flows as those flows associated with any component of streamwise vorticity, whereas Smith (1955) chose to define secondary flow as resulting from those velocities generated by streamwise vorticity in the real flow in excess of the streamwise vorticity associated with an axisymmetric rotational flow through closely spaced blades. More recently, Leylek and Wisler (1991) defined secondary flow as follows:

The flow deviation between the real three-dimensional flow and the "so called" primary flow (streamtubes) computed using axisymmetric, two-dimensional, blade-to-blade, etc. models is called secondary flow. These flows are associated with all components of streamwise vorticity. Strong secondary flows can be produced by turning of the mean flow, pressure gradients, boundary layer skew, blockage, tip leakage, and wake effects to name a few. Although they can also be produced by turbulence gradients such secondary flows are generally very weak. Secondary flows are deterministic and can produce bulk mixing. They can be computed by three-dimensional Navier-Stokes and Euler methods.

A summary of the main secondary flow features found in a turbomachinery blade passage is illustrated in Figure 2.1.

2.1.1 Experimental Investigations

A comprehensive review of experimental investigations of secondary flows in turbine blade passages was presented by Sieverding (1985). He focused on basic aspects of secondary flows including vortices and their effect on endwall boundary layers and the origin, growth, and distribution of losses associated with secondary flows.

The existence of a passage vortex (feature 1 in Figure 2.1) in turbomachinery blading has been known since the early 1950's and is the main generator of secondary velocities in high-turning blade rows (Langston et al., 1977, Marchal and Sieverding, 1977, Gregory-Smith and Graves, 1983). The passage vortex is the result of the redirection of the inlet shear flow (boundary layer). In higher-turning blade passages the passage vortex migrates along the endwall toward the blade suction surface and along the blade suction surface away from the endwall. The secondary velocities created by the passage vortex cause significant flow angle variation at the exit of the blade passage, resulting in flow overturning at the endwall and underturning away from the

Flow :	Origin :
1 Main passage vortex	Flow non-uniformities
2 Trailing shed and filament vortex	Spanwise changes in circulation
3 End wall boundary cross flow	Boundary layer overturning
4 End wall b. l. passage vortex	Boundary layer overturning
5 Tip clearance flow	Tip clearance pressure difference
6 Profile boundary layer radial flow	Centrifugation
7 Wake radial flow	Centrifugation
8 Radial diffusion	Turbulence

Figure 2.1 Main secondary flow features found in a turbomachinery blade passage (reproduced from De Ruyck and Hirsch ,1988).

endwall. The passage vortex will be discussed further in Chapter 3.

Two smaller vortical structures also exist in turbine blade passages. The more significant of the two is the so-called leading-edge vortex or horseshoe vortex. This vortex is formed when the endwall boundary layer separates and rolls up just in front of the blade leading-edge. The phenomenon was first observed in flows around cylinders where the vortex formed in front of the junction point of the cylinder and a flat surface (V1 in Figure 2.2). In the case of a turbine blade the boundary layer separation and roll-up is simply the result of blade leading-edge thickness. The name "horseshoe vortex" comes from the vortex shape since it wraps around the blade leading-edge with one "leg" positioned on either side of the blade. Since it has the same sense of rotation as the passage vortex, the pressure side leg of the horseshoe vortex merges with the passage vortex. The suction side leg, however, has the opposite sense of rotation and remains separate. It has been suggested that this leg of the horseshoe vortex remains in the endwall suction surface corner as illustrated in Figure 2.3. However, the results of Marchal and Sieverding (1977) show that it is sometimes located on the midspan side of the passage vortex. The third vortical structure sometimes observed in flow through turbine blade passages is called the corner vortex. It is quite small compared to the vortices already mentioned and is situated at the endwall suction surface corner. It rotates in the opposite sense to the passage vortex and has the effect of reducing the overturning at the endwall. It is thought that the corner vortex is formed by the endwall cross-flow encountering the blade suction surface.

Other secondary flow features include tip leakage flows, endwall crossflows, and rotor blade boundary layer and wake radial flow. Compared to the passage vortex, however, the effects of these mechanisms are considered to be of second order.

2.1.2 Predictive Methods for Cascades

A good review of "classical" analytical methods for the calculation of secondary flows in turbomachinery cascades was provided by Horlock and Lakshminarayana (1973). The purpose of this section is to briefly describe the development of relatively simple methods for computing secondary flow velocities in cascades. Fully three-dimensional viscous methods will not be

Figure 2.2 Horseshoe vortex (V1) formed near the junction of a cylinder and a flat surface (reproduced from Jabbari and Goldstein, 1993).

Figure 2.3 Passage and horseshoe vortices formed by a boundary layer entering a turbine cascade (reproduced from Langston, 1980).

considered.

Came and Marsh (1974) identified three components of vorticity at the exit of a many-bladed linear cascade: the distributed secondary vorticity; the trailing filament vorticity; and the trailing shed vorticity. Using two simple theories, Came and Marsh derived the same expression for the distributed secondary vorticity, ξ_{sec} . It was given by

$$\xi_{sec} = \xi_s \frac{\cos \alpha_1}{\cos \alpha_2} + \left[\frac{\xi_n}{\cos \alpha_1 \cos \alpha_2} \right] \times \left[\frac{1}{2} (\sin 2\alpha_2 - \sin 2\alpha_1) + (\alpha_2 - \alpha_1) \right], \quad (2.1)$$

where ξ_s and ξ_n are the streamwise and normal components of vorticity at the cascade inlet, respectively, and α_1 and α_2 are the absolute flow angles at inlet and exit, respectively. Once the distributed secondary vorticity is known at the cascade exit, the secondary flow velocities can be calculated using a simple Poisson-type equation which is derived from the continuity equation. The trailing filament and trailing shed vorticities were calculated using

$$\xi_{fil} = -\xi_n a \frac{(\alpha_2 - \alpha_1)}{\cos \alpha_1},$$

and

$$\xi_{shed} = -\frac{a\xi_n}{2\cos\alpha} [\sin 2\alpha_2 - \sin 2\alpha_1] - a\xi_s \cos \alpha_1,$$

respectively, where a is the blade pitch. These three components of vorticity will be discussed further in Chapter 3.

The method of Came and Marsh (1974) was only useful for predicting secondary flow in linear cascades. Glynn and Marsh (1980) extended this technique to handle flow in annular cascades by deriving a cylindrical coordinate form of the Poisson-type equation used to compute the secondary velocities. Equation (2.1) was still used to calculate the distributed secondary vorticity at the cascade exit.

Both the methods of Came and Marsh (1974) and Glynn and Marsh (1980) assumed that the convection of streamlines, by secondary flow, was negligible as the flow passed through the

blade passage. This is only true for cascades with relatively small turning angles. Bardon et al. (1975) and Glynn (1982) developed methods which took this streamline convection into account, although their techniques were for analysis of linear cascades only. The concept of streamline convection by secondary flow will be discussed further in Chapter 3.

Most recently, Gregory-Smith and Okan (1991) extended the method of Glynn (1982) to compute secondary flows in annular cascades. They also used improved computational techniques so that accurate results could be obtained for higher-turning turbine cascades.

The secondary flow computation method developed for the present study is a combination of the methods of Glynn (1982) and Gregory-Smith and Okan (1991). Full details of the present method are given in Chapters 3 and 4.

2.2 Turbomachinery Computation Methods

This section presents a brief classification of the various computation methods which have been fundamental in the design and analysis of turbomachinery for the past several decades. The methods are presented here to show where the streamline curvature method, used in the present study, fits in relative to other methods in terms of complexity and accuracy. Hirsch and Denton (1981) give a very comprehensive review of turbomachinery computation methods, tracing the development of each while citing an extensive list of references. However, their review contains little information on advanced three-dimensional methods and no information on the spanwise mixing models which have been developed recently.

Turbomachinery flows can be modelled with varying degrees of complexity depending on the simplifying assumptions used. Generally, these models fall into one of the four following categories:

Mean-line model. A line is defined at the mean radius of the flow passage under consideration. Using one-dimensional analysis, fluid properties and velocities are estimated at specified points along this mean line while losses and flow angles are determined with the use of empirical

correlations and machine geometry. At the specified points along the mean line, the computed conditions are assumed to be representative of the passage flow.

Two-dimensional model. The flow is assumed to be steady, adiabatic, and axisymmetric. The assumption of steady flow allows the motion of fluid particles to be identified by means of streamlines, streamsurfaces, and streamtubes. The adiabatic assumption means that there is no heat transferred between fluid elements or across stream surfaces. The assumption of axisymmetric flow means that the flow is assumed to have no property variations in the circumferential direction. The entire flow can thus be represented on a single hub-to-casing surface, or meridional section, of the machine. An example of a meridional section of a two-stage axial-flow compressor is illustrated in Figure 2.4. Computation methods which utilize this approximation of the flowfield are often called throughflow methods. A variety of numerical techniques can be used to solve the two-dimensional flowfield such as:

Simple radial equilibrium methods. The representative meridional flow passage is divided axially by a set of control surfaces or calculation planes which are perpendicular to the axis of the machine. Radial velocities are considered to be negligible along these calculation planes. Here the radial momentum equation is formulated such that fluid elements are in equilibrium, with accelerations due to swirl balanced by the radial pressure field. One-dimensional analysis and empirical data are used between calculation planes, at several radial positions, to compute fluid properties and velocities.

Streamline curvature methods. An irregular computational grid is formed on the meridional section of the machine by the intersection of calculation planes and the projection of streamlines. Initial placement of streamlines is guessed starting from the known inlet conditions. The streamlines, along which energy and momentum are conserved, are adjusted by iteration of the physical laws (which take radial velocities and accelerations into account) and consideration of empirical data. A detailed description of this method is presented in Chapter 5. Figure 2.5 shows a typical computational grid used in a streamline curvature computation.

Finite-difference methods. These methods are based on the initial definition of a system

Figure 2.4 A meridional section of a two-stage axial-flow compressor (reproduced from Hirsch and Denton, 1981).

Figure 2.5 Computational grid, for streamline curvature method, formed by the intersection of computational planes and the projection of streamlines (reproduced from Hirsch and Denton, 1981).

of grid points or nodes in the flow passage under consideration (see Figure 2.6). Finite-difference equations are then written at each grid point. The equations are solved (usually by iteration) such that the governing physical laws and empirical information are satisfied. This gives the fluid properties and velocities at each node, from which streamline patterns can be constructed. These methods are also commonly referred to as matrix methods.

Finite element methods. The flow passage under study is subdivided into elements by a network of lines such as those illustrated in Figure 2.7. The physical laws and empirical input are formulated so that flow properties and velocities may be calculated at the corner points of each element (the nodes) by some iterative method.

Quasi-three-dimensional model. The flow is represented by a network of intersecting blade-to-blade streamsurfaces (S_1 surfaces) and hub-to-casing streamsurfaces (S_2 surfaces) (and sometimes S_3 surfaces; perpendicular to both S_1 and S_2 surfaces). Since circumferential flow nonuniformities can be calculated on the S_1 surfaces, the axisymmetric flow assumption of the two-dimensional model is relaxed. The model is so-called because the computations on each set of surfaces are two-dimensional. The flowfield on each set of surfaces is solved interactively with the solution on any surface influencing the solutions on each intersecting surface. Examples of S_1 and S_2 surfaces are illustrated in Figure 2.8. Any of the numerical methods used to solve two-dimensional flowfields can be used on each set of surfaces here.

Three-dimensional model. Here the full flowfield is calculated by solving the three-dimensional governing equations directly. The Navier-Stokes equations or simplified forms such as the Euler equations (which exclude viscosity) may be used. Many of these methods also model turbulence to some degree. Development of such methods is a very active area of research at present and some impressive results have been published over the last several years. However, use of these computational methods in routine design and analysis of turbomachinery is still very limited due to the great time and expense required to run them. Solutions of the equations can be obtained using numerical schemes, such as **finite difference** and **finite element methods**, which may be extended to three-dimensions. Another technique used here can be categorized as follows.

Figure 2.6 Grid nodes for finite-difference computation (reproduced from Hirsch and Denton, 1981).

Figure 2.7 Flow passage of two-stage compressor divided into finite elements (dashed lines) (reproduced from Hirsch and Denton, 1981).

Figure 2.8 Blade-to-blade (S_1) surface and hub-to-casing (S_2) surface for a blade passage (reproduced from Hirsch and Denton, 1981).

Figure 2.9 Blade passage subdivided into finite volume elements (reproduced from Hirsch and Denton, 1981).

Finite volume methods. The flow passage is subdivided (in three dimensions) into a large number of small volume elements which conform to the passage geometry (see Figure 2.9). Integral conservation equations are then formulated from the physical laws and empirical data. The equations are solved (usually using an iterative time marching scheme) giving the fluid properties and velocities at each surface of every volume element.

2.3 Spanwise Mixing in Axial Turbomachinery

This section reviews the many investigations of spanwise mixing which have been conducted over the past decade. Much controversy and criticism surrounding the primary mechanisms responsible for spanwise transport arose as a result of the earlier investigations involving axial-flow compressors. Increased research activity into spanwise mixing, in the years following, resolved many of these conflicting views. Only very recently, however, have the origins of spanwise mixing been investigated in axial-flow turbines; these investigations have revealed some fundamental differences.

As well, this section documents the spanwise mixing models which have been developed. These models have been incorporated into various throughflow computation methods with varying degrees of success. Some of the models reviewed formed the basis of the techniques used in the present study.

2.3.1 Experimental Investigations

Adkins and Smith (1982) were the first to thoroughly address the issue of spanwise mixing in axial compressors. Although they never specifically performed any experiments to support their claims, Adkins and Smith hypothesised that the primary mechanism responsible for the radial spreading of flow properties was convection by secondary flows. Based on this hypothesis, they developed a theory which incorporated radial mixing into a meridional throughflow computation, the results from which were quite impressive. Their computation procedure is discussed further in Section 2.3.2.

A few years after the work of Adkins and Smith (1982) a different hypothesis for the cause of radial mixing was proposed. Gallimore and Cumpsty (1986) suggested that the dominant mechanism responsible for spanwise mixing in axial compressors was turbulent diffusion and not the deterministic secondary flows. Their hypothesis was not unreasonable, considering the many sources of turbulence in a multistage turbomachine which might give rise to turbulent diffusion of flow properties across the span. Here, the term turbulence was used rather loosely to include any non-deterministic unsteadiness in the flow.

In an effort to identify the dominant mechanism, Gallimore and Cumpsty (1986) performed ethylene tracer experiments in two low-speed multistage axial-flow compressors. By injecting ethylene gas into the compressors and measuring the lateral spread at downstream locations it was possible to deduce the level of mixing directly. Figure 2.10.a shows a plot of ethylene concentration contours measured downstream of the third stator row, the injection points being just upstream of the same stator. Some evidence of radial secondary flow can be noticed near mid-span along the blade trailing-edge; however, the concentric, approximately round nature of the majority of the contours is indicative of a turbulent-type diffusion. Although the mixing is slightly more intense in the endwall regions, Gallimore and Cumpsty suggested that the mixing could be approximated as uniform along the span. Figure 2.10.b shows ethylene concentration contours measured downstream of the fourth rotor; the injection point being upstream of the first rotor. This shows the spread of ethylene over several stages and reveals that the flow is affected by diffusion right across the annulus.

From their experiments Gallimore and Cumpsty (1986) concluded that the dominant mechanism causing spanwise mixing in the latter stages of multistage compressors is indeed random, turbulent-type, diffusion. The convective mixing contribution of deterministic radial secondary flows, as suggested by Adkins and Smith (1982), was concluded to be comparatively small.

The conflicting hypotheses of Adkins and Smith (1982) and Gallimore and Cumpsty (1986) created much controversy in the years following. To try and resolve this dispute, Wisler et al. (1987) conducted a comprehensive measurement program in a large low-speed four-stage axial compressor. Using the ethylene tracer-gas technique, hot-wire anemometry, as well as

Figure 2.10.a Ethylene concentration contours showing extensive mixing by turbulent diffusion in the later stages of a compressor (reproduced from Gallimore and Cumpsty, 1986).

Figure 2.10.b Ethylene concentration contours measured four stages downstream of injection point (reproduced from Gallimore and Cumpsty, 1986).

conventional methods, Wisler et al. collected data which indicated that both secondary flows and turbulent diffusion play important roles in spanwise mixing. They concluded that the relative importance of each mixing mechanism was dependent on location within the machine: turbulent mixing was important everywhere and dominates in the 'freestream' or 'core' flow while convective mixing was at least equally important in low-momentum flow regions like endwall boundary layers, blade boundary layers and wakes. Wisler et al. proposed a comprehensive mixing picture which included: diffusion by turbulence everywhere; convection of fluid near endwalls and along airfoil surfaces by secondary flow; tip-clearance vortices; endwall boundary layer skew; wake chopping; circumferential fluid transport by wake dispersion; and radial flow in the rotor blade wakes. The relative importance of each of these mechanisms was dependent on machine configuration and blade loading.

Despite the explanations of Wisler et al. (1987), the many Discussions which were submitted about the work left the mixing mechanism controversy somewhat unresolved. In their extensive Discussion, Gallimore and Cumpsty expressed reservations concerning the data interpretation and still maintained that the dominant contribution to radial mixing was turbulent diffusion. While they agreed that secondary flow was the primary cause of the circumferential spreading of the tracer-gas in the endwall region, they still questioned its contribution to spanwise mixing. The distortion of tracer-gas concentration contours on the endwall and blade surface corners were explained by Wisler et al. in terms of secondary flows; however, Gallimore and Cumpsty suggested that anisotropic turbulent diffusion could have the same effect. In fact, Li and Cumpsty (1991(b)) showed this to be the case in a simple turbulent duct flow. Tracer-gas injection near the corners of a rectilinear duct showed greater spreading parallel to the solid surfaces than normal to them, creating the types of concentration contours reported by Wisler et al. (1987).

The controversy over the importance of secondary flows in spanwise mixing was finally resolved when Leylek and Wisler (1991) published extensive experimental data as well as the results of flow analyses using a state-of-the-art CFD code. Leylek and Wisler performed extensive measurements using both the ethylene tracer-gas technique and hot-wire anemometry in a large, low-speed, four-stage axial compressor at design and off-design conditions. Their experimental results showed significant radial and circumferential convection, which increased in magnitude

with increased loading, in the endwall regions. These findings were confirmed using a three-dimensional Navier-Stokes code with high-level turbulence modelling. Figure 2.11 shows the computed secondary flow velocity vectors within a stator blade passage (95% chord) at design and off-design conditions. Thus, with both experimental and computational evidence, Leylek and Wisler were able to show conclusively that radial secondary flows were at least as important as turbulent diffusion as a mechanism for spanwise mixing in the endwall regions of axial-flow compressors.

This was not the first time that CFD had been used to confirm the significance of secondary flows in axial-flow turbomachinery. Kirtley et al. (1990) analyzed the flow in a two-stage turbine using a 3-D Navier-Stokes code and concluded that

'while diffusion models may work extremely well, they are fundamentally incorrect in regions of large secondary flows. We expect turbulent diffusion models to under-predict spanwise mixing near endwalls and in low aspect ratio machines where secondary flows are substantial.'

Li and Cumpsty (1991(a)) were the first to measure and compare the mixing levels in the various stages of a multistage axial compressor. From tracer-gas measurements taken in a four-stage compressor equipped with inlet guide vanes, they surprisingly found that the level of mixing was almost constant along the machine; with higher levels in the endwall regions. Li and Cumpsty (1991(b)) concluded that thickened endwall boundary layers caused by rotor tip clearance flows were a major cause of this localized increase in mixing level.

The effect of rotor tip clearance was further investigated by Goto (1991). With experiments performed on a single stage axial compressor, he confirmed that the tip clearance flow was dominant in creating increased levels of mixing in the casing region. As well, it was found that the peak mixing level near the casing increased linearly with rotor tip clearance, the extrapolated value for zero tip being equal to that found at midspan.

Kotidis and Epstein (1991) studied spanwise mixing in a high pressure ratio transonic compressor stage and found spanwise transport equal to several percent of the total mass flow (3.5% moved toward the tip while 1.6% moved toward the hub). They suggested that vortices

Figure 2.11 Computed secondary flow velocity vectors, at 95% chord of stator passage, showing significant radial velocity. (reproduced from Leylek and Wisler, 1991).

shed from the blading could be another mechanism for spanwise mixing and proposed a quasi-three-dimensional model of a vortex street wake. The model was able to predict all of the transport towards the hub but only about 20% of the outward transport. They hypothesized that the remainder of the transport towards the rotor tip occurred in the regions of separated flow on the blade suction surface.

As far as the author is aware, the only experimental investigation of spanwise mixing in axial-flow turbines is the very recent work of Lewis (1993(a)). It should be noted that the work of Lewis was published at a point when the present research was largely complete. Lewis took measurements in a lightly loaded four-stage turbine and a more heavily loaded three-stage turbine using hot-wire anemometry and a tracer-gas technique. Both turbines had fairly large hub-to-tip ratios of 0.85. In the first stage of each turbine it was found that spanwise transport was confined to the endwall regions where secondary flows were active. This is indicated in the tracer-gas concentration contours of Figure 2.12.(a) where the core flow is essentially inviscid with a low level of turbulence. However, in the later stages, significant spanwise transport was seen across the whole span. Tracer-gas results downstream of the third stator of the lightly loaded turbine clearly show this (Figures 2.12.(b) and (c)). The level of turbulent diffusion was found to be fairly uniform across the whole span but at levels which were slightly greater than those measured by Gallimore and Cumpsty (1986) in two multistage compressors. This was surprising considering that the environment within a compressor, with its thicker boundary layers and wakes, is more favourable for the development of turbulence than that within a turbine. This suggested that a significant portion of the turbulence was produced by the cutting of vortices through successive blade rows.

The contribution of convective secondary flows to spanwise mixing was found to be large in both of the turbines investigated by Lewis (1993(a)). The convective mixing levels computed by the tracer-gas contours were found to be approximately twice as great as those due to turbulent diffusion. This is almost certainly the result of the larger flow turning in turbines, as compared with compressors.

Figure 2.12 Tracer-gas concentration contours in the (a) first stage; (b) third stage (with injection at endwalls); and (c) third stage (with injection away from endwalls) of an axial turbine (reproduced from Lewis, 1993(a)).

2.3.2 Predictive Methods

A comprehensive review of predictive efforts for radial transport effects in axial compressors was given by Wennerstrom (1991). This section describes some of the more important works covered by Wennerstrom as well as recent developments. Fully three-dimensional methods, such as those which solve the Navier-Stokes equations directly, are not considered here. Rather, this section focuses on methods which Wennerstrom refers to as hybrid methods. These methods take the approach of superimposing semi-empirical models of secondary flow effects onto otherwise classical axisymmetric solutions.

One of the first such hybrid methods was that of Hearsey (1975). Hearsey assumed that the major secondary flow mechanism was blade boundary layer and wake centrifugation. Hearsey then assumed that the extent of radial movement of this low-momentum fluid was proportional to the ratio of radial to axial shear stress on a fluid element in the blade surface boundary layer. This analysis, when applied to a great number of positions along the blade span, resulted in a redistribution of the loss coefficient at the blade trailing edge plane. However, the analysis still did not allow for any transport of mass or energy between axisymmetric streamtubes which would result from secondary flow. To remedy this, Hearsey introduced a turbulent mixing model using the momentum and energy equations for laminar flow, except that the values of molecular viscosity and thermal conductivity were replaced by appropriate values of eddy viscosity and eddy thermal conductivity. These turbulent eddy properties were left as empirical inputs and were considered uniform throughout the flow. When applied to a five-stage axial compressor, Hearsey's computations showed the correct general trends.

The first hybrid method that could be considered comprehensive and successful was that of Adkins and Smith (1982). As briefly mentioned in the preceding section, Adkins and Smith hypothesised that secondary flows were primarily responsible for spanwise mixing in compressors. They developed five secondary flow models based on inviscid, small-perturbation theory to include the effects of mainstream non-free vortex flow, endwall boundary layers, tip clearances, blade end shrouding, and blade boundary layer and wake centrifugation. Because of the inviscid theory used, application of the method was restricted to design-point operation where losses are minimal and regions of separated flow are small.

Adkins and Smith (1982) modelled the spanwise mixing as a diffusion process. The mixing equation which they derived gave the streamwise fluid property gradients as

$$\frac{\partial \bar{P}}{\partial x} = \beta \frac{\partial^2 \bar{P}}{\partial z^2} \quad (2.2)$$

where $\bar{P}$ is a cross-passage area-averaged fluid property, x is the streamwise direction, z is the spanwise direction, and β is a mixing coefficient. This equation was applied to a meridional throughflow computation in which the properties mixed were stagnation pressure, stagnation temperature, and angular momentum. The mixing coefficient, β , was related to the calculated secondary flow velocities by an expression which, in simplified form, is

$$\beta = \frac{x}{a} \int_0^a \frac{w^2}{u^2} dy \quad (2.3)$$

where a is the blade passage width (pitch), y is the pitchwise direction, and w and u are the local secondary flow velocities in the spanwise and streamwise directions, respectively. The actual expression used to calculate the local mixing coefficient was a modified form of Equation (2.3) which included empirical factors. These factors allowed for viscous dissipation of the secondary flowfield and interactions with endwalls and succeeding blade rows.

Results from the computation method of Adkins and Smith (1982) were quite impressive. Figure 2.13.(a) shows the results near the mean radius for the radial flow angles downstream of an untwisted rotor. The measured and computed stagnation pressure profiles at the exit of a low aspect ratio three-stage compressor are shown in Figure 2.13.(b). The most striking differences between computations with and without spanwise mixing are seen in the stagnation temperature profiles for the rear stages of a nine-stage compressor, as shown in Figure 2.13.(c).

The computation method of Gallimore (1986) was based on a different premise than that of Adkins and Smith. Following the conclusions of Gallimore and Cumpsty (1986), the fundamental assumption of Gallimore was that spanwise mixing in compressors was mainly the result of random turbulent diffusion. He used a method similar to that of Hearsey (1975) to include the effects of turbulent spanwise mixing into a simple streamline curvature computation. Like Hearsey, Gallimore used the energy and momentum equations for steady laminar flow;

Figure 2.13 (a) Computed radial flow angle aft of rotor. (b) Stagnation pressure profiles at exit of three-stage compressor. (c) Stagnation temperature profiles in the rear stages of a nine-stage compressor (reproduced from Adkins and Smith, 1982).

replacing the molecular viscosity and thermal conductivity with an eddy viscosity and eddy thermal conductivity, respectively. The eddy viscosity, μ_t , was related to a mixing coefficient, ϵ , through the turbulent Schmidt number,

$$Sc_t = \frac{\mu_t}{\rho \epsilon} ,$$

and to the eddy thermal conductivity, k_t , through the turbulent Prandtl number,

$$Pr_t = \frac{\mu_t c_p}{k_t} .$$

Gallimore assumed that both Sc_t and Pr_t were equal to 1.0 and thus only the mixing coefficient, ϵ , was necessary to calculate the level of turbulent mixing (which was considered constant throughout the flow). Hearsy left the mixing coefficient as an empirical input. However, Gallimore derived an expression which defined the mixing coefficient in terms of stage geometry, loss and flow coefficient. Further details of this are given in the next section.

From the fundamental equations, Gallimore derived expressions which gave the streamwise gradients of stagnation enthalpy, entropy, and angular momentum. These expressions allowed for the radial diffusion of heat and momentum and modified the original isentropic flow assumptions of a streamline curvature program previously described by Denton (1978). From the energy equation, the streamwise change in entropy was calculated using

$$\frac{\partial S}{\partial m} = \frac{1}{r \rho T v_m} \frac{\partial}{\partial r} \left(r k_t \frac{\partial T}{\partial r} \right) + \frac{\Phi}{\rho T v_m} + \frac{\partial S_e}{\partial m} , \quad (2.4)$$

where v_m is the meridional velocity, and the last term on the right-hand-side represents the traditional empirical loss coefficients derived from cascade correlations. The dissipation function, Φ , was given by

$$\Phi = \mu_t \left[\left(\frac{\partial v_x}{\partial r} \right)^2 + \left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \right)^2 \right]$$

and represents the dissipation of kinetic energy through turbulent diffusion. When the effects of the eddy viscosity were included in the momentum equations, the existence of tangential and axial

shear stresses was implied. The tangential shear stress, given by

$$E_\theta = \frac{\partial}{\partial r} \left[\mu_t \left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \right) \right] + \frac{2\mu_t}{r} \left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \right),$$

was added to the tangential blade force term and the axial shear stress, given by

$$E_z = \frac{1}{r} \frac{\partial}{\partial r} \left(\mu_t r \frac{\partial v_z}{\partial r} \right),$$

was added to the axial blade force term. The boundary conditions used by Gallimore (1986) were those of frictionless, adiabatic walls. However, modifications by Howard and Gallimore (1992) allowed the no-slip condition to be enforced at the annulus walls, giving more realistic results in these regions.

Like those of Adkins and Smith (1982), the computational results published by Gallimore (1986) were impressive. Exit total temperature profiles from two three-stage axial compressors are shown in Figures 2.14.(a) and (b). The computed profiles which include the mixing model of Gallimore agree very well with measurements of Gallimore and Cumpsty (1986).

As Wisler et al. (1987) pointed out, in order for any throughflow computation to be realistic, the radial mixing effects of both turbulent diffusion and secondary flows must be modelled. The first effort along these lines was the computation method of De Ruyck and Hirsch (1988). Their method was a more complex quasi-three-dimensional computation performed on S_1 and S_2 surfaces. Instead of using inviscid secondary flow theory like Adkins and Smith (1982), De Ruyck and Hirsch employed an integral approach in a model consisting of an inviscid core flow and viscous boundary layer and wake regions. Spanwise mixing in this two-zone model was accomplished with a diffusion-type equation similar to that of Adkins and Smith but with a turbulent diffusion coefficient added to the mixing coefficient computed from secondary flows. With a great many assumptions, this turbulent diffusion coefficient was the only empirical parameter used in the computation. When the computational procedure was applied to a compressor cascade and two different compressor stages, the results were encouraging. A more thorough description of the computational procedure was given by De Ruyck et al. (1990).

Figure 2.14 Total temperature profiles from three-stage axial compressors with (a) low aspect ratio and (b) higher aspect ratio (reproduced from Gallimore, 1986).

All of the computational methods mentioned thus far have not attempted to include the effects of shocks caused by transonic flow. Kaldellis et al. (1990) appear to have been the first to incorporate a model for shock-secondary flow interaction aimed at dealing with this effect in the rotor tip regions. Like De Ruyck and Hirsch (1988), Kaldellis et al. used a two-zone flow model but rather than modelling turbulent diffusion directly they made use of the mean kinetic energy integral equation. The method was tested on two single-stage transonic compressors for which results looked reasonable. The fact that the method of Kaldellis et al. had some success without a turbulent diffusion model suggests that turbulent diffusion may be less important in transonic compressor flows. However, Erkilet and Ucer (1990) applied the method of Gallimore (1986) (which models spanwise mixing exclusively as a turbulent diffusion process) to a two-stage transonic fan and also obtained results which were quite good.

More recently, Li and Chen (1992) have developed a complex system of equations which unify the effects of turbulent diffusion and convection by secondary flow for meridional flowfield calculations in axial-flow compressors. The equation system revealed that spanwise mixing in the meridional flowfield is caused by molecular motion, turbulent diffusion, and circumferential non-uniformities, which include secondary flow effects. Li and Chen (1992) also proposed a turbulence model based on a three-dimensional shearing structure rather than the two dimensional structure of previous models. They compared their computational results to those of experiments performed on five compressors as well as to computations from previous theories. The computations of Li and Chen (1992) agreed well with the experiments and showed some improvement over the earlier theories.

Thus far, all of the computation methods discussed have been developed and somewhat fine tuned specifically for the analysis of compressor flows. As far as the author is aware, the only throughflow computation method which includes spanwise mixing and was developed specifically for the analysis of turbine flows was the very recent work of Lewis (1993(b)). At the expense of sophistication and rigour, Lewis took a simple approach to developing a spanwise transport model and incorporated it into an existing throughflow scheme previously described by Denton (1978). Based on the experimental findings of Lewis (1993(a)), the spanwise transport model included the effects of turbulent diffusion as well as secondary flow.

Since spanwise transport can only be incorporated into a meridional throughflow computation by the introduction of a diffusive term, Lewis (1993(b)) modelled the spanwise transport of some fluid property, Q , using

$$v_m \frac{\partial Q}{\partial m} = \epsilon \frac{\partial^2 Q}{\partial z^2} , \quad (2.5)$$

where m is the meridional direction (streamwise direction in the meridional plane), z is the spanwise direction, and ϵ is an effective diffusion (mixing) coefficient. This expression is very similar to Equation (2.2) used by Adkins and Smith (1982). However, unlike the mixing coefficient of Adkins and Smith (β in Equation (2.2)) which was based solely on computed secondary flows, the effective diffusion coefficient of Lewis (1993(b)) is composed of a turbulent diffusion term and a term based on secondary flow considerations. The effective diffusion coefficient was given by

$$\epsilon = \epsilon_t + \epsilon_{sf} , \quad (2.6)$$

where ϵ_t represented the contribution of turbulent diffusion and ϵ_{sf} the contribution of secondary flow. In order to estimate these terms, Lewis (1993(b)) developed two scaling expressions based on simple theory and adjusted the empirical parameters using the experimental data of Lewis (1993(a)). These expressions will be presented in the following section. Considering the simplicity of the spanwise transport model used, the results achieved by Lewis (1993(b)) were quite good.

2.3.3 Mixing Coefficients

The objective of this section is to present the methods by which the various mixing coefficients, introduced in several of the preceding computation methods, were calculated and also to give some of the published values for the many test cases covered in the literature.

Recall from Section 2.3.1 that Gallimore and Cumpsty (1986) used an ethylene tracer-gas technique to investigate mixing levels in two axial-flow compressors. Using the measured concentration spreads of the gas at downstream planes and an expression derived from the

diffusing gas theory of Hinze (1975), they were able to estimate typical turbulent mixing coefficients for the two compressors. They used the mixing theory of Adkins and Smith (1982) to calculate mixing coefficients based on both predicted and measured secondary velocities. For comparative purposes, Gallimore and Cumpsty also estimated typical eddy viscosities for the two compressors using the turbulent flat plate boundary layer and two-dimensional wake mixing theories of Schlichting (1979). Mixing coefficients were calculated from these eddy viscosities under the assumption that the turbulent Schmidt number for the flows was approximately equal to one.

As mentioned in the previous section, Gallimore and Cumpsty (1986) also developed an expression for predicting the turbulent mixing coefficient for use in the computation method of Gallimore (1986). The expression gave the mixing coefficient in terms of compressor stage geometry, loss and flow coefficients and was derived by considering the generation and decay of turbulence in a repeating stage. The predicted mixing coefficient, ϵ , nondimensionalized by stage length, L_s , and mean axial velocity, V_s , was estimated using

$$\frac{\epsilon}{V_s L_s} = \frac{A t}{L_s} \left[\frac{2\omega(t/L_s)}{3\phi^2} \right]^{1/3}, \quad (2.7)$$

where A was given by

$$A = \left[\frac{\overline{u'v'}}{q^2} \right].$$

In Equation (2.7), t is the blade thickness, ω is the loss coefficient, and ϕ is the flow coefficient. The loss and flow coefficients are given by

$$\omega = \frac{\Delta p_L}{\rho U_m^2}$$

and

$$\phi = \frac{V_r}{U_m}$$

respectively, where Δp_L is the difference between the ideal and actual stage static pressure rise and U_m is the mean blade speed. In the expression for A , q^2 is the turbulence velocity scale given by

$$q^2 = \frac{1}{3} \left(\overline{u'^2} + \overline{v'^2} + \overline{w'^2} \right).$$

Gallimore and Cumpsty (1986) suggested that A was approximately equal to 0.4.

The values of dimensionless mixing coefficient calculated by Gallimore and Cumpsty (1986) using the methods described above, are given in Table 2.1, for the two compressors tested. Both Compressor A and Compressor B are low-speed four-stage machines, Compressor A being the larger of the two.

Wisler et al. (1987) also used an ethylene tracer-gas technique to measure mixing levels in a large, low-speed, four-stage, axial compressor. For their test compressor, Wisler et al. computed mixing coefficients using the diffusing gas theory of Hinze (1975), the secondary flow mixing model of Adkins and Smith (1982), and the turbulent diffusion mixing model of Gallimore and Cumpsty (1986). Normalized values of mixing coefficients calculated at the third stator exit are presented in Figure 2.15. Using the model of Adkins and Smith (1982), the overall value of the normalized mixing coefficient for the entire compressor was found to be 2.57×10^{-4} .

In the quasi-three-dimensional throughflow computation method of De Ruyck and Hirsch (1988), the turbulent mixing coefficient was left as an empirical input. Using the two-dimensional wake mixing theory of Schlichting (1979), De Ruyck and Hirsch suggested dimensionless turbulent mixing coefficients of 1.0×10^{-3} for a compressor blade wake at design point and 6.0×10^{-4} for a compressor blade wake near stall. In applying Equation (2.7) (developed by Gallimore and Cumpsty, 1986) to one of their test rotors, De Ruyck and Hirsch suggested values for the constant A of 0.1 for a compressor blade wake at design point and 0.5 for a compressor blade wake near stall. These are lower than the value of 0.4 suggested by Gallimore and Cumpsty (1986). When

Calculation Method	Compressor A	Compressor B
Measured ethylene spreads	1.8×10^{-3}	3.8×10^{-3}
Model of Adkins and Smith (1982)		
• based on predicted radial velocities	-	3.9×10^{-3}
• based on observed radial shifts	5.6×10^{-4}	9.7×10^{-4}
Turbulent flat plate boundary layer	3.2×10^{-4}	7.8×10^{-4}
Two-dimensional wake mixing	2.7×10^{-3}	2.7×10^{-3}
Equation (2.7)	1.6×10^{-3}	2.1×10^{-3}

Table 2.1 Dimensionless mixing coefficients calculated by Gallimore and Cumpsty (1986).

Figure 2.15 Normalized mixing coefficients, aft of the third stage of a four-stage compressor, calculated using three different models (reproduced from Wisler et al., 1987).

De Ruyck and Hirsch (1988) tested their computation method on a linear cascade and two different compressor rotors, they used dimensionless mixing coefficients of 2.0×10^{-4} .

When Erkilet and Ucer (1990) applied the mixing model of Gallimore (1986) to their own computational scheme, they set the mixing coefficient, ϵ , equal to $0.0035 \text{ m}^2/\text{s}$ for an annular duct flow of constant area. Using Equation (2.7) of Gallimore and Cumpsty (1986), Erkilet and Ucer calculated a value of $\epsilon = 0.0018 \text{ m}^2/\text{s}$ for a two-stage, low-aspect-ratio, transonic fan used to verify their computations.

Li and Cumpsty (1991(a) and (b)) published mixing coefficients for the three different compressors which they investigated. Their values were calculated using the Hinze (1975) mixing model, as modified by Gallimore and Cumpsty (1986), and were presented along with values calculated by applying Equation (2.7) to one of the test compressors. These results are presented in dimensionless form in Figure 2.16.

Goto (1991) studied the effects of rotor tip clearance on spanwise mixing. He measured the diffusion of a tracer-gas through a single stage compressor and computed values of dimensionless turbulent mixing coefficient using the mixing model of Hinze (1975) (modified by Gallimore and Cumpsty, 1986). The spanwise variation of mixing coefficients, evaluated downstream of the stator, are presented in Figure 2.17 for different rotor tip clearances.

Radial transport in a transonic compressor stage was investigated by Kotidis and Epstein (1991), who also used a tracer-gas measuring technique. They again used the modified gas diffusion model of Hinze (1975), as well as Equation (2.7) of Gallimore and Cumpsty (1986), to calculate mixing coefficients for their single stage compressor. These results are shown in Figure 2.18 and compared with normalized mixing coefficients from a low-speed multistage compressor (that of Wisler et al., 1987).

Mixing coefficients computed by the models of Gallimore and Cumpsty (1986) and Adkins and Smith (1982) were compared by Dring (1992) for measurements taken in two axial compressors. The values obtained from the Adkins and Smith model were based on measured, rather than predicted, radial velocities and were found to be generally lower than those calculated

Figure 2.16 Dimensionless mixing coefficients calculated at various points in three compressors (reproduced from Li and Cumpsty, 1991(b)).

Figure 2.17 Dimensionless mixing coefficients in a single stage compressor with different tip clearances (note: z is axial direction) (reproduced from Goto, 1991).

Figure 2.18 Normalized mixing coefficients from a transonic compressor stage compared to those calculated in previous studies (reproduced from Kotidis and Epstein, 1991).

with the model of Gallimore and Cumpsty (Equation (2.7)).

Lewis (1993(a)) was the first to publish mixing coefficient values for turbines. As mentioned in the previous section, he performed tracer-gas measurements in a lightly-loaded, four-stage, axial-flow turbine and a more heavily-loaded, three-stage turbine. Lewis, however, assumed that an effective mixing coefficient, ϵ , could be defined such that

$$\epsilon = \epsilon_d + \epsilon_{cv} \quad (2.8)$$

where ϵ_d is the familiar turbulent diffusion (mixing) coefficient and ϵ_{cv} is a mixing coefficient based on convective mechanisms. ϵ_d was determined by applying the mixing model of Hinze (1975) to the diffusion of the tracer-gas through the stators. It was assumed that this value was approximately constant throughout the machine. The effective mixing coefficient, ϵ , was calculated by circumferentially averaging the diffusion results for the tracer-gas injected through the rotors. ϵ_{cv} was then calculated by applying Equation (2.8).

In his computational model, Lewis (1993(b)) derived expressions for estimating the turbulent diffusion coefficient, ϵ_d , and the diffusion coefficient based on calculated secondary flows, ϵ_{sf} , for use in Equation (2.6). Dimensionless values of ϵ_d were calculated using

$$\frac{\epsilon_d}{V_x L_d} = A_d \left[\frac{U_\infty (\tan \alpha_2 - \tan \alpha_1) (1 - \eta)}{\eta V_x} \right]^{\frac{1}{3}} \left[\frac{L_d}{L_s} \right]^{\frac{4}{3}} \quad (2.9)$$

where A_d is a constant of proportionality (empirically set to 0.2), α_1 and α_2 are the absolute flow angles (relative to axial) at stage inlet and exit, respectively, η is the isentropic efficiency, and L_d is a length scale for a typical turbulent eddy. Lewis (1993(b)) estimated L_d using

$$L_d = 5 Y_p a \cos \alpha_2$$

where Y_p is the profile loss coefficient and a is the blade pitch. The secondary flow diffusion coefficient was calculated using

$$\frac{\epsilon_{sf}}{V_1 L_s} = A_{sf} \left[\frac{\Delta^2 a' f(\delta/a')}{h V_1 \cos^2 \alpha_1} \right]^{\frac{1}{2}} \frac{a'}{L_s} ,$$

where A_{sf} is a constant (empirically set to 0.02), a' is the projected blade pitch equal to $a \cos \beta_1$, h is the enthalpy at mid-span, $f(\delta/a')$ is a series expansion and a function of inlet boundary layer thickness, δ , and Δ is given by

$$\Delta = \frac{-\sin(\alpha_1 - \beta_1) \cos \beta_1}{\cos \beta_2} + \frac{\cos(\alpha_1 - \beta_1)}{\cos \beta_1 \cos \beta_2} \left(\frac{\sin 2\beta_2 - \sin 2\beta_1}{2} + \beta_2 - \beta_1 \right)$$

for a rotor, and by

$$\Delta = \frac{-\sin(\beta_1 - \alpha_1) \cos \alpha_1}{\cos \alpha_2} + \frac{\cos(\beta_1 - \alpha_1)}{\cos \alpha_1 \cos \alpha_2} \left(\frac{\sin 2\alpha_2 - \sin 2\alpha_1}{2} + \alpha_2 - \alpha_1 \right)$$

for a stator, where β represents the relative flow angle with respect to the axial direction.

The experimentally determined and computed values of these dimensionless mixing coefficients are given in Table 2.2 for both turbines investigated by Lewis (1993(a)). Since it was found that the effective mixing levels were lower in the first stages of each turbine, Lewis suggested that Equation (2.6) be modified to the form

$$\epsilon = (\epsilon_d + \epsilon_{sf}) \tanh \left(\frac{z}{1.5 L_s} \right) ,$$

where z , in this case, is the axial direction. This modification was justified by the results shown in Figure 2.19.

The range of turbulent mixing coefficients presented in this section acted as a guide for the selection of values used in the present study. Average values of the mixing coefficient based on secondary flows (from Lewis, 1993(b)) are compared to values calculated in the present study. More details on this are given in Chapter 8.

Mixing Coefficient	LL Turbine	HL Turbine
Experimental $\epsilon_d/(V_x L_x)$	0.0007-0.0009	0.0018-0.0026
Calculated $\epsilon_d/(V_x L_x)$	0.0009	0.0020
Experimental $\epsilon_{sj}/(V_x L_x)$	0.0016-0.0026	0.0042-0.0056
Calculated $\epsilon_{sj}/(V_x L_x)$	0.0027 @ hub 0.0015 @ tip	0.0071 @ hub 0.0051 @ tip

Table 2.2 Dimensionless diffusive and secondary flow mixing coefficients for turbines investigated by Lewis, 1993(a).

Figure 2.19 Axial variation of midspan effective mixing coefficients (dimensionless) for two turbines (reproduced from Lewis, 1993(b)).

SECONDARY FLOW COMPUTATION: THEORY

3

In the present study, the method used for computation of secondary flows is based on that of both Glynn (1982) and Gregory-Smith and Okan (1991). The present computation combines elements of both methods which were briefly discussed in Section 2.1.2. This chapter describes the secondary flow model used, the simplifying assumptions made, and presents derivations of the equations involved. Also, the computational solution procedure is outlined in a series of steps. Details of some of the numerical procedures involved are presented in Chapter 4.

3.1 Flow Model

Although secondary flows encompass a wide variety of phenomena (see Section 2.1.1), the experiments of Marchal and Sieverding (1977) show that the passage vortex is the dominant feature in turbine cascades which exclude tip leakage flows. It is also known that, with the favourable pressure gradient present in turbines, boundary layer phenomena such as blade wakes and endwall crossflows become less significant as compared with compressor flows. Thus, the computational method used here is based on the classical inviscid secondary flow model first described by Hawthorne in 1955. This model predicts the development of a passage vortex and trailing vortex sheet when an inlet shear flow is deflected through a cascade. The model is illustrated in Figure 3.1, where x' indicates the primary flow direction, y' indicates the crossflow direction, and z indicates the spanwise direction. The inlet shear flow features hub and casing boundary layers which possess vorticity normal to the inlet flow direction. This is indicated in Figure 3.1 by means of vortex filaments in the boundary layers. When the flow is deflected in the blade passage, streamwise components of vorticity are set up in the exit flow due to the distortion of the inlet vortex filaments. This creates two counter-rotating vortices, which represents the distribution of secondary velocities at the exit plane of the cascade. The vortex sheet at the

Figure 3.1 Classical secondary flow model of Hawthorne (adapted from Sieverding, 1985).

blade trailing edge is composed of two components: the trailing filament vorticity; and the trailing shed vorticity. The trailing filament vorticity is caused by the stretching of the inlet vortex filaments as they pass through the cascade. This filament stretching is due to the difference in velocities on the blade pressure and suction surfaces. The trailing shed vorticity is the result of the spanwise variation of blade circulation.

The computation methods of Glynn (1982) and Gregory-Smith and Okan (1991) are based on the flow model described above and use incompressible, inviscid vortex theory to predict the secondary vorticity field (and thus the secondary velocity field) at the blade trailing edge plane of a cascade. Unlike classical methods, this method allows for the distortion, by convection, of constant stagnation pressure surfaces (referred to as Bernoulli surfaces) as the flow passes through the cascade. The Bernoulli surfaces, which are composed of streamlines having the same stagnation pressure, lie in planes normal to the spanwise direction at the cascade inlet. However, as the flow travels through the blade passage, the induced secondary velocities cause the surfaces to deform as the streamlines are convected. The distortion of a Bernoulli surface as it passes through a blade passage is illustrated in Figure 3.2. To compute the exit plane secondary vorticity, an integral expression is evaluated along the convected streamlines. Since the secondary velocities within the blade passage must be known in order to predict the streamline convection, it can be seen that this method is an iterative one.

The computation of secondary flows is a three-dimensional problem. However, under the assumption that local gradients of the secondary velocities are negligible in the primary flow direction, this problem can be reduced to the solution of a two-dimensional stream function at the blade passage exit plane (Glynn and Marsh, 1980). In fact, the secondary velocities can be found at any intermediate plane parallel to the exit plane by dividing the passage up axially into several regions and treating each region as a blade passage with only a fraction of the cascade turning angle. This approach was introduced by Gregory-Smith and Okan (1991).

The blade passage is first divided into a number of regions of small turning so that, starting at the first region, a classical small angle formula (which neglects streamline convection) can be used to estimate the secondary flow velocities at the regional exit plane. These secondary flow velocities are considered to vary linearly from inlet to exit plane of the region and are used

Figure 3.2 Distortion of a Bernoulli surface, by convective secondary flow, as it passes through a blade passage (adapted from Bardon et al., 1975).

in calculating the convection of streamlines as they pass through the region. On a second iteration a more accurate integral expression is evaluated along streamlines to get a better estimate of secondary flow velocities at the regional exit plane. This exit flow is then used as the input to the next region and the process is repeated until the blade passage exit plane is reached. Figure 3.3 illustrates a blade passage profile divided into three computational regions of equal axial length.

This method has been shown to give accurate predictions of secondary velocities in cascades and was chosen because of its relative simplicity, and minimal input requirements. As mentioned previously, the secondary flow computation developed for the present study is a combination of the methods of Glynn (1982) and Gregory-Smith and Okan (1991). The source of the various components of the present method will be indicated at appropriate points throughout the remainder of this chapter.

3.2 Secondary Stream Function Equation

In this section, a two-dimensional equation relating secondary flow velocities and streamwise vorticity is derived. The secondary velocities are expressed in terms of a stream function and the equation is to be solved at the exit plane of each region as described in the previous section. This so-called "secondary stream function equation" is derived for an annular cascade then simplified to give an expression for use in linear cascades. The following derivation has been adapted from Glynn and Marsh (1980).

From the local coordinate system illustrated in Figure 3.3 it can be seen that

$$\frac{\partial}{\partial x'} = \frac{\partial}{\partial x} \cos \alpha + \frac{\partial}{\partial y} \sin \alpha .$$

It is assumed that there is no local variation of the flow in the primary flow direction so that,

Figure 3.3 Blade passage profile divided into computational regions and illustrating local primary streamline coordinate system.

$$\frac{\partial}{\partial x'} = 0$$

and therefore,

$$\frac{\partial}{\partial x} = -\tan \alpha \frac{\partial}{\partial y} = -\frac{\tan \alpha}{r} \frac{\partial}{\partial \theta} , \quad (3.1)$$

since $\partial y = r \partial \theta$. Now, in the (r, θ, x) cylindrical polar coordinate system, the equation of continuity is

$$\frac{\partial v_x}{\partial x} + \frac{1}{r} \frac{\partial v_\theta}{\partial \theta} + \frac{1}{r} \frac{\partial (r v_r)}{\partial r} = 0 .$$

Using Equation (3.1) the continuity equation becomes

$$\frac{\partial}{\partial \theta} (v_\theta - v_x \tan \alpha) + \frac{\partial}{\partial r} (r v_r) = 0 .$$

The secondary velocity components can now be expressed in terms of a secondary stream function, ψ , such that

$$r v_r = \frac{\partial \psi}{\partial \theta} \quad (3.2.a)$$

and

$$v_\theta - v_x \tan \alpha = -\frac{\partial \psi}{\partial r} . \quad (3.2.b)$$

The secondary stream function must now be related to the streamwise component of vorticity, ξ_x . The components of vorticity in the x and y directions are given by

$$\xi_x = \frac{1}{r} \frac{\partial v_r}{\partial \theta} - \frac{1}{r} \frac{\partial (r v_\theta)}{\partial r}$$

and

$$\xi_y = \frac{\partial v_x}{\partial r} - \frac{\partial v_r}{\partial x} ,$$

respectively. With the use of Equation (3.2.a) they can be rewritten as

$$\xi_x = \frac{1}{r^2} \frac{\partial^2 \psi}{\partial \theta^2} - \frac{1}{r} \frac{\partial (r v_\theta)}{\partial r} \quad (3.3.a)$$

and

$$\xi_y = \frac{\partial v_x}{\partial r} + \frac{\tan \alpha}{r^2} \frac{\partial^2 \psi}{\partial \theta^2} , \quad (3.3.b)$$

respectively. Now, vorticity in the streamwise or primary flow direction is given by

$$\xi_s = \xi_x \cos \alpha + \xi_y \sin \alpha$$

which, after substitution for the x and y components of vorticity, can be rewritten as

$$\xi_s = \frac{1}{r^2 \cos \alpha} \frac{\partial^2 \psi}{\partial \theta^2} - \frac{\cos \alpha}{r} \frac{\partial (r v_\theta)}{\partial r} + \frac{\partial v_x}{\partial r} \sin \alpha . \quad (3.4)$$

Differentiating Equation (3.2.b) with respect to r gives

$$\frac{\partial^2 \psi}{\partial r^2} = -\frac{\partial v_\theta}{\partial r} + \tan \alpha \frac{\partial v_x}{\partial r} + \frac{v_x}{\cos^2 \alpha} \frac{\partial \alpha}{\partial r} \quad (3.5)$$

and substitution for v_θ and $\frac{\partial v_\theta}{\partial r}$ (from Equations (3.2.b) and (3.5) respectively) in Equation (3.4) finally gives

$$\frac{\partial^2 \psi}{\partial r^2} + \frac{1}{r} \frac{\partial \psi}{\partial r} + \frac{1}{r^2 \cos^2 \alpha} \frac{\partial^2 \psi}{\partial \theta^2} = \xi_s \sec \alpha + \frac{v_x}{r} \frac{\partial}{\partial r} (r \tan \alpha) .$$

This is the most general form of the secondary stream function equation first developed by Glynn and Marsh (1980) and used by Gregory-Smith and Okan (1991). It is for use with annular cascades and is solved for ψ at any regional exit plane. The secondary velocity components are those in directions normal to the primary flow at the exit plane and are given by

$$v_r = \frac{1}{r} \frac{\partial \psi}{\partial \theta}$$

for the radial component and, after some manipulation of Equation (3.2.b),

$$v_\theta' = -\frac{\partial \psi}{\partial r} \cos \alpha$$

for the crossflow component (normal to both the streamwise and radial directions).

For annular cascades with large hub-to-tip ratio or for linear cascades, the secondary stream function equation can be simplified and expressed more logically in Cartesian coordinates as

$$\frac{\partial^2 \psi}{\partial y^2} + \cos^2 \alpha \frac{\partial^2 \psi}{\partial z^2} = \xi_s \cos \alpha . \quad (3.6)$$

This was the equation used by Glynn (1982) and for the purposes of simplicity, this equation was used in the present study. It is solved numerically at each regional exit plane in the blade passage. The details of the numerical method used to solve Equation (3.6), including specification of boundary conditions, are given in Section 4.1. Angle α , in the above equation, is the primary flow angle at the regional exit plane. The streamwise vorticity, ξ_s , in Equation (3.6) is the distributed secondary vorticity at the regional exit plane and is discussed further in the next section. The other streamwise vorticity components, contained in the trailing vortex sheet, do not enter the analysis but must be consistent with the calculated secondary velocities at the cascade exit plane. Expressions for these trailing vorticity components were presented in Section 2.1.2.

Once Equation (3.6) is solved for ψ , the secondary velocity components at the regional exit plane can be found using

$$w = \frac{\partial \psi}{\partial y} \quad (3.7.a)$$

for the spanwise component and

$$v' = -\frac{\partial \psi}{\partial z} \cos \alpha \quad (3.7.b)$$

for the crossflow component.

3.3 Distributed Secondary Vorticity

As mentioned in the previous section, it is necessary to know the distributed secondary vorticity at a regional exit plane in order to solve Equation (3.6). This section presents derivations of two expressions which can be used to calculate the required streamwise vorticity distribution. The two expressions differ in that one is a simplification of the other.

As a starting point, consider the intrinsic coordinate system of Lakshminarayana and Horlock (1972), illustrated in Figure 3.4. Using the notation of this coordinate system, the exact equation for streamwise vorticity can be written,

$$\begin{aligned} \frac{\partial}{\partial s} \left(\frac{\xi_s}{\rho V} \right) = & \frac{2\xi_n}{\rho V R_c} - \frac{1}{V^2 \rho^3} \left[\frac{\partial \rho}{\partial n} \frac{\partial \rho}{\partial b} - \frac{\partial \rho}{\partial n} \frac{\partial \rho}{\partial b} \right] + \frac{\mu}{\rho^2 V^2} s \cdot \nabla^2 \xi - \frac{1}{\rho V^2} s \cdot \nabla \left(\frac{\mu}{\rho} \right) \times (\nabla \times \xi) \\ & + \frac{4}{3} \frac{s}{\rho V^2} \cdot \nabla \left(\frac{\mu}{\rho} \right) \times \nabla (\nabla \cdot \mathbf{V}) . \end{aligned}$$

This equation comes from the Navier-Stokes equations. If the flow is assumed to be inviscid and incompressible, this equation greatly simplifies to

$$\frac{\partial}{\partial s} \left(\frac{\xi_s}{V} \right) = \frac{2\xi_n}{V R_c} . \quad (3.8)$$

With the assumption of inviscid flow, the vortex filaments and streamlines lie in surfaces of constant stagnation pressure (Bernoulli surfaces) then, with reference to Figure 3.4,

Figure 3.4 Intrinsic coordinate system (adapted from Lakshminarayana and Horlock, 1972).

$$\mathbf{b} \cdot \frac{\nabla p_o}{\rho} = \mathbf{b} \cdot (\mathbf{V} \times \boldsymbol{\xi}) = \boldsymbol{\xi} \cdot (\mathbf{b} \times \mathbf{V}) = (\boldsymbol{\xi} \cdot \mathbf{n}) V = \xi_n V . \quad (3.9)$$

Also,

$$\mathbf{b} \cdot \frac{\nabla p_o}{\rho} = \left| \frac{\nabla p_o}{\rho} \right| \cos \Gamma , \quad (3.10)$$

and thus substituting Equations (3.9) and (3.10) into Equation (3.8) and integrating in the streamwise direction from point 1 to point 2 along a streamline gives

$$\left(\frac{\xi_r}{V} \right)_2 - \left(\frac{\xi_r}{V} \right)_1 = 2 \int_1^2 \frac{|\nabla p_o / \rho|}{V^2 R} \cos \Gamma ds . \quad (3.11)$$

Now, for the purposes of the present study, it is assumed that the intrinsic (s, n, b) coordinate system corresponds to the local (x', y', z) coordinate system within the blade passage (see Figure 3.3) and that all streamlines closely correspond to primary streamlines. Thus, the principle radius of curvature for a streamline may be expressed as

$$\frac{1}{R_c} = \frac{\partial \alpha}{\partial s} .$$

If it is also assumed that the blade span is constant over the axial chord of the blades then the axial velocity is constant along streamlines and is given by

$$u = V \cos \alpha .$$

Substitution of the above two equations into Equation (3.11) gives

$$\xi_{r_2} = \frac{\cos \alpha_1}{\cos \alpha_2} \xi_{r_1} + \frac{2}{\rho u \cos \alpha_2} \int_{\alpha_1}^{\alpha_2} |\nabla p_o| \cos \Gamma \cos^2 \epsilon d\alpha .$$

Since the intrinsic binormal direction, given by $\mathbf{b}$, corresponds to the z direction it is easily seen that

$$|\nabla p_o| \cos \Gamma = \frac{\partial p_o}{\partial z} ,$$

so that the streamwise component of vorticity at point 2 along the streamline can be expressed as

$$\xi_{s_2} = \frac{\cos \alpha_1}{\cos \alpha_2} \xi_{s_1} + \frac{2}{\rho u \cos \alpha_2} \int_{\alpha_1}^{\alpha_2} \frac{\partial p_o}{\partial z} \cos^2 \alpha d\alpha . \quad (3.12)$$

This expression may be integrated along a passage streamline in order to find the streamwise component of vorticity at any point downstream of point 1 along the streamline. Variations of Equation (3.12) were used in both Glynn (1982) and Gregory-Smith and Okan (1991). In the context of the present computation, point 1 is located on a regional inlet plane while point 2 is located on a regional exit plane.

Now, if the angle of flow turning is small, then the convection of the Bernoulli surfaces can be neglected and $\frac{\partial p_o}{\partial z}$ remains constant along streamlines. In this case,

$$\left. \frac{\partial p_o}{\partial z} \right|_1 \frac{1}{\rho} = \frac{u \xi_{s_1}}{\cos \alpha_1} ,$$

at point 1 and Equation (3.12) reduces to

$$\xi_{s_2} = \frac{\cos \alpha_1}{\cos \alpha_2} \xi_{s_1} + \frac{\xi_{s_1}}{\cos \alpha_1 \cos \alpha_2} [(\alpha_2 - \alpha_1) + \frac{1}{2}(\sin 2\alpha_2 - \sin 2\alpha_1)] , \quad (3.13)$$

which is identical to Equation (2.1) developed by Came and Marsh (1974).

In order to evaluate Equation (3.12), it is necessary to know the distribution of $\partial p_o / \partial z$ along the streamlines. In order to know this, however, it is necessary to have knowledge of how the Bernoulli surfaces deform as they pass through the blade passage and thus have knowledge of the secondary flow within the passage. It is for this reason that Equation (3.13) must be used. Equation (3.13) is used to estimate the secondary vorticity distribution (and thus secondary velocities) at the regional exit plane. These secondary velocities are use to estimate the

convection of the streamlines through the region (see Section 3.4.). Then, in a second iteration through the region, Equation (3.12) is evaluated along each streamline to give a better estimate of the streamwise vorticity distribution at the regional exit plane.

To use Equations (3.12) and (3.13), it is also necessary to know the distributed secondary vorticity (and normal vorticity for Eqn. (3.13)) at each regional inlet plane. For regions downstream of the inlet region this presents no particular difficulties since the distributed secondary vorticity is known at the exit plane of the previous region. However, for the inlet plane of the first region (the inlet plane of the cascade) the streamwise and normal vorticity distributions must be calculated using known inlet conditions as must the normal vorticity for each regional inlet plane thereafter. The following section discusses how these distributions are calculated.

3.3.1 Inlet Vorticity Components

First consider the components of vorticity in the x and y directions which, in Cartesian coordinates, are given by

$$\xi_x = \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \quad (3.14.a)$$

and

$$\xi_y = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} . \quad (3.14.b)$$

respectively. Now, all components of velocity are generally known at the cascade inlet plane; however, derivatives in the x direction are generally not. Thus, ξ_x can be easily found by numerical differentiation of velocity components while ξ_y cannot. In order to determine ξ_y using known conditions at the inlet plane, first consider the Euler equation for steady incompressible flow in the z direction,

$$u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} = - \frac{1}{\rho} \frac{\partial p}{\partial z} .$$

By rearranging and adding additional terms to both sides of the equation we get,

$$u \frac{\partial u}{\partial z} - u \frac{\partial w}{\partial x} - v \frac{\partial w}{\partial y} + v \frac{\partial v}{\partial z} = \frac{1}{\rho} \frac{\partial p}{\partial z} + u \frac{\partial u}{\partial z} + v \frac{\partial v}{\partial z} + w \frac{\partial w}{\partial z}$$

which can be rewritten as,

$$u \left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \right) - v \left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \right) = \frac{1}{\rho} \left(\frac{\partial p}{\partial z} + \frac{\rho}{2} \frac{\partial u^2}{\partial z} + \frac{\rho}{2} \frac{\partial v^2}{\partial z} + \frac{\rho}{2} \frac{\partial w^2}{\partial z} \right).$$

Substituting Equations (3.14.a) and (3.14.b) into the above expression and rearranging gives

$$\xi_y = \frac{1}{u} \left(\frac{1}{\rho} \frac{\partial p_o}{\partial z} + v \xi_x \right). \quad (3.15)$$

The y component of vorticity is now expressed in terms of the known inlet conditions.

It should be noted here that all flow and blade angles are measured relative to the axial direction. Angles measured counter-clockwise from the axial direction are considered positive while angles measured clockwise from the axial direction are considered negative. Because the inlet flow is assumed to vary in the z direction only, the inlet velocity profile can be represented as $V(z)$ and the inlet total pressure profile can be represented as $p_o(z)$. Thus, the x and y components of vorticity at the cascade inlet may be expressed as

$$\xi_x = - \frac{dV(z)}{dz} \sin \alpha_i \quad (3.16.a)$$

and

$$\xi_y = \frac{1}{V(z) \cos \alpha_i} \left(\frac{1}{\rho} \frac{dp_o(z)}{dz} + V(z) \sin \alpha_i \xi_x \right) \quad (3.16.b)$$

respectively, where α_i is the inlet flow angle. The streamwise and normal components of vorticity at the cascade inlet can now be found using

$$\xi_z = \xi_x \cos \alpha_i + \xi_y \sin \alpha_i \quad (3.17.a)$$

and

$$\xi_n = \xi_y \cos \alpha_i - \xi_z \sin \alpha_i \quad (3.17.b)$$

respectively.

As mentioned in the previous section, the normal component (y' component) of vorticity is required at each regional inlet plane in order to evaluate Equation (3.13). An expression analogous to Equation (3.15) can be obtained by a simple coordinate system rotation so that the normal component of vorticity is given by

$$\xi_n = \frac{\cos \alpha}{u} \left[\frac{1}{\rho} \frac{\partial p_o}{\partial z} - \frac{\partial \psi}{\partial z} \xi_z \cos \alpha \right], \quad (3.18)$$

where α is the local primary flow angle and the local normal velocity component, v' , has been expressed in terms of the local secondary stream function.

3.4 Convection of Streamlines

This section presents the development of two simple expressions which are used to calculate the convection of streamlines due to secondary flow in the blade passage. The equations are first developed for an annular cascade, as in Gregory-Smith and Okan (1991), and then simplified, to the form used by Glynn (1982), for use in the present study.

In order to calculate the convection of streamlines through a region, it is necessary to divide the region into several sub-regions of axial length Δx . The region is thus divided axially by intermediate planes parallel to the regional inlet and exit planes. Two intermediate planes are illustrated in Figure 3.5. Now, in the (r, θ, x) coordinate system, the tangential and radial convection of a streamline due to secondary flow, over an axial distance Δx , is given by

$$\delta \theta = \frac{1}{r} \frac{v_\theta}{v_x} \Delta x$$

and

$$\delta r = \frac{v_r}{v_x} \Delta x$$

respectively. In terms of the secondary stream function, these expressions are

$$\delta \theta = -\frac{1}{r} \frac{\partial \psi}{\partial r} \frac{\Delta x}{v_x} \quad (3.19.a)$$

and

$$\delta r = \frac{1}{r} \frac{\partial \psi}{\partial \theta} \frac{\Delta x}{v_x} \quad (3.19.b)$$

respectively. It should be noted that this is only the convection caused by secondary flow. The streamline convection in the tangential direction due to the primary flow has been excluded since it does not affect the distribution of streamlines from one intermediate plane to the next.

As stated before, a simplification employed in the present study is that only linear cascades or annular cascades with large hub-to-tip ratio are to be analyzed. Thus, Equations (3.19.a) and (3.19.b) can be expressed more conveniently in Cartesian coordinates as

$$\delta y = -\frac{\partial \psi}{\partial z} \frac{\Delta x}{u}$$

and

$$\delta z = \frac{\partial \psi}{\partial y} \frac{\Delta x}{u}$$

respectively, since $\partial y = r \partial \theta$ and $\delta y = r \delta \theta$. The value of the secondary stream function used in the above equations is, in fact, a function of axial position but can be taken as an average value over the interval Δx . In terms of the secondary stream function values on intermediate planes 1 and 2 separated by an axial distance Δx , the above equations can be expressed as

$$\delta y = -\frac{1}{2} \left[\frac{\partial \psi_1}{\partial z} + \frac{\partial \psi_2}{\partial z} \right] \frac{\Delta x}{u} \quad (3.20.a)$$

and

$$\delta z = \frac{1}{2} \left[\frac{\partial \psi_1}{\partial y} + \frac{\partial \psi_2}{\partial y} \right] \frac{\Delta x}{u} \quad (3.20.b)$$

respectively. The derivatives of ψ_2 used in the above equations are those values found at the point where the streamline intersects intermediate plane 2. Since this intersection point is not known before the streamline convection is calculated, it can be seen that Equations (3.20.a) and (3.20.b) must be used iteratively to find the final intersection point on intermediate plane 2. Figure 3.5 illustrates the convection of a streamline between two intermediate planes after the first iteration. Here it can also be seen that the streamline convection due to the primary flow has no effect on streamline distribution from plane to plane since each plane has its own local coordinate system and the primary streamline intersects both planes at the same point.

3.5 Solution Procedure

This section gives a step-by-step description of how the theory, presented in the previous sections, is used to calculate the distribution of secondary flow velocities at the blade trailing edge plane of a cascade. Some procedures introduced in this section require more thorough discussion and are presented in Chapter 4.

Step 1: The blade passage is first divided into a number of computational regions (CRs) of equal axial length and turning angle. Each computational region is then further divided into four sub-regions, also of equal axial length and turning angle. Four sub-regions were chosen because the minimum number of points required to numerically evaluate an integral, using Simpson's one-third rule, is four. This will be discussed in a later step. The regions described above are illustrated in Figure 3.6, which shows the blade passage profile in the x - y plane.

Step 2: The computation starts at the blade leading-edge plane which is divided into a rectangular computational grid of points. The grid spacing in the pitchwise direction is uniform while grid point clustering is used in the spanwise direction to resolve the properties of thin inlet boundary layers. The details of this clustering procedure are given in Appendix B.

Figure 3.6 Blade passage terminology for secondary flow computation.

Step 3: The rectangular grid of points at the inlet is representative of the intersection of streamlines and the leading-edge plane where the span-wise distributions of V and p_o are known. Here, the streamwise and normal components of vorticity are calculated at each grid point using Equations (3.16) and (3.17). It now becomes possible to solve the secondary stream function equation (Eqn. (3.6)) and obtain the secondary velocity components at the passage inlet. Details of the method used for solving Equation (3.6) are given in Section 4.1.

Step 4: In order to compute the convection of streamlines from inlet to exit of the first CR, it is necessary to know the secondary velocity components at the CR exit plane. The exit distribution of secondary vorticity must first be known but Equation (3.12) cannot be used because the total pressure distribution through the CR is unknown. Since the flow turning angle is small through the CR, Equation (3.13) can be used to give a first approximation of the distributed secondary vorticity at the CR exit plane. Equation (3.6) can then be used to find the approximate secondary velocity components.

Step 5: With secondary velocities known at inlet and estimated at exit plane of the CR, the convection of streamlines can be computed through each sub-region with the use of Equations (3.20.a) and (3.20.b). In the following discussion, the inlet plane to a sub-region will be referred to as plane 1 (intermediate plane 1) while the exit plane to a sub-region will be referred to as plane 2 (intermediate plane 2). The secondary velocity distributions (derivatives of stream function) at planes 1 and 2 are found by simple linear interpolation (using axial distances) of those values found at the inlet and exit planes of the CR. Once the streamlines are convected through the first sub-region they no longer form a rectangular grid of points upon intersection with plane 2. The secondary velocity distributions at planes 1 and 2, however, are given at the points of a rectangular grid and therefore must be interpolated onto the now distorted grid points. The method used to interpolate properties from a rectangular to non-rectangular grid is described in Section 4.2. As mentioned before, Equations (3.20.a) and (3.20.b) must be used iteratively in order to find the final streamline intersection points on plane 2. Figure 3.7 illustrates how a hypothetical streamline intersection point on plane 2 converges to its final position after three iterations.

Figure 3.7 Iterating to find the streamline intersection point on intermediate plane 2 (k is the iteration number).

Step 6: The reason for computing the convection of streamlines through the CR is to evaluate the total pressure field at each intermediate plane and numerically integrate Equation (3.12) along each streamline. This gives a more accurate distribution of secondary vorticity (and thus secondary velocity components) at the CR exit plane. Total pressure is considered to be conserved along streamlines. In order to obtain accurate approximations of the total pressure gradients at the intermediate planes, two bivariate interpolation procedures must be followed. First the irregularly distributed values of total pressure at the intermediate plane must be interpolated onto points forming a rectangular grid. The procedure for doing this is described in Section 4.3. Once total pressure is known on a rectangular grid, numerically approximating $\partial p_c / \partial z$ is a simple task. The values of $\partial p_c / \partial z$ are then interpolated back onto the irregularly distributed points of intersection of the streamlines and the intermediate plane. This is the technique used by Gregory-Smith and Okan (1991); although the author is unaware of the methods they used for interpolation.

Step 7: The computation proceeds one sub-region at a time until the exit plane of the CR is reached. Equation (3.12) is now evaluated along each streamline to give a second, more accurate, estimation of the distributed secondary vorticity at the CR exit plane. This vorticity, however, is given at grid points which are irregularly distributed. Before they can be used in Equation (3.6), the values of secondary vorticity must be interpolated onto a rectangular grid. Evaluation of Equation (3.6) gives a second, more accurate, approximation of the secondary velocity components at the exit to the CR. Axial velocity and total pressure (which are considered to be constant along streamlines) must also be interpolated onto the rectangular grid at the CR exit plane in order to be used as input to the next CR.

It should be noted that steps 5 to 7 can now be used in an iterative loop using Equation (3.12) to calculate the distributed secondary vorticity. However, if a sufficient number of computational regions is used, such that the flow angle turned in each is small, then only the two iterations of the region are required for accurate results.

Step 8: The distributions of total pressure and three velocity components are now known on a rectangular grid of points at the exit plane of the CR. If the exit plane to the blade passage has not been reached these properties are transferred to the inlet of the next CR. The streamwise

component of vorticity is known at the next CR inlet plane but the normal component must be calculated using Equation (3.18).

Step 9. Steps 4 to 8 are repeated until the exit of the blade passage is reached.

SECONDARY FLOW COMPUTATION: NUMERICAL PROCEDURES

4

This chapter gives specific details of some of the numerical procedures used in the secondary flow computation (program BLADE). The method used to solve the secondary stream function equation (Eqn. 3.6), derived in Section 3.2, is discussed including the specification of boundary and initial conditions. As well, the details of two bivariate interpolation techniques, used in the secondary flow computation, are presented.

4.1 Solution of the Secondary Stream Function Equation

In order to compute the secondary flow velocities at the exit plane of each computational region, the secondary stream function equation,

$$\frac{\partial^2 \psi}{\partial y^2} + \cos^2 \alpha \frac{\partial^2 \psi}{\partial z^2} = \xi_z \cos \alpha, \quad (3.6)$$

must be solved for ψ at each grid point of a rectangular grid. This is done via subroutine SOLPSI. Now, the spatial coordinates, y and z , are known at each grid point as are the streamwise vorticity, ξ_z , and the primary flow angle, α . Applying the above equation at point i,j , as illustrated in Figure 4.1, and replacing the derivatives with second-order accurate three-point central-difference formulae which allow for non-uniform grid spacing in the z -direction (see Appendix D) gives

$$\frac{\psi_{i-1,j} - 2\psi_{i,j} + \psi_{i+1,j}}{(\Delta y)^2} + \cos^2 \alpha \frac{2[c\psi_{i,j-1} - (c+1)\psi_{i,j} + \psi_{i,j+1}]}{c(c+1)(\Delta z_{i,j})^2} = \xi_{z,i,j} \cos \alpha,$$

where c is the local grid spacing ratio in the z -direction. As explained in Appendix B, this local grid spacing ratio is different for the upper and lower halves of the blade passage. The reader is

Figure 4.1 Grid points employed in the Point Successive Over Relaxation iteration formula.

referred to Appendix B for further details regarding spanwise grid spacing. Now, letting

$$\beta_{i,j} = \cos\alpha \left(\frac{\Delta y}{\Delta z_{i,j}} \right)$$

and rearranging gives

$$\psi_{i,j}^{k+1} = \frac{1}{2 \left(1 + \frac{\beta_{i,j}^2}{c} \right)} \left[\psi_{i+1,j}^k + \psi_{i-1,j}^{k+1} + \frac{2\beta_{i,j}^2}{(c+1)} \left(\frac{1}{c} \psi_{i,j+1}^k + \psi_{i,j-1}^{k+1} \right) - \xi_{i,j} \cos\alpha (\Delta y)^2 \right], \quad (4.1)$$

where k is the iteration number. This is called the Point Gauss-Seidel Iteration formula for solving Equation (3.6). As the name implies, initial values of ψ are guessed and Equation (4.1) is used iteratively at each grid point (except those at the boundaries) until changes in ψ become very small in accordance with some convergence criterion. It is possible to obtain faster convergence by damping the changes in ψ such that Equation (4.1) becomes

$$\psi_{i,j}^{k+1} = (1-\omega)\psi_{i,j}^k + \frac{\omega}{2 \left(1 + \frac{\beta_{i,j}^2}{c} \right)} \left[\psi_{i+1,j}^k + \psi_{i-1,j}^{k+1} + \frac{2\beta_{i,j}^2}{(c+1)} \left(\frac{1}{c} \psi_{i,j+1}^k + \psi_{i,j-1}^{k+1} \right) - \xi_{i,j} \cos\alpha (\Delta y)^2 \right],$$

where ω is the relaxation factor and $0 < \omega < 2$. This is called the Point Successive Over Relaxation (PSOR) iteration formula for solving Equation (3.6) and was used in the present study. Hoffmann (1989) illustrates the use of both the above iteration formulae; albeit for an expression which is slightly simpler than Equation (3.6).

4.1.1 Estimating the Optimum Relaxation Factor

The number of iterations required for convergence can be minimized by finding the optimum value of ω . Hoffmann (1989) gives the optimum value of ω for a rectangular domain with constant step sizes (grid spacing) as

$$\omega_{opt} = \frac{2 - 2\sqrt{1-a}}{a}, \quad (4.2)$$

where

$$a = \left[\frac{\cos\left(\frac{\pi}{IMAX-1}\right) + \beta^2 \cos\left(\frac{\pi}{JMAX-1}\right)}{1 + \beta^2} \right]^2$$

and $IMAX$ and $JMAX$ are the maximum values of i and j , respectively. However, since Equation (3.6) must be solved on a grid with non-uniform step size, the value of β^2 in the above expression is estimated using

$$\beta^2 = \cos^2 \alpha \left(\frac{\Delta y}{\overline{\Delta z}} \right)^2$$

in the present study, where $\overline{\Delta z}$ is an average of all Δz in the grid.

4.1.2 Boundary and Initial Conditions

The boundary conditions on the borders of the rectangular grid simply dictate that fluid velocities perpendicular to these boundaries must be zero. Thus, at the hub and casing surfaces, w , must be zero and at the blade surfaces, v , must be zero. Referring to Equations (3.7.a) and (3.7.b) it can therefore be seen that ψ must be constant along the boundaries. For simplicity, the boundary condition

$$\psi_{boundary} = 0$$

is applied at all borders of the grid.

The initial values of ψ at the interior grid points can be arbitrary. However, the further the initial values are from their converged values, the greater the number of iterations required to reach the converged solution. At the outset of the secondary flow computation (at the leading-edge plane) the value of ψ at each grid point is given a value of zero. As the computation

proceeds from one computational region to the next down the blade passage, however, the initial values of ψ are set to those obtained from the converged solution at the exit plane of the previous computational region. This greatly reduces the number of iterations required to reach a converged solution at any one exit plane to a computational region.

4.2 Bivariate Interpolation on a Rectangular Grid

This section describes the method used to interpolate values, of the form $P(y,z)$, from a rectangular grid onto a non-rectangular grid. What is meant by a rectangular grid is a structured set of points in the y - z plane for which every set of four adjacent points forms a rectangle. Here, a non-rectangular grid is one which is unstructured. The (y,z) points on the unstructured grid cover the same domain (e.g. an intermediate plane) as the structured grid but are irregularly distributed about this area. This form of bivariate interpolation is required in order to determine the secondary flow velocities and spanwise total pressure gradient at points where streamlines intersect intermediate planes, as discussed in steps 5 and 6, respectively, of the solution procedure presented in Section 3.5.

Bivariate interpolation of the type described above can be performed in many different ways giving various degrees of accuracy and surface smoothness (see, for example, Press et al. (1992) or Akima (1974)). The method used in the present study is, perhaps, the simplest. It is referred to by Press et al. (1992) as bilinear interpolation since the surface function $P(y,z)$ is considered to vary linearly between points in the two spatial coordinate directions. The surface produced by this method is not smooth but is continuous. This method was used solely on the basis of its simplicity so that execution times would still be reasonable given that the interpolation must be performed twice on every intermediate plane.

If the point (y,z) lies inside the rectangle produced by points (y_1,z_1) , (y_2,z_1) , (y_2,z_2) , and (y_1,z_2) , as illustrated in Figure 4.2., then the value of $P(y,z)$ is given by

Figure 4.2 Grid points used in bilinear interpolation of $P(y, z)$.

$$P(y,z) = P_4 + \frac{y-y_1}{y_2-y_1}(P_3-P_4) + \frac{z-z_1}{z_2-z_1} \left[P_1 - P_4 + \frac{y-y_1}{y_2-y_1}(P_4-P_1+P_2) \right],$$

where $P_1=P(y_1,z_2)$, $P_2=P(y_2,z_2)$, $P_3=P(y_2,z_1)$, and $P_4=P(y_1,z_1)$. The value of P for each (y,z) point on the non-rectangular grid is found by first locating the four points on the rectangular grid which surround the point at which P is to be found and then applying the above expression. This is the technique used in subroutine BLI. Interpolation in the opposite direction (from unstructured grid to rectangular grid) is a much more difficult procedure as will be discussed in the following section.

4.3 Bivariate Interpolation on an Unstructured Grid

This section describes the method used, in the secondary flow computation, to interpolate values of the form $P(y,z)$ in a rectangular domain filled with data points at which P is known but which are irregularly distributed in the y - z plane. The goal is to obtain values of P at points which form a rectangular grid. This type of bivariate interpolation is necessary in order to more easily calculate the distribution of spanwise total pressure gradient on each intermediate plane, as outlined in Step 6 of the secondary flow computation procedure presented in Section 3.5. It is also required in Step 7 where, at the exit plane of a computational region, the properties along streamlines (total pressure, streamwise vorticity, and axial velocity) must be known on a rectangular grid to serve as input to the next computational region.

Bivariate interpolation among a set of irregularly distributed data points is far from a simple task and the author is not aware of any numerical methods texts which deal with this type of problem in a practical way. There are two types of approaches possible, one using a single function over the entire domain, and the other based on a collection of local functions. The first approach involves fitting a surface to all of the data points but, as the number of data points increases, this procedure becomes too complicated to manage. Furthermore, the resulting surface will sometimes exhibit spurious oscillations. It was for these reasons that only methods based on local functions were considered for the present study.

2

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

This type of bivariate interpolation sees applications in many fields but most notably in geology where it is used in contouring. Many methods have been developed for this purpose, each having specific merits and drawbacks. It should be emphasized that there is little theoretical basis for any of the methodologies currently in use. Instead, most are founded on commonsense assumptions about the way in which surfaces behave and the results of practical experience. A review of the more popular interpolation methods is given in the following sub-sections.

4.3.1 Triangulation Methods

This group of interpolation techniques involves, as a first step, the division of the y - z plane into a number of triangles (creating a triangular mesh), the vertex of each triangle being the projection of a data point, $P(y,z)$, onto the y - z plane. It is obvious that, for a given set of data points, there are a number of ways in which the y - z plane can be divided, creating different meshes. To avoid this problem, attempts were made to create a unique "optimal" triangular mesh. Some criteria suggested by Gold et al. (1977) were that the individual triangles should be as near to equilateral as possible, or that each triangle should have the minimum variation in P as possible, or that the longest side of each triangle be as short as possible. Unfortunately, the earlier triangulation techniques relied on iteration to obtain an optimum configuration of triangles; requiring excessive computation times.

Many of the more recent algorithms (Gold et al., 1977; McCullagh and Ross, 1980) are capable of producing an almost optimal triangular mesh on a single pass through the domain. The meshes produced by these algorithms are referred to as *Delaunay triangulations* and are uniquely defined for a given set of points. One can imagine, in a field of irregularly distributed points, that each point is surrounded by an irregular polygon such that every location inside a polygon is closer to the enclosed point than to any other. This is illustrated in Figure 4.3.a and represents the most compact division of space possible. Sets of such polygons are often referred to as Thiessen, Dirichlet, or Voronoi polygons. A familiar example of such a division of space occurs in a cluster of soap bubbles. Now, if the points in neighbouring polygons are connected to point A, as illustrated in Figure 4.3.b, the result is a Delaunay triangular network. Both polygons and triangles are unique for any given set of points. Although this process of triangulation is

Figure 4.3.a Thiessen polygon (shaded) around point A. All locations within polygon are closer to A than to any other point (reproduced from Davis, 1986).

Figure 4.3.b Delaunay triangle network around point A. All connected points are Thiessen neighbours of A (reproduced from Davis, 1986).

complicated (see Davis, 1986). McCullagh and Ross (1980) state that the number of steps necessary to triangulate a set of n points is proportional to $n \log n$, whereas the number of operations required to produce a triangular mesh by trial and error may be proportional to n^4 .

Once the y - z plane has been divided into a triangular mesh, the next step is to fit some type of surface to each triangle. Obviously, the simplest type of surface to fit would be a plane. This was the approach taken by Bengtsson and Nordbeck (1964) and Heap (1972). The resulting surface is not smooth over the domain but is continuous.

Whitten and Koelling (1975) attempted to fit curved surfaces to the triangles. Their function for each triangle had fifteen coefficients to be determined by fifteen conditions. These conditions, however, could not always be satisfied and even if they could, the resulting surface was not guaranteed to be smooth.

Zlamal (1968) discussed a local surface fitting procedure using fifth-degree bivariate polynomials for each triangle in the y - z plane. In order to determine the coefficients, he uses the P values and approximations of the first and second partial derivatives of P at the vertices of each triangle. In addition, the normal derivatives at the midpoints of the sides of each triangle are necessary. This theory was later generalized, by Zenisek (1976), to $(4m+1)$ th-degree polynomials for functions m -times continuously differentiable on a closed triangular domain.

Akima (1978) also proposed a method in which a bivariate fifth-degree polynomial was fit to each triangle. Determination of the twenty-one coefficients necessary for each triangle was accomplished in much the same manner as Zlamal (1968). Akima (1978), however, presented substantially better algorithms for triangulation in terms of required storage area and computation times.

4.3.2 Inverse-Distance Methods

The algorithms of inverse-distance methods are considerably simpler than those of triangulation methods. These methods are based on searching the area about the point at which

P is to be interpolated (point (y,z)), and basing this interpolated value on the P values at neighbouring points. The P values from neighbouring points are weighted by a function of distance to point (y,z) according to

$$w_i = D_i^{-e} ,$$

where D is the distance from the neighbouring point to (y,z) and e is an exponent which typically ranges from 1 to 5. In terms of this weighting function, the interpolated value of P at (y,z) is

$$P(y,z) = \frac{\sum_{i=1}^n w_i P_i}{\sum_{i=1}^n w_i} ,$$

where n is the number of neighbouring points and P_i is the value of P at each of these points.

The most time consuming component of inverse-distance algorithms is that of selecting the neighbouring points. Selection criteria vary widely and are usually left up to the user. It is possible to choose all points in the domain as the neighbouring points. A more usual criterion is to select the n closest points to the point being interpolated (point (y,z)). However, as shown in Figure 4.4.(a), these n closest points may lie to one side of (y,z) ; thereby giving an interpolated value which may be inaccurate. To ensure equitable distribution of the selected neighbouring points, a quadrant search pattern may be used. This technique, illustrated in Figure 4.4. (b), finds the $n/4$ closest points in each quadrant. A further extension of this technique, the octant search, finds the $n/8$ closest points in each octant, as illustrated in Figure 4.4.(c).

The first inverse-distance method considered by the author was that of Smallwood (1992). He suggested that velocity vectors could be interpolated at specified locations (on a uniformly spaced grid) throughout a randomly seeded flowfield using a simple inverse-distance weighting ($e=1$) of velocities at known locations. Only those points falling inside a user-specified distance of the desired location would be considered in the interpolation. Smallwood suggested that this user-specified distance be at least equal to the uniform grid spacing and that data smoothing could be achieved by increasing this distance.

Figure 4.4 (a) Search technique that locates n nearest neighbours around point (y,z) . (b) Quadrant search pattern used to insure equitable distribution of neighbouring points. (c) Octant search pattern around point (y,z) (adapted from Davis, 1986).

The method of Shepard (1968) was similar to that of Smallwood (1992) but instead of using a simple inverse-distance weighting for the selected neighbouring points Shepard used an inverse-distance squared weighting ($e=2$). Due to the inverse-square weighting, this method fails to produce a plane when all the given data points lie in a slanted plane. This is considered a drawback of all inverse-distance methods with $e>1$.

4.3.3 Kriging

Kriging is perhaps the most sophisticated method of bivariate interpolation among irregularly distributed data. The method is named after D.G. Krige, a South African mining engineer and pioneer in the application of statistical analysis to mine evaluation. Kriging works in much the same way as the inverse-distance methods described in the previous section. In a kriging algorithm, however, the weighting factors and selected neighbouring points are determined statistically in order to produce interpolated values with the minimum estimation error. The mathematics of the kriging technique are quite involved and will not be given here. The reader is referred to Clark (1979) or Davis (1986) for a more thorough explanation of kriging.

4.3.4 Present Method

In developing an interpolation technique for the secondary flow computation, simplicity (which translates into shorter run times) was considered paramount due to the sheer number of times that the interpolation technique is repeated throughout the secondary flow computation. Of course, reasonable accuracy was also desired.

The author's experience with inverse-distance methods has shown that the user-specified values of the weighting exponent, e , and the number of neighbouring points considered, n , significantly affects the outcome of the secondary flow calculation. This may very well be due to the cumulative effects of repeated interpolations. The method of Smallwood (1992), in which the neighbouring points are those falling inside a user-specified distance of the point being interpolated, was also considered. This circular area was dubbed the "circle of influence" in the

present study; its radius being termed the "radius of influence". In some instances no points or only a single point was found to lie inside the circle of influence when its radius was set equal to the grid spacing of the desired uniform rectangular grid (as suggested by Smallwood, 1992). Using only the single point causes data smoothing while encountering no points forces the user to increase the radius of influence. This raised questions concerning how large the radius of influence should be and what should be an acceptable minimum number of points found to lie inside the circle of influence. A parametric study revealed that varying both of these parameters independently also caused results to differ significantly.

The above-mentioned properties of the inverse-distance methods made them somewhat unattractive for use in the secondary flow computation. Triangulation of the y - z plane was avoided on the basis of its complexity and the technique of kriging was never considered because of this reason also. The interpolation method developed for the present study (subroutine INTERP) was a combination of some of those methods discussed previously and is presented next.

Unlike the methods described in Section 4.3.1, the present method avoids the complete triangulation of the y - z plane before the required points are interpolated. Instead, the present technique uses a local triangulation procedure. As illustrated in Figure 4.5, the area around the point at which the interpolation is to be performed (point (y,z)) is divided into quadrants and the closest point to (y,z) in each quadrant is found (a quadrant search). In searching for these four closest points, only those points within a certain distance of point (y,z) are tested. This distance is set sufficiently large so that at least one point lies in each quadrant. Finding four points in this manner guarantees that point (y,z) lies inside two of the four possible triangles that can be formed by the four points. In the case shown in Figure 4.5, point (y,z) lies inside the triangle formed by points (y_1,z_1) , (y_2,z_2) , and (y_4,z_4) , and the triangle formed by points (y_1,z_1) , (y_3,z_3) , and (y_4,z_4) . Planes are passed through each of these sets of three data points and the value of $P(y,z)$ is calculated on both planes. The interpolated value of $P(y,z)$ is taken to be the average of those calculated on each plane. This averaging technique produces a smoother surface than that created when the entire domain is divided into triangles and fitted with planes before interpolating. Finding $P(y,z)$ on each plane is described in the following.

Given three points $P(y_1,z_1)$, $P(y_2,z_2)$, and $P(y_3,z_3)$, the projections of which in the y - z plane

Figure 4.5 Closest point in each quadrant surrounding point to be interpolated; $P(y, z)$.

surround point (y,z) , the value of $P(y,z)$ is given by

$$P(y,z) = \frac{-(d + ay + bz)}{c} ,$$

where the coefficients a , b , c , and d are given by

$$a = u_2 v_3 - u_3 v_2 ,$$

$$b = u_3 v_1 - u_1 v_3 ,$$

$$c = u_1 v_2 - u_2 v_1 ,$$

$$d = -(ay_1 + bz_1 + cP(y_1, z_1)) ,$$

and

$$u_1 = y_2 - y_1 , \quad v_1 = y_3 - y_1 ,$$

$$u_2 = z_2 - z_1 , \quad v_2 = z_3 - z_1 ,$$

$$u_3 = P(y_2, z_2) - P(y_1, z_1) , \quad v_3 = P(y_3, z_3) - P(y_1, z_1)$$

ANNULAR DUCT FLOW COMPUTATION: THEORY

5

Although many computational techniques have been developed to predict flows through turbomachines (see Section 2.2), the method chosen for the annular duct flow computation of program DUCT is of the streamline curvature type. This chapter describes the flow model used along with the simplifying assumptions made in its development. The streamline curvature computation method is described along with the derivation of the governing equations. It should be noted here that the method described does not permit any radial mixing of flow properties and is referred to as the basic method. Modifications to this basic method, which permit spanwise mixing, are presented in Chapter 6. Also in this chapter, a brief procedure for solving the governing equations is outlined and some problematic aspects of the streamline curvature method are discussed. Specific details of the solution procedure are presented in Chapter 7.

5.1 Flow Model

To follow the formally correct approach introduced by Wu in 1952 a full, though approximate, analysis of the annular flow is performed separately on two sets of intersecting surfaces. The first set are surfaces of revolution about the annular axis, previously referred to as S_1 surfaces. Since the S_1 surfaces are streamsurfaces they will actually twist as they pass through blade rows, however, in annular duct regions they will quickly return to being axisymmetric. The second set of surfaces, previously referred to as S_2 surfaces, are those which extend from hub to casing and are also streamsurfaces. The intersection of the S_1 and S_2 surfaces form a set of streamlines in the annulus. Although this analysis is applied to the entire volume of the annulus, it is still considered quasi-three-dimensional since the flow is treated as two-dimensional on the each of the S_1 and S_2 surfaces.

The model commonly adopted, and is done so here, is one in which the flow is assumed to be circumferentially uniform ($\partial/\partial\theta=0$) so that only a single S_2 surface is required to describe the flow. This single S_2 surface is sometimes referred to as the mean hub-to-casing streamsurface, in that at any given axial and radial location on it the fluid properties are equal to their circumferential averages. This is a somewhat loose definition of the mean streamsurface, but as Marsh (1976) pointed out, a rigorous definition does not appear to be available, particularly for flow through bladed regions. However, the mean streamsurface approximation has been shown to be generally adequate for turbomachinery flows and is expected to be even better for annular duct flow due to the near axisymmetric nature of the S_1 surfaces.

Before the calculations can be performed it is traditional to project the velocities of the mean S_2 surface onto another surface called the meridional plane. The meridional plane contains the x and r axes and is normal to the θ -direction. It is generally not a streamsurface but can be so in the instance where all tangential velocities are zero (the flow contains no swirl). The meridional plane for a hypothetical diverging annulus is illustrated in Figure 5.1 along with some axisymmetric S_1 surfaces which have been sectioned for clarity. The axisymmetric streamsurfaces form a group of concentric streamtubes which intersect the meridional plane to produce a number of so-called meridional "streamlines". By definition these are not actually streamlines since they have tangential velocities associated with them, but for the purposes of this discussion they will be referred to as meridional streamlines or simply as streamlines.

Several methods exist for calculating the flow on the meridional plane, the most popular being the streamline curvature method and the so-called matrix through-flow method. While these two methods are quite different computationally, the level of flow approximation is essentially equivalent and there is little to choose between them in terms of relative advantages. Historically, the streamline curvature method seems to be the more popular of the two. Reasons for this popularity may lie in the streamline curvature method's ability to better cope with regions of transonic flow, although some difficulties in this area still arise. Other reasons for its popularity, over the matrix method, may be due to the more intuitive nature of the governing equations used and the simpler programming involved in its implementation. A comprehensive comparison of the two methods by Davis and Millar (1975) concluded that there was a marginal advantage in using the matrix through-flow method on the grounds of greater solution stability and accuracy.

Figure 5.1 The meridional plane for a hypothetical diverging annulus.

In the present study, the streamline curvature method was used to compute annular duct flows on the meridional plane. There were several reasons for favouring this method over the matrix through-flow method including the availability of open literature, the simplicity of programming, and the streamline curvature method's ability to directly accommodate the radial mixing theory presented in Chapter 6. The remainder of this chapter is devoted to describing the streamline curvature method including the derivation of the governing equations and some discussion of stability considerations and Mach number limitations.

5.2 Streamline Curvature Method

This section describes the meridional flow field calculation using the streamline curvature method. Here the governing equations are derived and a brief eight step procedure for their solution is presented. Solution stability and Mach number limitations are also discussed.

To obtain a set of reasonably simple equations which can be used to describe the meridional flow field, certain simplifying assumptions must be made. As described in the previous section, the flow in the annulus is considered to be axisymmetric, that is, that there are no fluid property gradients in the circumferential direction. The flow is also assumed to be steady and inviscid which is a reasonable approximation for a circumferentially averaged annular flow with thin boundary layers. The fluid is considered to be an ideal gas and compressibility effects are taken into account, although shocks are not permitted. Section 5.2.4 deals with some of the difficulties that transonic flow presents when using the streamline curvature method.

As indicated before, this chapter only presents the basic streamline curvature method, which does not allow for any radial mixing of flow properties. Thus, the flow in this basic method is assumed to be isentropic (in duct regions), where the flow in each axisymmetric streamtube is adiabatic and lossless. This isentropic flow assumption will be relaxed when the radial mixing theory is presented in Chapter 6.

Given the above simplifying assumptions a workable set of equations governing the flow in the meridional plane can be derived as in the following section.

5.2.1 Governing Equations

The principle of the streamline curvature method is to first write the momentum equation in the meridional plane in terms of the radial pressure gradient and the slope and curvature of the meridional streamlines. Throughout the annulus the momentum equation is applied in directions which are roughly perpendicular to the streamlines and which remain fixed during the calculation. These directions are commonly referred to as quasi-orthogonals (QOs) and are set out in advance of the calculation. Figure 5.2 shows a hypothetical QO (indicated by the q -direction) and meridional streamline. QOs are most simply placed in the radial direction; however, for calculations involving blade rows it is desirable to place them parallel to the leading and trailing edges of the blades.

In developing the desired form of the momentum equation in the q -direction it is necessary to start with Euler's equation for inviscid flow, which in vector form is

$$\frac{DV}{Dt} = -\frac{\nabla p}{\rho} + F.$$

The force term, F , takes into account any external body forces, such as gravity or a blade force, which may be acting on the fluid. For an ideal gas, the second law of thermodynamics can be expressed in vector form as

$$T \nabla S = \nabla h - \frac{\nabla p}{\rho}$$

or as

$$T \nabla S = \nabla h_o - \nabla \left(\frac{1}{2} V^2 \right) - \frac{\nabla p}{\rho}.$$

realizing that

$$h = h_o - \frac{V^2}{2}.$$

Expressing the pressure gradient term in Euler's equation in terms of the second law equation and taking components in the q -direction gives

Figure 5.2 Notation for meridional plane.

$$T \frac{\partial S}{\partial q} = \frac{\partial h_o}{\partial q} - \frac{\partial}{\partial q} \left(\frac{1}{2} V^2 \right) + \frac{Dv_q}{Dt} - F_q \quad (5.1)$$

Referring to Figure 5.2, it can be seen that the acceleration of a fluid particle at point P on the meridional streamline, when resolved in the q -direction, is given by

$$\frac{Dv_q}{Dt} = -\frac{v_o^2}{r} \sin(\phi + \zeta) + \frac{v_m^2}{R_c} \sin \phi + \frac{Dv_m}{Dt} \cos \phi$$

in the (r, θ, x) coordinate system. The meridional velocity component, v_m , is the velocity in the meridional plane and is the vector sum of the axial and radial velocity components. The first and second terms on the right-hand-side of the above expression represent centrifugal accelerations about the annular axis and the streamline centre of curvature respectively. The last term represents linear acceleration along the streamline in the m -direction. Now, substitution of this into Equation (5.1), expressing the velocity as

$$V^2 = v_m^2 + v_o^2$$

and rearranging gives

$$\frac{\partial \left(\frac{1}{2} v_m^2 \right)}{\partial q} = \frac{\partial h_o}{\partial q} - T \frac{\partial S}{\partial q} - \frac{1}{2} \frac{\partial (v_o^2)}{\partial q} - \frac{v_o^2}{r} \sin(\phi + \zeta) - F_q + \frac{v_m^2}{R_c} \sin \phi + v_m \frac{\partial v_m}{\partial m} \cos \phi \quad (5.2)$$

since the substantial derivative

$$\frac{Dv_m}{Dt} = v_m \frac{\partial v_m}{\partial m}$$

for steady axisymmetric flow. Also, from geometry it is easily seen that

$$\sin(\phi + \zeta) = \frac{\partial r}{\partial q}$$

so that the third and fourth terms on the right-hand-side of Equation (5.2) can be rewritten as

$$-\frac{1}{2} \frac{\partial(v_0^2)}{\partial q} - \frac{v_0^2}{r} \sin(\phi + \zeta) - \frac{1}{2r^2} \frac{\partial(r^2 v_0^2)}{\partial q} .$$

Substitution of this into Equation (5.2) gives

$$\frac{\partial\left(\frac{1}{2} v_m^2\right)}{\partial q} = \frac{\partial h_o}{\partial q} - T \frac{\partial S}{\partial q} - \frac{1}{2r^2} \frac{\partial(r^2 v_0^2)}{\partial q} - F_q + \frac{v_m^2}{R_c} \sin \phi + v_m \frac{dv_m}{dm} \cos \phi .$$

This equation is commonly referred to as the radial equilibrium equation (REE) and is the basis of all streamline curvature through-flow computation methods. It is usual to express the last term on the right-hand-side of the REE in terms of the rate of divergence of the streamlines and the local meridional Mach number (see, for example, Novak (1967)). However, as Denton (1978) points out, this introduces considerable additional complexity into the equation and produces a singularity when the local meridional Mach number equals unity. Denton has shown that dv_m/dm can be adequately approximated numerically so that the REE can be used directly in the form as presented above.

The REE can be simplified for use in an annular duct since there are no blades present. The force term can be dropped and the q -direction can be made to correspond with the r -direction by placing all of the QOs radially. The REE can then be written

$$\frac{\partial\left(\frac{1}{2} v_m^2\right)}{\partial r} = \frac{\partial h_o}{\partial r} - T \frac{\partial S}{\partial r} - \frac{1}{2r^2} \frac{\partial(r^2 v_0^2)}{\partial r} + \frac{v_m^2}{R_c} \sin \phi + v_m \frac{dv_m}{dm} \cos \phi . \quad (5.3)$$

The last term on the right-hand-side is often neglected when the REE is used in calculating compressor flows. In turbine flows, however, this term becomes more significant due to the greater annulus taper generally found in turbines. As such, this term will be retained here. Equation (5.3) gives the gradient of meridional velocity along any QO.

In the basic streamline curvature method (no mixing) angular momentum is conserved along streamlines for annular duct flow. This can be shown by considering the tangential component of Euler's equation,

$$\rho \left(\frac{Dv_\theta}{Dt} + \frac{v_r v_\theta}{r} \right) = \rho F_\theta .$$

Since the force term is zero, the substantial derivative

$$\frac{Dv_\theta}{Dt} = v_m \frac{\partial v_\theta}{\partial m}$$

for steady axisymmetric flow and

$$v_r = v_m \frac{\partial r}{\partial m} ,$$

so that the tangential momentum equation can be written

$$\frac{\partial}{\partial m} (r v_\theta) = 0 \quad (5.4)$$

after simplification.

For a given mass flow rate, $\dot{m}$, a form of the continuity equation for an annular flow can be expressed as

$$\dot{m} = \int_{h_{uh}}^{r_{out}} 2\pi r \rho v_m \sin\phi (1-b) dr \quad (5.5)$$

where b is called the blockage factor and represents the fraction of the annulus cross sectional area blocked by hub and casing boundary layers. Within blade rows, blade thickness and blade boundary layers are also included in b . For an inviscid duct flow, however, the blockage factor equals zero.

Flow along streamlines is also considered to be isentropic in the basic method so that the steady flow energy equation takes the form

$$\frac{\partial h_e}{\partial m} = 0 . \quad (5.6)$$

Since the flow is adiabatic and lossless it is also implicit that entropy, total pressure, and total

temperature are conserved along streamlines. Thus,

$$\frac{\partial S}{\partial m} = 0 . \quad (5.7)$$

An equation of state is used to close the system. Since the flow is considered to be an ideal gas, the ideal gas law,

$$p = \rho RT , \quad (5.8)$$

is used. Equations (5.3) to (5.8) form a system of partial differential-integral equations for which a special technique must be used to solve. This technique, commonly referred to as the streamline curvature method, is described in the following section.

5.2.2 Solution Procedure

This section presents a somewhat brief step-by-step procedure for solving the governing equations developed in the previous section. Specific details of each step are presented in various sections of Chapter 7.

The governing equations of Section 5.2.1 are solved at the intersection points of the meridional streamlines and the quasi-orthogonals (QOs). In physical space this generally forms a non-rectangular grid of points in the meridional plane with unequal grid spacing in both the radial and streamwise directions. The computational domain is directly superimposed on the physical domain so that transformations of the coordinate variables are not required. The irregularly shaped computational grid poses no particular difficulties when solving the governing equations. QOs placed at the inlet and exit planes of the annulus as well as the hub and casing boundary streamlines form the boundaries of the grid. If hub and casing boundary layers are to be considered, then the boundary streamlines will be displaced from the hub and casing surfaces. Otherwise, the boundary streamlines will coincide with the hub and casing surfaces. Physical and computational domains for a diverging annulus are illustrated in Figure 5.3.

Physical Domain

Computational Domain

Figure 5.3 Physical and computational domains for a diverging annulus.

Minimal input is required to start the computation. Annulus geometry, defined by QO endpoint coordinates, must be specified along with the total mass flow rate and inlet conditions. The inlet conditions consist of radial distributions of stagnation temperature, stagnation pressure, and swirl angle spanning from hub to casing.

It should be noted that the streamline curvature method is a rather specialized iterative computational technique and many different approaches to solving the governing equations exist. The following steps illustrate the approach used in the program, DUCT, developed for the present study.

Step 1: In order to start the computation, some initial state of the flow field properties must be guessed at including the initial placement of streamlines. Details of this initialization are given in Section 7.1.

Step 2: With all properties of the meridional flow field known from either initialization or the previous iteration, the radial equilibrium equation (Eqn. 5.3) is used to update the meridional velocity distribution starting at the inlet QO. The radial equilibrium equation is treated as an ordinary differential equation and is integrated along the QO using the value of v_m at the hub, from the previous iteration, as an initial value. This gives an updated distribution of v_m along the QO but does not yield the correct mass flow rate across the QO.

Step 3: The continuity equation (Eqn. 5.5) is evaluated to give the mass flow rate crossing the QO. If this mass flow rate is not within a certain fraction of the specified mass flow rate, the difference is used to adjust the v_m distribution by a constant component of axial velocity making the mass flow crossing the QO equal to that specified. This results in an updated value of v_m at the hub which can again be used as an initial value in integrating the radial equilibrium equation along the QO.

Step 4: Steps 2 and 3 are repeated at the inlet QO until the difference between calculated and specified mass flow rates is within a certain tolerance. The same iterative procedure is applied to all QOs resulting in updated values of meridional velocity for the entire flow field. Details of the methods used in Steps 2 and 3 can be found in Section 7.3.

analogous to molecular shear stress. The total x -direction shear stress can now be represented by

$$\tau_{xy} = (\mu + \mu_t) \frac{\partial \bar{u}}{\partial y} ,$$

where μ_t is called the eddy viscosity. Unlike molecular viscosity, the eddy viscosity is not a fluid property but varies with flow conditions and geometry (ie. the state of the turbulent eddies).

In an analogous way, it is also possible to define an eddy thermal conductivity, k_t , which relates the heat transfer rate per unit area in the y -direction to the local mean temperature gradient in the fluid by

$$q_y = - (k + k_t) \frac{\partial \bar{T}}{\partial y} ,$$

where k is the molecular thermal conductivity of the fluid.

In the mixing model of Gallimore (1986) it is assumed that the radial mixing of heat and momentum can be represented by k_t and μ_t respectively and that k and μ are negligible since $k_t \gg k$ and $\mu_t \gg \mu$ in a turbulent free stream flow. Eddy viscosity and eddy thermal conductivity can be formed into a dimensionless ratio called the turbulent Prandtl number, given by,

$$Pr_t = \frac{\mu_t c_p}{k_t} . \quad (6.1)$$

The eddy viscosity can also be related to a parameter, ϵ , which describes the turbulent diffusivity of mass, using the turbulent Schmidt number,

$$Sc_t = \frac{\mu_t}{\rho \epsilon} . \quad (6.2)$$

Gallimore (1986) refers to ϵ as the mixing coefficient which has dimensions of length squared per unit time. This will be the terminology used in the present study.

The turbulent Prandtl number represents the ratio of turbulent momentum diffusivity to turbulent thermal diffusivity while the turbulent Schmidt number represents the ratio of turbulent momentum diffusivity to turbulent mass diffusivity. Since these three diffusivities are due to the

5.2.3 Stability Considerations

It is well known that the streamline curvature method's iterative procedures over-correct the initial errors and in order for the solution to be convergent some type of damping or relaxation must be used. Many factors are known to affect the solution stability including the local meridional Mach number and the type of curve fit used for the streamlines. Some discussion of these factors will be presented in later sections. This section presents several methods that have been used by others to maintain solution stability.

The traditional method used to stabilize the solution is to restrict changes in streamline position. Streamlines are adjusted by only a fraction of their predicted change for each iteration. A thorough study of streamline curvature solution stability for two-dimensional duct flow has been undertaken by Wilkinson (1970). He revealed that relaxation factors used for damping changes in streamline position or radius of curvature may range from 0.5 to 0.025 and are very case specific. He also revealed that the relaxation factor must be reduced as the distance between QOs is reduced in relation to the distance between hub and casing or as the Mach number rises. Wilkinson showed that the optimum relaxation factor is given by

$$\omega_{opt} = \frac{1}{1 + k A^2 (1 - M^2)} \quad , \quad (5.9)$$

where k is likely to be of order 0.2 and A denotes the grid aspect ratio; the distance between hub and casing divided by the distance between QOs. Although this expression was developed for two-dimensional duct flow several authors have extended its use to annular duct flow with bladed regions.

Denton (1978) suggests that faster convergence can be attained if changes in streamline slope and radius of curvature are damped rather than changes in streamline position. His suggestion, however, is based on experience in applying the streamline curvature method to annular flows with bladed regions and turbines in particular. Davis and Millar (1975) recommend that, along with damping changes in streamline position, changes in velocity be damped between iterations if they exceed certain bounds.

To maintain solution stability in the program developed for the present study the traditional method of damping changes in streamline position was used. Also, for simplicity, no attempt was made to optimize the relaxation factor. Given the speed of current computers, this should have little impact on program performance in terms of execution times.

5.2.4 Mach Number Limitations

Regions of transonic flow are frequently encountered in turbomachinery, especially in turbines. These regions are characterized by adjacent zones of subsonic and supersonic flow. It is well known that the equations for supersonic flow are hyperbolic in nature and no influence of downstream points is realized upstream. This presents difficulties for the elliptic nature of the streamline curvature method which uses central differences in approximating streamwise derivatives. Thus, the governing equations need to be of mixed hyperbolic-elliptic type for which separate finite-difference procedures are required. This presents enormous computational difficulties since it is not known a priori where the regions of subsonic and supersonic flow are located. Therefore, strictly speaking, the streamline curvature method must be restricted to use in subsonic flows if the solution is to be valid.

Despite this, small patches of transonic flow can usually be accommodated without much difficulty since the streamline curvature method considers continuity across the entire annulus. Also, regions of transonic flow are most likely to occur within blade passages so problems can usually be avoided by only placing quasi-orthogonals outside of blade rows. However, problems such as solution divergence, still arise when larger areas of transonic flow occur in duct regions or when a choked flow condition is approached. Several authors have overcome some of these problems with the use of careful programming techniques. Denton (1978) deals with the problems of transonic flow in a multi-stage low-pressure steam turbine but does not attempt a solution for fully choked flow. Novak (1976) gives a thorough account of the specific computational difficulties encountered when dealing with choked flow and describes a method for obtaining solutions in choked duct regions.

It is a fundamental assumption that, when using the streamline curvature method, there

exists a unique solution for the flow field. The lack of such a solution would be a prime cause of instability and divergence. Marsh (1971) examines this question of uniqueness for meridional flow field calculations using both the streamline curvature method and the matrix throughflow method. It is a usual practice, in the streamline curvature method, to update density along with meridional velocity in the same iteration, when solving the radial equilibrium equation (see Section 5.2.2). If this practice is followed, Marsh states that a sufficient condition for uniqueness of the solution is

$$M_m < 1, \text{ for duct flow, and}$$

$$M_{rel} < 1 \text{ at all radii or } M_{rel} > 1 \text{ for all radii, for flow within or behind a blade row,}$$

where M_m is the meridional Mach number and M_{rel} is the Mach number relative to the blade passage. This states that the solution for duct flow is only unique if the flow is entirely subsonic and that the solution within or behind a blade row is only unique if the flow is either entirely subsonic or entirely supersonic. If, however, the meridional velocity is solved for using values of density from the previous iteration, Marsh concludes that the resulting solution is always unique. The resulting lag in the calculation of density behind that of velocity has no effect on the converged solution. Smith (1970) used this technique in a matrix throughflow calculation and found that it improved solution stability at high Mach numbers. Others have found that it makes the matrix method usable up to Mach numbers of 1.2 although shocks are excluded from the solution.

In light of these findings, the streamline curvature method used in program DUCT incorporates this lagging density calculation in the hopes that it will improve stability.

Experimental evidence has suggested that substantial amounts of radial or spanwise mixing of properties can occur in axial-flow turbomachines (see Section 2.3.1). The mechanism responsible for this mixing in compressors has been the subject of great debate over recent years. Only recently has conclusive evidence been found which indicates that mixing due to radial secondary flows is equally as important as turbulent diffusion near the endwall regions of axial compressors. However, a very recent study on spanwise mixing in turbines reveals that mixing due to radial secondary flows is just as important as, and in some cases more important than, turbulent diffusion in almost all regions of the machine.

The program, DUCT, developed for the present study, includes both mixing mechanisms. This chapter describes the methods by which turbulent and convective radial mixing are incorporated into the basic streamline curvature theory presented in Chapter 5.

6.1 Mixing Due to Turbulent Diffusion

The method used to include the effects of spanwise mixing due to turbulent diffusion is based on the model of Gallimore (1986). The model has been incorporated into the streamline curvature computation described in Chapter 5 and uses the momentum and energy equations to calculate streamwise changes of stagnation enthalpy, entropy, and tangential momentum caused by the radial diffusion of heat and momentum. The turbulence model, which is described in the following section, modifies the adiabatic and inviscid assumptions of the basic streamline curvature method allowing the radial transfer of heat and momentum and implying the presence of axial and tangential shear stresses. The mixing model of Gallimore was developed for use in the analysis of compressor flows where streamlines remain almost parallel to the machine axis and

radial velocities are quite small. Due to the rapidly diverging nature of turbine annuli, however, turbine flows can have radial velocities which are quite significant. Thus, in order for the model to be valid in the analysis of turbine flows, Gallimore's assumptions of negligible radial velocities and spatial derivatives of radial velocities have been relaxed. This leads to some extra terms being included in the mixing equations presented by Gallimore (1986).

6.1.1 Turbulence Model

This section describes the turbulence model used by Gallimore (1986) and by the present author. The model is based on the familiar eddy viscosity concept which is introduced here using some fundamental concepts of fluid mechanics.

First consider a two-dimensional laminar viscous flow in which the x -direction shear stress, τ_{xy} , is linearly related to the fluid strain rate in the x -direction, $\partial u/\partial y$, by

$$\tau_{xy} = \mu \frac{\partial u}{\partial y} ,$$

where μ is called the coefficient of viscosity. Fluids for which the above relationship holds are referred to as Newtonian fluids. The coefficient of viscosity, μ , is usually called the molecular viscosity of the fluid and is a function of fluid temperature and pressure.

Now, consider a two-dimensional turbulent viscous flow where the x -direction shear stress at any point in the fluid is given by

$$\tau_{xy} = \mu \frac{\partial \bar{u}}{\partial y} - \rho \overline{u'v'} .$$

Here, u' and v' represent the fluctuating velocity components in the x and y -directions respectively and the over-bars indicate time-averaged or mean values. The second term on the right hand side represents the turbulent mixing effect of the fluid. Mathematically, however, this term can be treated as a shear stress commonly referred to as a "Reynolds" stress. A traditional modelling assumption attributed to J. Boussinesq in 1877 is to replace this turbulent mixing term with one

analogous to molecular shear stress. The total x-direction shear stress can now be represented by

$$\tau_{xy} = (\mu + \mu_t) \frac{\partial \bar{u}}{\partial y} ,$$

where μ_t is called the eddy viscosity. Unlike molecular viscosity, the eddy viscosity is not a fluid property but varies with flow conditions and geometry (ie. the state of the turbulent eddies).

In an analogous way, it is also possible to define an eddy thermal conductivity, k_t , which relates the heat transfer rate per unit area in the y-direction to the local mean temperature gradient in the fluid by

$$q_y = -(k + k_t) \frac{\partial \bar{T}}{\partial y} ,$$

where k is the molecular thermal conductivity of the fluid.

In the mixing model of Gallimore (1986) it is assumed that the radial mixing of heat and momentum can be represented by k_t and μ_t respectively and that k and μ are negligible since $k_t \gg k$ and $\mu_t \gg \mu$ in a turbulent free stream flow. Eddy viscosity and eddy thermal conductivity can be formed into a dimensionless ratio called the turbulent Prandtl number, given by,

$$Pr_t = \frac{\mu_t c_p}{k_t} . \quad (6.1)$$

The eddy viscosity can also be related to a parameter, ϵ , which describes the turbulent diffusivity of mass, using the turbulent Schmidt number,

$$Sc_t = \frac{\mu_t}{\rho \epsilon} . \quad (6.2)$$

Gallimore (1986) refers to ϵ as the mixing coefficient which has dimensions of length squared per unit time. This will be the terminology used in the present study.

The turbulent Prandtl number represents the ratio of turbulent momentum diffusivity to turbulent thermal diffusivity while the turbulent Schmidt number represents the ratio of turbulent momentum diffusivity to turbulent mass diffusivity. Since these three diffusivities are due to the

same mechanism, namely turbulent mixing, it stands to reason that Pr_t and Sc_t ought to be of order unity. Experimental evidence has shown this to be the case for many flow conditions, especially for gases (see, for example, Eckert and Drake (1972) or White (1991)). For the purposes of the present study it is assumed that $Pr_t = Sc_t = 1$.

In order for the turbulence model to be complete one needs to determine reasonable values of μ_t and k_t and thus reasonable values of the mixing coefficient, ϵ . Methods for doing this will be discussed in Section 6.3.

6.1.2 Mixing Equations

This section presents the mixing equations which are used in the streamline curvature method to predict the streamwise changes in entropy, angular momentum, and stagnation enthalpy due to the radial diffusion of heat and momentum. Since the derivation of the equations is quite lengthy, only a summary will be presented here. The reader is referred to Appendix C for the detailed derivations.

The mixing equations are derived from the energy and momentum equations for steady laminar viscous flow. To employ the turbulence model described in the previous section, the molecular viscosity, μ , and thermal conductivity, k , are replaced by the eddy viscosity, μ_t , and eddy thermal conductivity, k_t , respectively. As before, the flow is assumed to be axisymmetric; however, a viscosity is now implied and the flow along meridional streamlines is no longer isentropic. There is radial heat transfer within the flow and the generation of entropy due to viscous dissipation. The fluid is considered to be an ideal gas with compressibility effects accounted for with the exception of shocks and choking. It is also assumed, for the development of the mixing equations, that axial gradients of fluid properties are negligibly small so that mixing in the axial direction is insignificant.

Now, the energy equation for steady laminar viscous flow can be written in vector form as

$$\rho \frac{Dh}{Dt} = \frac{Dp}{Dt} + \nabla \cdot (k \nabla T) + \Phi ,$$

where Φ is called the dissipation function. Under the assumptions stated above, the energy equation can be transformed into

$$\frac{\partial S}{\partial m} = \frac{1}{r \rho T v_m} \left[\frac{\partial}{\partial r} \left(r k_t \frac{\partial T}{\partial r} \right) \right] + \frac{\Phi}{\rho T v_m} \quad (6.3)$$

after some manipulation, where Φ is given by

$$\Phi = \mu_t \left[2 \left(\left(\frac{\partial v_r}{\partial r} \right)^2 + \left(\frac{v_r}{r} \right)^2 \right) + \left(\frac{\partial v_x}{\partial r} \right)^2 + \left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \right)^2 \right]$$

in the (r, θ, x) coordinate system. Equation (6.3) gives the streamwise entropy gradient at any point along a meridional streamline due to the turbulent diffusion of thermal energy and entropy production by viscous dissipation. The reader is referred to Figure 5.2 for a summary of nomenclature. Empirical correlations can also be included to calculate entropy changes across blade rows. If this is the case, Equation (6.3) becomes

$$\frac{\partial S}{\partial m} = \frac{1}{r \rho T v_m} \left[\frac{\partial}{\partial r} \left(r k_t \frac{\partial T}{\partial r} \right) \right] + \frac{\Phi}{\rho T v_m} + \frac{\partial S_e}{\partial m} ,$$

where the last term on the right-hand-side is based on empiricism. This is the same as Equation (2.4) given by Gallimore (1986) except that here the expression for Φ contains some additional terms.

The momentum equations for steady laminar viscous flow are the Navier-Stokes equations, which in vector form can be written,

$$\rho \frac{DV}{Dt} = -\nabla p + \nabla(\lambda \nabla \cdot V) + \mu(\nabla^2 V + \nabla(\nabla \cdot V)) + 2(\nabla \mu \cdot \nabla)V + \nabla \mu \times (\nabla \times V) + \rho F .$$

Under the assumptions stated previously, the above equation can be written in component form for the (r, θ, x) coordinate system as

$$\rho \left(\frac{Dv_r}{Dt} - \frac{v_\theta^2}{r} \right) = \rho F_r - \frac{\partial p}{\partial r} + E_r,$$

$$\rho \left(\frac{Dv_\theta}{Dt} + \frac{v_r v_\theta}{r} \right) = \rho F_\theta + E_\theta,$$

and

$$\rho \left(\frac{Dv_z}{Dt} \right) = \rho F_z - \frac{\partial p}{\partial z} + E_z,$$

where E_r , E_θ , and E_z are the stress terms and are given by

$$E_r = \frac{2}{r} \left[\frac{\partial}{\partial r} \left(\mu_t r \frac{\partial v_r}{\partial r} \right) - \frac{\mu_t v_r}{r} \right],$$

$$E_\theta = \frac{\partial}{\partial r} \left[\mu_t \left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \right) \right] + \frac{2\mu_t}{r} \left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \right),$$

and

$$E_z = \frac{1}{r} \frac{\partial}{\partial r} \left[\mu_t r \frac{\partial v_z}{\partial r} \right]$$

respectively. The radial stress term, E_r , was assumed to be negligible by Gallimore (1986). Since the mixing analysis is to be applied to an annular duct flow, the body force terms in the above equations can be dropped. The three momentum equations can be manipulated to give the streamwise gradient of angular momentum along a meridional streamline as given by

$$\frac{\partial}{\partial m} (r v_\theta) = \frac{r}{v_\theta} \left(\frac{E_\theta}{\rho} \right) \quad (6.4)$$

and the streamwise gradient of stagnation enthalpy along a meridional streamline given by

$$\frac{\partial h_o}{\partial m} = T \frac{\partial S}{\partial m} + \frac{v_\theta}{r} \frac{\partial}{\partial m} (r v_\theta) + \frac{E_r}{\rho} \sin \zeta + \frac{E_s}{\rho} \cos \zeta . \quad (6.5)$$

Again, the reader is referred to Appendix C for detailed derivations of Equations (6.3), (6.4), and (6.5).

It should be noted here that the mixing equations have been expressed in terms of the radial direction for simplicity in the present study. For cases involving annuli with bladed regions or annuli with sharply diverging endwalls, quasi-orthogonals should be placed at angles to the radial direction. In such cases the mixing equations (as well as the radial equilibrium equation) should be expressed in terms of the q -direction.

6.1.3 Method of Implementation

This section discusses how the mixing equations, presented in the previous section, are incorporated into the streamline curvature computation method described in Chapter 5.

In the basic streamline curvature method, flow along meridional streamlines is assumed to be isentropic. This condition dictates that entropy, angular momentum, and stagnation enthalpy are conserved along streamlines. With the use of the mixing equations, the isentropic flow assumption is relaxed. Equations (5.4), (5.6), and (5.7) are no longer valid and are replaced by Equations (6.4), (6.5), and (6.3) respectively. These equations are solved to give the changes in angular momentum, stagnation enthalpy, and entropy along streamlines in Step 7 of the solution procedure presented in Section 5.2.2. The method used to solve Equations (6.3), (6.4), and (6.5) is presented in Section 7.5.

6.1.4 Boundary Conditions

When solving the mixing equations it is necessary to enforce certain boundary conditions at the hub and casing surfaces of the annulus. These surfaces are assumed to be adiabatic, such

that the temperature gradients normal to the surfaces are zero. This condition can be approximated by

$$\frac{\partial T}{\partial r} \Big|_{wall} = 0 .$$

The annulus boundaries are also assumed to be frictionless such that the velocity component gradients normal to the boundaries are zero. This condition can be approximated by

$$\frac{\partial v_x}{\partial r} \Big|_{wall} = 0 , \quad \frac{\partial v_\theta}{\partial r} \Big|_{wall} = 0 , \quad \text{and} \quad \frac{\partial v_r}{\partial r} \Big|_{wall} = 0 .$$

The above boundary conditions become more inaccurate as the annulus walls deviate from the axial direction. It should be noted that if the mixing equations are written in terms of the q -direction, then it is more accurate to use gradients in the q -direction for the above boundary conditions.

6.2 Mixing Due to Secondary Flows

This section explains how the computed secondary flow velocities, obtained from program BLADE, are used to incorporate convective radial mixing in the streamline curvature computation.

The method used to incorporate radial mixing due to secondary flows, into the meridional flowfield calculation, is based on that of Adkins and Smith (1982). The secondary flow computation yields a y - z distribution of three velocity components at the blade passage exit plane. These velocity components are in the x , y' , and z -directions (see Figure 6.1). It is impossible to use the secondary flow velocities directly in the streamline curvature computation since this computation takes place on the two-dimensional meridional plane (an x - r or x - z plane) and is supposed to represent the circumferentially averaged flowfield. Circumferentially (pitchwise) mass-averaging the secondary flow velocities would result in zero radial velocities in order to

Figure 6.1 Idealized flow channel downstream of blade passage.

satisfy continuity. Thus, the radial mixing due to secondary flows must be modelled in such a way that the level of mixing depends directly on the computed radial velocities. The approach taken by Adkins and Smith (1982) is to model the radial convection as a diffusion process. The following derivation of a convective mixing equation has been adapted from Adkins and Smith (1982) with modifications as indicated.

For simplicity it is assumed that the flow downstream of the blade passage travels in a straight channel of constant rectangular cross-section. As indicated in Figure 6.1, the channel has width, a , in the pitchwise direction and height, l , equal to the blade span. Downstream of the blade row trailing edge, the distribution of a fluid property is determined by its distribution at the blade passage exit and by its convection under the action of the secondary flow field. Consider a fluid particle which is convected from a point on the blade passage exit plane to a point on a y - z plane (parallel to the exit plane) downstream. The path of this fluid particle is a streamline, as illustrated in Figure 6.1. If fluid property P is conserved along the streamline then

$$P(x, y, z) = P(y_e, z_e) \quad , \quad (6.6)$$

where the subscript, e , indicates coordinates on the blade trailing edge plane.

The z coordinate of the fluid particle at the exit and downstream planes is related to the radial velocity component by

$$z = z_e + \frac{w}{u'} x' \quad ,$$

where x' is the downstream distance from the blade trailing edge plane and u' is the velocity component in the primary flow direction. The axial and primary flow directions are related by

$$x = x' \cos \alpha_e$$

while the axial and primary flow velocities are related by

$$u = u' \cos \alpha_e \quad ,$$

where α_e is the blade exit flow angle. Thus the local and exit plane z coordinates of the particle are related through

$$z = z_e + \frac{w}{u} x, \quad (6.7)$$

where x is the axial distance from the blade trailing edge plane.

Now, for each location y_e , the function $P(y_e, z_e)$ can be represented as a Taylor series given by

$$P(y_e, z_e) = P(y_e)_{z_e=0} + z_e \frac{\partial P(y_e)}{\partial z} \Big|_{z_e=0} + \frac{1}{2} z_e^2 \frac{\partial^2 P(y_e)}{\partial z^2} \Big|_{z_e=0} + \dots$$

Combining this with Equations (6.6) and (6.7) gives

$$P(x, y, z) = P(y_e)_{z_e=0} + \left(z - \frac{w}{u} x \right) \frac{\partial P(y_e)}{\partial z} \Big|_{z_e=0} + \frac{1}{2} \left(z - \frac{w}{u} x \right)^2 \frac{\partial^2 P(y_e)}{\partial z^2} \Big|_{z_e=0} + \dots \quad (6.8)$$

Since the mixing calculation is performed in the meridional plane (x - z or x - r plane), the cross-passage or y -direction distribution of P is unknown and thus a more suitable quantity to use is the cross-passage (or tangential) average value of P . Adkins and Smith (1982) used the cross-passage area-average of P in their calculations but, as Dring (1992) pointed out, the cross-passage mass-average of P is a more appropriate quantity to use. The cross-passage mass-average of property P is given by

$$\bar{P}^m(x, z) = \frac{\frac{1}{a} \int_0^a \rho u P(x, y, z) dy}{\frac{1}{a} \int_0^a \rho u dy}$$

and thus

$$\frac{\partial \bar{P}^m(x, z)}{\partial x} = \frac{\frac{1}{a} \int_0^a \rho u \frac{\partial P(x, y, z)}{\partial x} dy}{\frac{1}{a} \int_0^a \rho u dy} .$$

Combining this with Equation (6.8) and evaluating at the spanwise reference value of $z=0$ gives

$$\frac{\partial \bar{P}^m(x, z)}{\partial x} = \frac{1}{\frac{1}{a} \int_0^a \rho u dy} \left[-\frac{1}{a} \int_0^a \rho u \frac{w}{u} \frac{\partial P(y_r)}{\partial z} \Big|_{z=0} dy + \frac{x}{a} \int_0^a \rho u \left(\frac{w}{u} \right)^2 \frac{\partial^2 P(y_r)}{\partial z^2} \Big|_{z=0} dy + \dots \right] \quad (6.9)$$

To simplify the above series it is assumed that the first and second spanwise derivatives of P are constant in the y -direction at the blade trailing edge plane. Note also that, as Dring (1992) pointed out, the first term inside the brackets on the right-hand-side of Equation (6.9) must be equal to zero in order to satisfy continuity. If the second spanwise derivative is replaced by its local value and the higher order terms are neglected, the series then becomes

$$\frac{\partial \bar{P}^m(x, z)}{\partial x} = \frac{\frac{x}{a} \int_0^a \rho u \left(\frac{w}{u} \right)^2 dy}{\frac{1}{a} \int_0^a \rho u dy} \frac{\partial^2 \bar{P}^m(x, z)}{\partial z^2}$$

which can be written

$$\frac{\partial \bar{P}^m(x, z)}{\partial x} = \beta \frac{\partial^2 \bar{P}^m(x, z)}{\partial z^2} , \quad (6.10)$$

where β is a type of diffusion coefficient given by

$$\beta = \frac{\frac{x}{a} \int_0^a \rho u \left(\frac{w}{u} \right)^2 dy}{\frac{1}{a} \int_0^a \rho u dy} \quad (6.11)$$

Equation (6.10) is the convective transport equation (2.2) used by Adkins and Smith (1982). It states that, locally, the axial gradient of a fluid property is directly proportional to the second spanwise derivative of the fluid property. The properties which are mixed are values of entropy, total enthalpy, and angular momentum on the meridional plane. The local constant of proportionality is β , which has dimensions of length. The expression for β , Equation (6.11), is different from Equation (2.3) used by Adkins and Smith in that it employs a mass-average rather than an area-average.

The use of Equation (6.10) in modelling the convective effects of secondary flows has been questioned by several authors. De Ruyck and Hirsch (1988) conclude that, in general, convective effects cannot be simulated by a diffusion type equation. Dring (1992) found that, although the use of Equation (6.10) smoothed out the fluid property profiles in computations, it was not driving the computed results towards those measured in experiments.

Several peculiarities in Equations (6.10) and (6.11) can be quickly noticed. First, if the second spanwise derivative of a fluid property is zero (ie. a linear profile) then there is no mixing regardless of the value of β . Second, the value of β increases linearly with axial distance. This means that if the secondary velocities remained constant downstream (they will actually decrease due to viscous dissipation), for example, β would continue to increase linearly, thereby increasing the level of mixing. Finally, since x is the axial distance from the blade trailing edge plane, β is zero here and thus no convective mixing will take place in this region where it is expected to be a maximum.

The author acknowledges the difficulties associated with Equation (6.10), but in the absence of an alternative method to model the effects of convective mixing in a meridional flow-

field calculation, the author has used it in the present study.* Actually, only Equation (6.11) is used in the present study but its use implies the validity of Equation (6.10). Several empirical modifications can be made to Equation (6.11) to remove some of the peculiarities mentioned above. This will be discussed further in Section 6.3.2. Adkins and Smith (1982) refer to β as the "mixing coefficient", but it is different from the mixing coefficient, ϵ , of Equation (6.2). β , however, can be related to ϵ , as described in the following section.

6.3 Mixing Coefficients

This section explains how suitable values of the mixing coefficients are obtained and also clarifies how the effects of both turbulent and convective mixing are incorporated into the streamline curvature computation.

It was mentioned in the previous section that β , the modified convective "mixing coefficient" of Adkins and Smith (1982), was related to the mixing coefficient, ϵ , of Equation (6.2). This relationship takes the form

$$\epsilon_c = \bar{u}\beta \quad (6.12)$$

where $\bar{u}$ represents the cross-passage mass-average axial velocity or simply the axial velocity in the meridional plane. The subscript c signifies that this mixing coefficient is one based on convective radial flows.

To incorporate the effects of radial mixing due to both turbulent diffusion and convective secondary flows in the present study, the mixing coefficient, ϵ , of Equation (6.2) is considered to have two components and is thus given by

$$\epsilon = \epsilon_t + \epsilon_c \quad (6.13)$$

where the subscripts t and c represent the contributions of turbulent mixing and convective mixing

* In the defence of this thesis it was noted that Reichardt used an expression similar to Equation (6.10) to describe lateral momentum transport in a free turbulent flow (see Hinze, 1975)

respectively. ϵ is now referred to as the effective mixing coefficient. Simple superposition of the two effects does not seem unreasonable since the experiments of Gregory-Smith and Cleak (1992) revealed that cascade exit secondary flow velocities were essentially unaffected by the level of inlet turbulence. This suggests that the two effects are not strongly interdependent. Methods for obtaining suitable values for ϵ_t and ϵ_c are discussed in the following sections.

6.3.1 Turbulent Mixing Coefficient

The turbulent mixing coefficient, ϵ_t , is that used by Gallimore (1986) and has a constant value throughout the entire meridional flowfield. Its value is obviously related to the turbulence intensity of the flow. Gallimore and Cumpsty (1986) have developed an expression for predicting the turbulent mixing coefficient in compressors. This was given earlier as Equation (2.7). Equation (2.7) gives ϵ_t in terms of typical turbomachinery parameters so there appears to be no difficulty in adapting it for use in turbine analysis. However, for the purposes of the duct flow analysis in the present study, Equation (2.7) is not useful. Thus, ϵ_t has been left as an empirical input in program DUCT. Several authors have published values of the dimensionless turbulent mixing coefficient, ϵ_t/uL_r , for a variety of turbulent flows (see Section 2.3.3). These values were used as a guide for the present study.

6.3.2 Convective Mixing Coefficient

Ideally the convective mixing coefficient, ϵ_c , should be calculated using the tangential mass-averaged values of radial and axial velocities in the annular flow. The computed secondary flow velocities, however, are only available at the blade row exit plane and thus must be empirically modelled downstream. Adkins and Smith (1982) employ a number of empirical terms in their expression for β in order to make a better match between their computed and experimental results. Some of these empirical terms have been used in the present study in order to alleviate the peculiarities associated with Equation (6.11).

Following Adkins and Smith (1982), the linear increase in β with axial distance has been

decreased by a linear function of axial distance. As well, it is recognized that mixing starts before the blade trailing edge plane. In order for β to be non-zero at the blade trailing edge plane, the origin for $\beta=0$ has been placed at the axial mid-chord of the blade row. The resulting term is parabolic, equalling zero at the blade row mid-axial chord, peaking to a maximum downstream, then decreasing to zero at distance equal to five blade spans in the axial direction. This is in accordance with Adkins and Smith.

The development in Section 6.2 which led to Equation (6.11) did not take into account the viscous dissipation of the secondary flow field downstream. Thus it was assumed that the integrand $(w/u)^2$ in Equation (6.11) was not a function of axial position. Experimental data collected by Dring (1992) suggested an axial power-law decay for the tangential mass-average of $(w/u)^2$; the power-law exponent being between -0.5 and -1.0. In the computations of Adkins and Smith (1982), however, the cross-passage area-average values of $(w/u)^2$, obtained at the trailing edge plane, were decreased in a linear manner to reach a constant value at five blade spans downstream of the trailing edge. This constant value was taken to be the spanwise mass average of the trailing edge $(w/u)^2$ values obtained by cross-passage area-averaging. In light of the more recent findings of Dring (1992), the author has opted to use a power-law decay of radial velocities in flow downstream of the blade trailing edge plane. The power-law exponent has been left as an empirical input to program DUCT and the range of exponents suggested by Dring was used as a guide. Following Adkins and Smith (1982), however, the axial distance required for the power-law function to reach zero, was taken as five blade spans from the blade trailing edge plane.

The analysis which led to Equation (6.11) also made no allowance for axial variation of the annulus height. This was compensated for by Adkins and Smith (1982) with the use of an axial velocity ratio term in their final expression for β . This term was adopted in the present study.

Combining all the factors mentioned above with Equations (6.11) and (6.12) gives

$$\epsilon_t(x, r) = \left[\left(x + \frac{c_x}{2} \right) \left(1 - \frac{1}{5l} \left(x + \frac{c_x}{2} \right) \right) \right] \left[\overline{\left(\frac{w}{u} \right)^2}^m \left(1 - \left(\frac{x}{5l} \right)^d \right) \right] \left[\frac{\overline{u_e^2}^m}{u} \right], \quad (6.15)$$

where

$$\overline{\left(\frac{w}{u} \right)^2}^m = \frac{\int_0^a u_e \left(\frac{w_e}{u_e} \right)^2 dy}{\int_0^a u_e dy},$$

c_x is the blade axial-chord length, l is the blade span, and d is the power-law exponent. It should be noted that the power-law exponent is positive in Equation (6.15) and corresponds to values of opposite sign to those quoted by Dring (1992). The over-bars with m superscripts in Equation (6.15) indicate a pitch-wise mass-average. The density, ρ , has disappeared because the secondary flow computation is incompressible and there is no variation of density at the blade passage exit plane.

With the use of Equation (6.15), an empirically determined value of the turbulent mixing coefficient, ϵ_t , and Equation (6.13) the effective mixing coefficient, ϵ , can be determined at every point in the meridional plane. With the mixing coefficient known, both eddy viscosity and eddy thermal conductivity can be determined using Equations (6.2) and (6.1) respectively and recalling that Pr_t and Sc_t are assumed equal to one.

ANNULAR DUCT FLOW COMPUTATION: NUMERICAL PROCEDURES

7

This chapter gives specific details of the methods and procedures used in the annular duct flow computation of program DUCT. The various steps of the streamline curvature method, outlined in Section 5.2.2, are described in detail and presented in a logical order. Also in this chapter, the solution procedure for the mixing equations developed in Section 6.1.2 is presented.

7.1 Initial Conditions

In order to begin the iterative procedures of the streamline curvature method it is necessary to establish some initial state of the flow field (Step 1 in Section 5.2.2). This section gives details on the methods used to initially estimate the required properties throughout the meridional flow field. The initial conditions are set up in the main body of program DUCT.

It is first necessary to define a computational grid for the meridional plane. The spatial coordinates (x and r) of the grid nodes are calculated by finding the intersection points of quasi-orthogonals (QOs) and initially placed streamlines. The coordinates of the QO endpoints are user specified and define the annulus boundaries. There is evidence that the relative positioning of the QOs can have a significant effect on solution convergence and Cox (1976) recommends that equal axial spacing of QOs should be used whenever possible. This, of course, is always possible in an annular duct. Now, the initial placement of streamlines is such that each axisymmetric streamtube has an equal cross-sectional area at a particular QO, the total number of streamtubes also being user specified. As described in Section 5.2.2, the computational grid formed by the intersection of QOs and streamlines may be directly superimposed on the physical domain of the meridional plane. This is illustrated in Figure 7.1 for the meridional plane of a hypothetical annulus. Streamlines are numbered from 1 to $SLMAX$ starting at the hub while QOs are numbered

Figure 7.1 Node indices for the computational grid of a hypothetical annulus.

from 1 to *QOMAX* starting at the inlet. The computational nodes are identified by the indices *i* and *j* which also identify the column and row numbers, respectively, in the corresponding computational arrays.

Once the initial (*x*,*r*) coordinates of the nodes have been found, streamline radius of curvature, *R_c*, slope, *ζ*, and angle *φ* (see Fig. 5.2) must be calculated at each node. In order to calculate these quantities it is first necessary to know the first and second derivatives of *r* with respect to *x* at each node. The user is given a choice of two different methods for doing this. The first method involves fitting each streamline with a cubic spline while the second method simply utilizes three-point finite difference approximations. Details of both methods are given in Section 7.2.

Now, in order to compute the other flow field properties necessary to begin the iteration procedure, some property must be guessed. Since the total mass flow rate is user specified, an initial guess of axial velocity seemed to be a logical choice. This quantity could be reasonably approximated for a turbomachine, given its annulus dimensions and mass flow rate. Therefore, the user is prompted for an initial value of *v_x* which is assumed uniform throughout the flow field. This assumption is also reasonable since many turbomachines are designed to have approximately constant axial velocity throughout.

It now becomes possible to calculate the radial and meridional velocity components at each node using

$$v_{r,i,j} = v_x \tan \zeta_{i,j}$$

and

$$v_{m,i,j} = \sqrt{v_x^2 + v_{r,i,j}^2}$$

respectively.

Since radial distributions of swirl angle, *Ω*, stagnation temperature, and stagnation pressure are given at the inlet QO (*i*=1), it becomes possible to estimate these quantities at the inlet nodes

using linear interpolation. Tangential velocity, temperature, pressure, and stagnation enthalpy are calculated at the inlet nodes using

$$v_{\theta_{1,j}} = v_x \tan \Omega_j ,$$

$$T_{1,j} = T_o - \frac{\gamma-1}{2} \frac{V_{1,j}^2}{\gamma R} ,$$

$$p_{1,j} = p_o \left(\frac{T_o}{T_{1,j}} \right)^{\frac{\gamma}{1-\gamma}} ,$$

and

$$h_{o,j} = c_p T_{1,j} + \frac{V_{1,j}^2}{2} ,$$

respectively, where the absolute velocity is given by

$$V_{1,j}^2 = v_x^2 + v_{r,j}^2 + v_{\theta,j}^2 .$$

Entropy values at the inlet nodes can also be calculated using

$$S_{1,j+1} = S_{1,j} + c_p \ln \left(\frac{T_{1,j+1}}{T_{1,j}} \right) - R \ln \left(\frac{p_{1,j+1}}{p_{1,j}} \right) ,$$

where the hub-inlet-corner node is used as a datum by setting $S_{1,1} = 0$.

For the purposes of setting up the initial flow field, it is assumed that the flow along streamlines is isentropic. This requires that angular momentum, stagnation enthalpy, and entropy be conserved along streamlines and is one of the assumptions of the basic streamline curvature method in which radial mixing is not permitted. The assumption is used here since there is not yet enough information at the nodes to evaluate the mixing equations ((6.3), (6.4), and (6.5)) which give streamwise gradients of angular momentum, stagnation enthalpy, and entropy. Thus, tangential velocity, stagnation enthalpy, and entropy for the remainder of the flow field is calculated via

$$v_{\theta,i+1,j} = \frac{r_{i,j}}{r_{i+1,j}} v_{\theta,i,j} ,$$

$$h_{o,i+1,j} = h_{o,i,j} ,$$

and

$$S_{i+1,j} = S_{i,j} ,$$

respectively, where $i = 1, 2, \dots, QOMAX-1$. The remaining properties of temperature, pressure, and density can then be calculated in the entire flow field using

$$T_{i+1,j} = \frac{1}{c_p} \left(h_{o,i+1,j} - \frac{V_{i+1,j}^2}{2} \right) ,$$

$$p_{i+1,j} = p_{i,j} \exp \left[\frac{c_p \ln \left(\frac{T_{i+1,j}}{T_{i,j}} \right)}{R} \right] ,$$

and

$$\rho_{i+1,j} = \frac{p_{i+1,j}}{RT_{i+1,j}} ,$$

respectively. The necessary properties are now estimated at all computational nodes and the iterative procedures of the streamline curvature method can begin.

7.2 Estimating Streamline Slope and Curvature

In the streamline curvature method it is necessary to estimate streamline slope and radius of curvature at each node of the computational grid. Getting accurate estimates of the curvature has long been one of the major difficulties in applying the streamline curvature method since significant errors in curvature can affect solution stability. This section presents details of how streamline slope and curvature are calculated in program DUCT.

Several authors have discussed various ways of approximating streamline slope and curvature. Wilkinson (1970) investigated the accuracy and stability of a large number of curve fits on a sinusoidal streamline and concluded that polynomial fits, using the same curve to obtain both slope and curvature, were the most accurate and stable while the use of splines and of 'double differentiation' were undesirable. Shaalan and Daneshyar (1972) suggested that accurate estimates of curvature can be obtained by passing a second spline through data points of streamline slope obtained by differentiating an initial streamline spline fit. They refer to this as a double-spline fit and conclude that only four or five points are needed to obtain good estimates of curvature as opposed to ten points for a single spline fit. Novak and Hearsay (1977), however, reported that they found little difference between solutions obtained using spline fits and quadratic fits to compute slope and curvature, although the quadratic method was more stable. Denton (1978) opted for a simple parabolic curve fit through three points since his earlier experience with streamline curvature computations showed that overall accuracy was determined more by empirical inputs than by the numerical scheme used. On a purely analytical basis, Walsh et al. (1962), noted the marked superiority of the quality of the derivatives obtained with the use of a cubic spline fit over those obtained from polynomials of interpolation.

In light of these findings it was decided to give the user a choice of either using a cubic spline fit to estimate streamline slope and curvature or to simply use numerical differentiation to obtain the required derivatives. Both options are discussed below.

7.2.1 Rational Cubic Spline

Using a natural cubic spline to pass a smooth curve through a set of data points sometimes results in unacceptable spline oscillations simply due to the positions of the points. These unwanted oscillations can be greatly reduced by introducing a damping factor into the cubic spline formulation. This modified cubic spline is sometimes referred to as a rational cubic spline and is illustrated in Figure 7.2 using a set of hypothetical data points and various values of the damping factor q . The following development concerning the rational cubic spline has been adapted from Spaeth (1974) and used in subroutine SPLINE.

Figure 7.2 Damping effects of the rational cubic spline (adapted from Spaeth, 1974).

Given a set of n data points of the form (r_i, x_i) , such that $x_1 < x_2 < \dots < x_n$, curves of the form,

$$f_i(x) = A_i u + B_i t + \frac{C_i u^3}{qt+1} + \frac{D_i t^3}{qu+1}, \quad (7.1)$$

can be passed through the data between points i and $i+1$, where

$$t = \frac{x - x_i}{x_{i+1} - x_i}, \quad u = 1 - t,$$

and q is the damping coefficient which can range from 0 to ∞ . If q is set to zero, the natural cubic spline is obtained whereas if q approaches ∞ the curve takes the form of a series of straight line segments between data points. This trend can be seen in Figure 7.2. Now, differentiating Equation (7.1) gives

$$\Delta x_i f'_i(x) = B_i - A_i + C_i \frac{2qu^3 - 3u^2(q+1)}{(qt+1)^2} - D_i \frac{2qt^3 - 3t^2(q+1)}{(qu+1)^2}$$

where $\Delta x_i = x_{i+1} - x_i$ and $i=1, \dots, n-1$. Differentiating Equation (7.1) a second time yields

$$(\Delta x_i)^2 f''_i(x) = C_i \frac{2q^2u^3 - 6q(q+1)u^2 + 6(q+1)^2u}{(qt+1)^3} + D_i \frac{2q^2t^3 - 6q(q+1)t^2 + 6(q+1)^2t}{(qu+1)^3}$$

and thus

$$r_i = A_i + C_i, \quad (7.2.a)$$

$$r_{i+1} = B_i + D_i, \quad (7.2.b)$$

$$\Delta x_i r'_i = B_i - A_i - (3+q)C_i, \quad (7.3.a)$$

$$\Delta x_i r'_{i+1} = B_i - A_i + (3+q)D_i, \quad (7.3.b)$$

$$(\Delta x_i)^2 r''_i = 2(q^2 + 3q + 3)C_i, \quad (7.4.a)$$

and

$$(\Delta x_i)^2 r_{i+1}'' = 2(q^2 + 3q + 3)D_i. \quad (7.4.b)$$

Here $r' = \partial r / \partial x$ and $r'' = \partial^2 r / \partial x^2$. If the r_i'' 's are introduced as unknowns, then from Equations (7.2) and (7.4) the coefficients are given by

$$C_i = \frac{1}{2} \frac{(\Delta x_i)^2}{q^2 + 3q + 3} r_i'',$$

$$D_i = \frac{1}{2} \frac{(\Delta x_i)^2}{q^2 + 3q + 3} r_{i+1}'', \quad (7.5)$$

$$A_i = r_i - C_i,$$

and

$$B_i = r_{i+1} - D_i.$$

Finally, since the first derivatives are continuous for $f_i(x)$, the system

$$\Delta x_{i-1} r_{i-1}'' + (2+q)(\Delta x_{i-1} + \Delta x_i) r_i'' + \Delta x_i r_{i+1}'' = 2(q^2 + 3q + 3) \left(\frac{\Delta r_i}{\Delta x_i} - \frac{\Delta r_{i-1}}{\Delta x_{i-1}} \right)$$

results for $i=2, \dots, n-1$, where $\Delta r_i = r_{i+1} - r_i$. With the specified boundary conditions, $r_1''=0$ and $r_n''=0$, the solution of this system (obtained from subroutine SOLVE) gives r'' at every data point and thus r' at every data point using Equations (7.3) and (7.5). Equation (7.3.a) is used to find r' for all data points except the last one where Equation (7.3.b) is used. Once the first and second derivatives of r with respect to x have been found for all nodes along a particular streamline, slope and radius of curvature can be calculated. The formulae for doing this are included in the next section.

7.2.2 Slope and Curvature Using Finite Differences

This section presents an alternative method to the rational cubic spline for computing the spatial derivatives necessary to calculate streamline slope and curvature.

For any given streamline, the first and second derivatives of r with respect to x can be calculated at all nodes using simple three-point finite-difference approximations. Derivatives at the inlet plane are calculated using forward-difference approximations while backward-difference approximations are applied to nodes at the exit plane. Of course derivatives at interior nodes are calculated using central-difference expressions. Once the first derivative is known for all nodes along a streamline, these values may be differentiated a second time to get the second derivative. This is the method used in subroutine FINDIF. Formulae for all of these finite difference approximations are presented in Appendix D.

Once the first and second spatial derivatives are known along the streamline the slope angle, ζ , is calculated at all nodes using

$$\zeta_{i,j} = \tan^{-1}(r'_{i,j}) .$$

However, in the radial equilibrium equation, streamline slope is expressed in terms of angle ϕ which is given by

$$\phi_{i,j} = \frac{\pi}{2} + \kappa_i - \zeta_{i,j} .$$

Here, κ_i is the angle between the q and r -directions and is calculated using

$$\kappa_i = \tan^{-1} \left(\frac{x_{hub} - x_{casing}}{r_{casing} - r_{hub}} \right) ,$$

for each QO.

The radius of curvature, R_c , can be represented by

$$R_c = \frac{\partial m}{\partial \zeta}$$

and, as indicated in Figure 5.2, is positive for streamline curvature which is concave in the positive r -direction. The radius of curvature can be expressed in terms of known quantities using

$$R_{c,i} = \frac{1}{\cos^3 \zeta_{c,i} r_{c,i}}$$

at every node.

7.3 Solution of the Radial Equilibrium Equation

This section details the technique in which the radial equilibrium and continuity equations are solved iteratively in order to update the meridional velocity field. This technique represents the inner iterative loop of the streamline curvature method and is outlined in Steps 2, 3, and 4 of Section 5.2.2. The code for this procedure is contained in subroutine NEWVM.

It is necessary to solve the radial equilibrium equation (REE),

$$\frac{\partial \left(\frac{1}{2} v_m^2 \right)}{\partial r} = \frac{\partial h_o}{\partial r} - T \frac{\partial S}{\partial r} - \frac{1}{2r^2} \frac{\partial (r^2 v_\theta^2)}{\partial r} + \frac{v_m^2}{R_c} \sin \phi + v_m \frac{\partial v_m}{\partial m} \cos \phi, \quad (5.3)$$

for v_m , along each QO while proceeding one QO at a time. The first three terms on the right-hand-side of the above equation can be evaluated at each node along the QO using the conditions from the previous iteration (or initial conditions), as can R_c , ϕ , and $\partial v_m / \partial m$. It should be noted here that all derivatives are approximated by three-point central-, forward-, and backward differences depending on node location. The reader is referred to Appendix D for details of these finite difference approximations. Now, the REE can be expressed as

$$\frac{dv_m}{dr} = \frac{1}{v_m} \left(\frac{dh_o}{dr} - T \frac{dS}{dr} - \frac{v_\theta}{r} \frac{d(r v_\theta)}{dr} \right) + \frac{v_m}{R_c} \sin \phi + \frac{dv_m}{dm} \cos \phi \quad (7.6)$$

or simply as

$$\frac{dv_m}{dr} = f(r, v_m);$$

a first-order ordinary differential equation. Ordinary derivatives replace the partial derivatives in

$$v_0 = \frac{\chi_i + k_1}{r_{i+1}} .$$

Once k_2 has been calculated along the QO, χ and subsequently v_0 can be updated at QO $i+1$ with the use of Equations (7.13) and (7.14).

After the tangential velocity components have been updated at QO $i+1$, it is possible to update entropy along this QO. This can be done by integrating Equation (7.10) from the QO i to QO $i+1$ for all streamlines. This is also done using a second order Runge-Kutta method. The recursive formulae are

$$S_{i+1} = S_i + \frac{1}{2}(k_1 + k_2) ,$$

where

$$k_1 = \Delta m_i f_1(m_i) ,$$

$$k_2 = \Delta m_i f_1(m_{i+1}) ,$$

and

$$\Delta m_i = m_{i+1} - m_i .$$

In a similar manner, stagnation enthalpy can be updated at QO $i+1$. Equation (7.12) is solved using

$$h_{0,i+1} = h_{0,i} + \frac{1}{2}(k_1 + k_2) ,$$

where

$$k_1 = \Delta m_i f_3(m_i) ,$$

$$k_2 = \Delta m_i f_3(m_{i+1}) ,$$

and

second-order accuracy in integrating the continuity equation an integration scheme based on Simpson's one-third rule was developed and is presented in Appendix E. Subroutine M-LOW utilizes this method in which the mass flow rate across the QO is approximated by

$$\dot{m} = 2\pi \sum_{k=1}^{n/2} \frac{a_k}{3} (r_{2k+1}^3 - r_{2k-1}^3) + \frac{b_k}{2} (r_{2k+1}^2 - r_{2k-1}^2) + c_k (r_{2k+1} - r_{2k-1}) , \quad (7.7)$$

where

$$a_k = \frac{((\rho v_x r)_{2k+1} - (\rho v_x r)_{2k})(r_{2k} - r_{2k-1}) - ((\rho v_x r)_{2k} - (\rho v_x r)_{2k-1})(r_{2k+1} - r_{2k})}{(r_{2k+1}^2 - r_{2k}^2)(r_{2k} - r_{2k-1}) - (r_{2k}^2 - r_{2k-1}^2)(r_{2k+1} - r_{2k})} ,$$

$$b_k = \frac{((\rho v_x r)_{2k+1} - (\rho v_x r)_{2k})(r_{2k}^2 - r_{2k-1}^2) - ((\rho v_x r)_{2k} - (\rho v_x r)_{2k-1})(r_{2k+1}^2 - r_{2k}^2)}{(r_{2k}^2 - r_{2k-1}^2)(r_{2k+1} - r_{2k}) - (r_{2k+1}^2 - r_{2k}^2)(r_{2k} - r_{2k-1})} ,$$

$$c_k = (\rho v_x r)_{2k-1} - a_k r_{2k-1}^2 - b_k r_{2k-1} ,$$

and n is the total number of streamtubes (equal to $SLMAX - 1$).

If the above calculated mass flow rate, $\dot{m}$, is not within a certain fraction (say $\pm 1\%$) of the specified mass flow rate, $\dot{m}_p$, then all values of v_m along the QO are adjusted by a constant component of axial velocity. The change in axial velocity component is calculated using the second-order integration method described in Appendix E, and is given by

$$\Delta v_x = \frac{\dot{m}_p - \dot{m}}{2\pi \sum_{k=1}^{n/2} \frac{a_k}{3} (r_{2k+1}^3 - r_{2k-1}^3) + \frac{b_k}{2} (r_{2k+1}^2 - r_{2k-1}^2) + c_k (r_{2k+1} - r_{2k-1})} ,$$

where a_k , b_k , and c_k are now given by

$$a_k = \frac{((\rho r)_{2k+1} - (\rho r)_{2k})(r_{2k} - r_{2k-1}) - ((\rho r)_{2k} - (\rho r)_{2k-1})(r_{2k+1} - r_{2k})}{(r_{2k+1}^2 - r_{2k}^2)(r_{2k} - r_{2k-1}) - (r_{2k}^2 - r_{2k-1}^2)(r_{2k+1} - r_{2k})} ,$$

$$b_k = \frac{((\rho r)_{2k+1} - (\rho r)_{2k})(r_{2k}^2 - r_{2k-1}^2) - ((\rho r)_{2k} - (\rho r)_{2k-1})(r_{2k+1}^2 - r_{2k}^2)}{(r_{2k}^2 - r_{2k-1}^2)(r_{2k+1} - r_{2k}) - (r_{2k+1}^2 - r_{2k}^2)(r_{2k} - r_{2k-1})},$$

and

$$c_k = (\rho r)_{2k-1} - a_k r_{2k-1}^2 - b_k r_{2k-1}.$$

There now exists an updated value of v_m at the hub so that the REE can be integrated along the QO once again. This iterative routine is continued until the difference between the calculated and specified mass flow rates falls within a certain tolerance ($\pm 1\%$ of the specified mass flow rate). The method is applied to each QO until the entire meridional velocity field has been updated.

7.4 Adjustment of Streamlines

This section gives details of the technique used to shift the streamline-QO intersection points. The adjustment of streamlines represents the outer iterative loop in the streamline curvature method and is outlined in Step 5 of Section 5.2.2. Subroutine ADJUST contains the code for this procedure.

After updating values of v_m (and subsequently v_r) in the entire meridional flow field, the total mass flow across each QO is correct. However, now the mass flow rate in each axisymmetric streamtube is no longer equal (in general). The condition governing streamline adjustment is that each streamtube contain the same mass flow and in order to make this so it is necessary to shift the positions of the streamlines at each QO. The new intersection points along a particular QO are found by integrating the mass flow from hub to casing using Simpson's one-third rule and a step-size, Δr , which is much smaller than the average streamtube height (ie. outer radius minus inner radius). The procedure for doing this is as follows.

For a particular QO, a series of second-order polynomials is first passed through data

points of the form $(r, \rho v_z r)$, as was done to calculate the total mass flow rate across the QO. The polynomial coefficients a_k , b_k , and c_k , as given by Equation (7.7), are stored for all of the curves. Now the quantity $\rho v_z r$ is continuously defined along the QO by

$$\rho v_z r(r) = a_k r^2 + b_k r + c_k, \quad (7.8)$$

where k ranges from 1 to $n/2$; n being the number of streamtubes. Each value of k is only valid over a certain range of r along the QO such that

$$r_{2k-1} < r < r_{2k+1}.$$

Here, the subscripts on the limiting values of r correspond to the streamline numbers (indicated by j in Figure 7.1)

Now, in re-positioning the streamlines along the QO, the inner and outer radii of the new streamtubes (r_i and r_o , respectively) are such that

$$\dot{m}_{st} = 2\pi \int_{r_i}^{r_o} \rho v_z r dr, \quad (7.9)$$

where $\dot{m}_{st}$ is the mass flow rate for each streamtube. The radii of the new streamline-QO intersection points are found along the QO by starting at the hub (setting r_i equal to the hub radius) and integrating Equation (7.9) along the QO using Simpson's one-third rule, incrementing r by $2\Delta r$ at a time, until the mass flow rate is within a certain fraction of that specified for a streamtube. This gives the value of r_o for the first streamtube. For the second streamtube, r_i is set to the value of r_o found for the first streamtube and Equation (7.9) is again integrated until the outer radius of the second streamtube is found. This process continues until the radii of all streamtubes are found. The integrand of Equation (7.9) is given by Equation (7.8) and Δr , as mentioned previously, is a small fraction of the average streamtube height.

Recall that, in order to maintain stability in the solution, changes in streamline position must be damped. Thus, the actual radii of the new streamline-QO intersection points are given by

$$r_{actual} = r_{old} + \omega(r_{new} - r_{old}) ,$$

where r_{old} and r_{new} represent streamline radii from the previous and current iterations, respectively.

Before proceeding to the next QO, new values of density and axial velocity must be found at the new node locations. This was done by fitting the old distributions of these quantities with second-order polynomials and interpolating values of ρ and v_x at the new node locations.

7.5 Solution of the Mixing Equations

This section presents the specific details of the method used to solve the mixing equations developed in Section 6.1.2. The equations to be solved are (6.3), (6.4), and (6.5). Since they are quite lengthy, these equations will not be reproduced here and the reader is referred to Section 6.1.2 or Appendix C for their presentation.

In the order presented above, the mixing equations give the streamwise changes in entropy, angular momentum, and stagnation enthalpy along streamlines due to the radial diffusion of heat and momentum. As described in Section 6.1.3, they replace the equations for isentropic flow along streamlines in the basic streamline curvature method. Since many terms in the mixing equations can be evaluated using properties from the previous iteration, all may be treated as ordinary differential equations and solved in a manner similar to that used in Section 7.3 to solve the radial equilibrium equation. Equations (6.3), (6.4), and (6.5) may be represented as

$$\frac{dS}{dm} = f_1(m) , \quad (7.10)$$

$$\frac{d(rv_\theta)}{dm} = f_2(m, v_\theta) , \quad (7.11)$$

and

$$\frac{dh_o}{dm} = f_3(m) , \quad (7.12)$$

respectively. To reduce computing time, all three equations are solved simultaneously during a single pass through the computational grid. Equation (7.11) is solved first as described below.

Equation (7.11) is a first-order ordinary differential equation giving the streamwise gradient of angular momentum at any node. Note that $f_2(m, v_\theta)$ is not a function of r since the streamline positions have been fixed for the integration. Since the value of rv_θ can be calculated from the given inlet conditions, the solution of Equation (7.11) represents another initial value problem. Again, since $f_2(m, v_\theta)$ can only be evaluated at discrete points along streamlines a second-order Runge-Kutta method is used. Letting $\chi = rv_\theta$, the recursive formulae for integrating Equation (7.11) along a streamline are

$$\chi_{i+1} = \chi_i + \frac{1}{2}(k_1 + k_2) , \quad (7.13)$$

where

$$k_1 = \Delta m_i f_2(m_i, \chi_i) ,$$

$$k_2 = \Delta m_i f_2(m_{i+1}, \chi_i + k_1) ,$$

$$\Delta m_i = m_{i+1} - m_i ,$$

and $i = 1, 2, \dots, QOMAX-1$. The updated tangential velocity components are then given by

$$v_{\theta, i+1} = \frac{\chi_{i+1}}{r_{i+1}} . \quad (7.14)$$

Now, it is important to note that $f_2(m, v_\theta)$ contains radial derivatives of v_θ . These derivatives are calculated using finite difference approximations which use values of v_θ from adjacent streamlines. Thus, in order to calculate k_2 in the above Runge-Kutta formula, it is necessary to solve Equation (7.11) for all streamlines simultaneously. This means that for each value of i (each QO), k_1 must be calculated for all values of j (all streamlines). Before k_2 can be evaluated, for all j , the radial derivatives of v_θ must be calculated, for all j , using values of v_θ based on k_1 . These v_θ values are first estimates at QO $i+1$ and can be expressed as

$$v_\theta = \frac{\chi_i + k_1}{r_{i+1}} .$$

Once k_2 has been calculated along the QO, χ and subsequently v_θ can be updated at QO $i+1$ with the use of Equations (7.13) and (7.14).

After the tangential velocity components have been updated at QO $i+1$, it is possible to update entropy along this QO. This can be done by integrating Equation (7.10) from the QO i to QO $i+1$ for all streamlines. This is also done using a second order Runge-Kutta method. The recursive formulae are

$$S_{i+1} = S_i + \frac{1}{2}(k_1 + k_2) ,$$

where

$$k_1 = \Delta m_i f_1(m_i) ,$$

$$k_2 = \Delta m_i f_1(m_{i+1}) ,$$

and

$$\Delta m_i = m_{i+1} - m_i .$$

In a similar manner, stagnation enthalpy can be updated at QO $i+1$. Equation (7.12) is solved using

$$h_{o,i+1} = h_{o,i} + \frac{1}{2}(k_1 + k_2) ,$$

where

$$k_1 = \Delta m_i f_3(m_i) ,$$

$$k_2 = \Delta m_i f_3(m_{i+1}) ,$$

and

$$\Delta m_i = m_{i+1} - m_i.$$

Therefore, it can be seen that tangential velocity, entropy, and stagnation enthalpy are updated for all streamlines proceeding one QO at a time from inlet to exit, during a single pass through the computational grid.

It is also possible, in this single pass, to update temperature, pressure, and density using

$$T_{i,j} = \frac{1}{c_p} \left(h_{o,i,j} - \frac{V_{i,j}^2}{2} \right),$$

$$p_{i+1,j} = p_{i,j} \exp \left[\frac{c_p \ln \left(\frac{T_{i+1,j}}{T_{i,j}} \right) - S_{i+1,j} + S_{i,j}}{R} \right],$$

and

$$\rho_{i,j} = \frac{p_{i,j}}{RT_{i,j}},$$

respectively.

7.5.1 Applying the Boundary Conditions

In order to implement the boundary conditions of Section 6.1.4, the use of so called "phantom streamlines" was required. These are streamlines placed beyond the hub and casing surfaces and are only used in the computation of radial derivatives of temperature and velocity at these surfaces. The locations of the phantom streamlines are obtained by reflecting the streamlines adjacent to the solid surfaces across these surfaces. Values of temperature and fluid velocity at nodes along the phantom streamlines are set equal to corresponding values at adjacent nodes on the solid surfaces. In this manner, radial gradients of temperature and velocity can be approximated as zero at the hub and casing surfaces when three-point central-differences are used in calculating these surface derivatives.

$$\frac{V}{V_i} = \frac{\ln\left(\frac{z/l}{2.02067 \times 10^{-5}}\right)}{9.80455}$$

fitted the data quite well up to the point where V/V_i equalled unity. It should be noted that all inlet velocity profiles used in the computations of the present study were assumed symmetric about the blade mid-span. Thus, the computed exit flowfield is symmetric about the passage mid-span and results are only presented for half of the passage span.

The density of the fluid was estimated using the Reynolds number quoted in Table 8.2. Once the fluid kinematic viscosity was known the density was found using standard property tables for air. The fluid density used for both inlet boundary layer cases was 1.22 kg/m^3 .

The purpose of the secondary flow computation developed in the present study was to predict the secondary flowfield at the blade passage exit plane with reasonable accuracy. In order to calculate an appropriate distribution of the convective mixing coefficient, ϵ_c , along the span of the blade, reasonable secondary velocities and passage vortex position must be predicted. Figure 8.3.a shows the secondary flow velocity vectors measured by Marchal and Sieverding (1977) near the blade trailing edge plane ($x/c_t=1.01$) of the nozzle cascade. The results were only available for the thin inlet boundary layer case and unfortunately no reference scale for the vector magnitudes was given. The passage vortex can clearly be seen near the wall of the passage. Figure 8.3.b shows the secondary velocity vectors computed at the passage exit plane, using program BLADE, for the same test conditions. Obviously no comparison of Figures 8.3.a and 8.3.b can be made in terms of the magnitudes of the secondary velocities; however, some qualitative comparisons can still be made.

The two figures are spatially equivalent and thus it can be said that the computed passage vortex is approximately the same size and in roughly the same location as that found in the experimental investigation of Marchal and Sieverding. The cores of both vortices are approximately the same distance from the endwall; however, the computed vortex seems to be located a little closer to the suction side of the passage. Convection of the passage vortex towards the blade suction surface is to be expected due to the pressure gradient within the passage.

Blade chord (c)	120 mm
Pitch to chord ratio (a/c)	0.725
Stagger angle (γ)	42.5°
Inlet blade angle (α'_i)	-10.0°
$\cos^{-1}(o/a)$	67.7°
Blade span (l)	100 mm
Blade aspect ratio (l/c)	0.83

Table 8.1 Turbine nozzle cascade geometry of Marchal and Sieverding (1977).

boundary layer", was allowed to develop naturally between the settling chamber and the cascade. The second inlet boundary layer used was one which had been artificially thickened with the use of vortex generators placed upstream of the cascade at the outlet of the settling chamber. This second case will be referred to as the "thick boundary layer" case. The test conditions for both inlet boundary layer cases are presented in Table 8.2. The turbine nozzle blade profiles and the appropriate nomenclature for the cascade are presented in Figure 8.1 while the inlet velocity profiles for the two boundary layer cases are presented in Figure 8.2.

Not all of the information given in Tables 8.1 and 8.2 was needed to formulate input data for this test case. The only pieces of information required to run program BLADE are the blade passage dimensions, inlet and exit flow angles, an inlet velocity profile, and the fluid density. The required passage dimensions were easily calculated with the information given in Table 8.1 while the flow angles used were those given in Table 8.2. In order to construct an inlet velocity profile it was necessary to first estimate the value of, V_i , the inlet fluid velocity at mid-span. This was done using the value of the exit fluid velocity at mid-span, V_e , and the flow angles. Values of 12.2 m/s and 12.0 m/s were used in the computations for the thin and thick inlet boundary layer cases respectively.

To construct the inlet velocity profile for the thin boundary layer case a power-law profile was assumed. In the boundary layer, this profile takes the form

$$\frac{V}{V_i} = \left(\frac{z}{\delta_{99\%}} \right)^d \quad (8.1)$$

where the power-law exponent d was given a value of 0.151 as suggested by Gregory-Smith and Okan (1991). The power-law profile was used for simplicity and because a value of the power-law exponent was available in the literature for this particular investigation. The shortcomings of the power-law profile will be discussed later in this section. In order to construct the inlet velocity profile for the thick boundary layer case, it was necessary to fit a curve to the experimental data presented in Figure 8.2. The logarithmic curve given by

Test parameter	Thin BL	Thick BL
BL thickness ($\delta_{99\%}$)	8.4 mm	31.1 mm
Displacement thickness (δ^*)	1.10 mm	3.75 mm
Shape factor (H)	1.48	1.36
Inlet total Pressure (p_o)	70 mm H ₂ O	70 mm H ₂ O
Inlet total Temperature (T_o)	291 K	291 K
Inlet flow angle (α_i)	0.0°	0.0°
Exit flow angle (α_e)	67.8°	68.3°
Exit velocity (V_e)	32.2 m/s	32.5 m/s
Exit Mach number (M_e)	0.1	0.1
Reynolds number (cV_e/ν)	2.61×10^5	2.64×10^5
Turbulence intensity (u'/u)	0.5%	0.5%

Table 8.2 Inlet boundary layer and test conditions of Marchal and Sieverding (1977).

Figure 8.1 Blade profile and nomenclature for nozzle cascade of Marchal and Sieverding (1977).

Figure 8.2 Measured inlet boundary layer profiles of Marchal and Sieverding (1977).

$$\frac{V}{V_i} = \frac{\ln\left(\frac{z/l}{2.02067 \times 10^{-5}}\right)}{9.80455}$$

fitted the data quite well up to the point where V/V_i equalled unity. It should be noted that all inlet velocity profiles used in the computations of the present study were assumed symmetric about the blade mid-span. Thus, the computed exit flowfield is symmetric about the passage mid-span and results are only presented for half of the passage span.

The density of the fluid was estimated using the Reynolds number quoted in Table 8.2. Once the fluid kinematic viscosity was known the density was found using standard property tables for air. The fluid density used for both inlet boundary layer cases was 1.22 kg/m^3 .

The purpose of the secondary flow computation developed in the present study was to predict the secondary flowfield at the blade passage exit plane with reasonable accuracy. In order to calculate an appropriate distribution of the convective mixing coefficient, ϵ_c , along the span of the blade, reasonable secondary velocities and passage vortex position must be predicted. Figure 8.3.a shows the secondary flow velocity vectors measured by Marchal and Sieverding (1977) near the blade trailing edge plane ($x/c_t \approx 1.01$) of the nozzle cascade. The results were only available for the thin inlet boundary layer case and unfortunately no reference scale for the vector magnitudes was given. The passage vortex can clearly be seen near the wall of the passage. Figure 8.3.b shows the secondary velocity vectors computed at the passage exit plane, using program BLADE, for the same test conditions. Obviously no comparison of Figures 8.3.a and 8.3.b can be made in terms of the magnitudes of the secondary velocities; however, some qualitative comparisons can still be made.

The two figures are spatially equivalent and thus it can be said that the computed passage vortex is approximately the same size and in roughly the same location as that found in the experimental investigation of Marchal and Sieverding. The cores of both vortices are approximately the same distance from the endwall; however, the computed vortex seems to be located a little closer to the suction side of the passage. Convection of the passage vortex towards the blade suction surface is to be expected due to the pressure gradient within the passage.

Figure 8.3.a Measured secondary flow velocity vectors near trailing edge plane of Marchal and Sieverding cascade (reproduced from Marchal and Sieverding, 1977).

Figure 8.3.b Computed secondary flow velocity vectors at the blade passage exit plane for the cascade of Marchal and Sieverding ($V_i \approx 12.2$ m/s).

However, since the convective mixing coefficient is calculated using pitchwise average velocities, the vortex location with respect to the endwall is more important than its pitchwise location.

It is well known that the presence of secondary flow velocities at the exit of a blade passage affects the exit flow angle, α_e . The pitchwise velocities associated with the passage vortex result in overturning of the flow at the endwall and underturning away from the endwall. This deviation in exit flow angle is clearly illustrated in Figures 8.4.a and 8.4.b for the thin and thick inlet boundary layer cases, respectively. Both figures present the experimental results from Marchal and Sieverding (1977) along with computational results obtained from the present study. The exit flow angle deviation is given by

$$\overline{\Delta\alpha}^m = \overline{\alpha}^m - \overline{\alpha}_e^m$$

for cascades with a positive exit flow angle at mid-span and by

$$\overline{\Delta\alpha}^m = \overline{\alpha}_e^m - \overline{\alpha}^m$$

for cascades with a negative exit flow angle at mid-span. It should be noted that all angles here are pitchwise mass-averaged quantities.

Consider Figure 8.4.a in which two curves are illustrated for the present computation; one using a coarse computational grid (27 pitchwise points $\times$ 30 spanwise points) and one for a more dense computational grid (91 $\times$ 104). The computed curve using the coarse grid agrees quite well with the experimental results of Marchal and Sieverding while the curve using the dense grid predicts a vortex with greater secondary velocities than those seen in the experiments. The spanwise location of the vortex centre is at the point where the exit flow angle deviation becomes zero between the regions of overturning and underturning. Both curves predict the spanwise location of the vortex quite well. Possible reasons for the differences of the two computational results will be discussed later in Section 8.1.2. The same trend in the computational results can be seen in Figure 8.4.b where use of the dense computational grid has produced more severe flow angle deviation. In this case, however, both computations have predicted a larger vortex with greater pitchwise velocities than that seen in the experiments.

Figure 8.4.a Comparison of exit flow angle deviation for Marchal and Sieverding cascade (thin inlet boundary layer).

Figure 8.4.b Comparison of exit flow angle deviation for Marchal and Sieverding cascade (thick inlet boundary layer).

The second test case used to verify program **BLADE** was developed from the experiments of Walsh and Gregory-Smith (1990). They investigated how inlet boundary layer skew affected the secondary loss and vorticity growth in a large-scale linear cascade of turbine rotor blades. They used boundary layers with positive, zero, and negative skew. A skewed boundary layer is three-dimensional in nature, and since the secondary flow code was only designed to handle two-dimensional inlet boundary layers, only the results of the zero skew case were used.

The cascade of Walsh and Gregory-Smith was chosen as the second test case because of its large flow turning angle and span-to-pitch ratio (passage aspect ratio). This provided a contrast to the lower turning, small aspect ratio cascade of Marchal and Sieverding (1977). The blades investigated by Marchal and Sieverding are likely to be found in a first stage turbine whereas those of Walsh and Gregory-Smith are more likely to be based on rotor blades found in later stage lower pressure turbines. The geometry of the rotor blade cascade of Walsh and Gregory-Smith is given in Table 8.3 and pertinent inlet boundary layer parameters are given in Table 8.4. Figure 8.5 illustrates the cascade blade profiles and shows the locations of various dimensions.

To formulate the required inlet data for the secondary flow computation, the blade passage dimensions given in Table 8.3 were used as were the inlet and exit angles. For the computation it is more appropriate to use flow angles but since the exit flow angle was not given in the literature, the blade exit angle, α' , was used. The fluid density was assumed to be that of air at 1 atmosphere pressure and 20°C and had a value of 1.20 kg/m³. To construct the velocity profile the mid-span velocity was given a value of 13 m/s. This value was not given in the literature but was estimated using the Reynolds number based on exit velocity, a fluid kinematic viscosity of 1.51×10^{-5} m²/s (air @ 1 atm and 20°C), and the flow turning angles.

As with the first test case using the thin inlet boundary layer, the inlet velocity profile was constructed using the power-law (Equation 8.1). In this case the power-law exponent, d , and boundary layer thickness, $\delta_{99\%}$, were given values of 0.147 and 39.6 mm respectively as indicated in Table 8.4. This inlet velocity profile is illustrated in dimensionless form in Figure 8.6. It can be noticed that the profile is not smooth at the free-stream edge of the boundary layer. This means that the spanwise velocity gradient here abruptly reaches a zero value. Of greater concern is the fact that the spanwise velocity gradient is infinite at the wall. The velocity gradient at any

Flow inlet angle (α_i)	42.75°
Blade exit angle (α'_e)	-68.60°
Blade chord (c)	224 mm
Blade axial chord (c_x)	181 mm
Blade span (l)	400 mm
Blade pitch (a)	191 mm
Reynolds number (cV_∞/ν)	4.0×10^5

Table 8.3 Cascade geometry of Walsh and Gregory-Smith (1990).

Power-law exponent (d)	0.147
BL thickness ($\delta_{99\%}$)	39.6 mm
Displacement thickness (δ^*)	6.0 mm
Momentum thickness (θ)	3.6 mm
Shape factor (H)	1.7

Table 8.4 Inlet boundary layer data of Walsh and Gregory-Smith (1990).

Figure 8.5 Blade profile and nomenclature for rotor cascade of Walsh and Gregory-Smith (1990).

Figure 8.6 Inlet boundary layer profile based on power-law (as indicated by Walsh and Gregory-Smith, 1990).

point in the boundary layer is, of course, equal to the vorticity at that point. In a real boundary layer the vorticity is finite at the wall and smoothly approaches zero at the boundary layer edge. These are short-comings of the power-law profile but their impact is thought to be reduced by the fact that, in the present study, spanwise gradients in the velocity profile were calculated using finite difference formulae.

Figure 8.7.a shows the measured secondary flow velocity vectors near the blade trailing edge plane ($x/c_t \approx 0.97$) of the cascade of Walsh and Gregory-Smith (1990). Figure 8.7.b shows the computed secondary velocity vectors for the same conditions. Again, the two figures are spatially equivalent so that the passage vortex locations can be directly compared. Both the measured and computed vortices are located at approximately the same pitchwise position; however, the measured vortex appears slightly larger and thus its centre is located further from the endwall than the computed vortex. This discrepancy in vortex size can be attributed to viscous effects which the secondary flow computation does not model. Fluid viscosity would tend to diffuse the vortex laterally as it develops in the blade passage. As more free-stream fluid is entrained in the vortex it grows in diameter causing its centre to move away from the endwall. The migration of the vortices toward the blade suction surface is again caused by the pressure gradient in the passage. The effect is more pronounced in this test case than it was in the first due to the larger turning angle of the Walsh and Gregory-Smith cascade.

Both Figures 8.7.a and 8.7.b have reference velocity vectors so that the magnitudes of the velocity vectors can be compared. It appears that the velocities in the computed results are slightly less than those measured. A possible explanation for this discrepancy is that the passage vortex in the measured results has merged with the pressure side leg of the leading edge vortex (horseshoe vortex) creating a single vortex with greater secondary velocities. The secondary flow computation developed in the present study only predicts the development of the passage vortex. Still, the agreement is quite reasonable.

A comparison of exit flow angle deviation for this test case is presented in Figure 8.8. The computed results shown are typical of those obtained with quite moderate grid density (29 pitchwise points $\times$ 62 spanwise points). Unlike the first test case, the results did not vary greatly as the grid density was changed. Perhaps greater variations would have been seen if even more

Figure 8.7.a Measured secondary flow velocity vectors at trailing edge plane of Walsh and Gregory-Smith cascade (reproduced from Walsh and Gregory-Smith, 1990).

Figure 8.7.b Computed secondary velocity vectors at the blade passage exit plane for the cascade of Walsh and Gregory-Smith ($V_i \approx 13.4$ m/s).

Figure 8.8 Comparison of exit flow angle deviation for Walsh and Gregory-Smith cascade.

Figure 8.9 Comparison of pitchwise mass-averaged dimensionless streamwise vorticity at blade passage exit plane for Walsh and Gregory-Smith cascade.

coarse grids were used (the coarsest grid used was 21 pitchwise points $\times$ 46 spanwise points). Regardless, the computation tends to over-predict the deviation at the endwall and under-predict the deviation away from the endwall. However, the location of the vortex centre seems to be quite reasonably predicted here in terms of its distance from the endwall. The peculiar shape produced by the experimental data points suggests that there are other mechanisms producing secondary flows in this cascade, most likely the leading edge horseshoe vortex.

Figure 8.9 shows a comparison of the pitchwise mass-averaged dimensionless streamwise vorticity for the experiments of Walsh and Gregory-Smith (1990) and the computations of the present study. Again, the results shown are for the blade passage exit plane. The dimensionless streamwise vorticity, ξ_s^* , is given by

$$\xi_s^* = \frac{\xi_s a}{V_i}$$

and is negative for the sense of passage vortex rotation shown in Figures 8.7.a and 8.7.b. Here a is the blade pitch. Streamwise vorticity is an important quantity to examine because, in the secondary flow computation, it determines the secondary flow velocities. In experimental results, however, the streamwise vorticity is calculated by differentiating the measured secondary flow velocities. Referring to Figure 8.9, it can be seen that the maximum values of streamwise vorticity for the experiment and the computation are virtually the same. However, the spanwise location of these maxima differ. The maximum streamwise vorticity for the experimental data is located away from the endwall while the maximum for the computation is located nearly on the endwall. This type of result is somewhat expected given the different spanwise locations of the measured and computed passage vortices illustrated in Figures 8.7.a and 8.7.b respectively. The peak vorticity being located near the endwall in the computed results could also be attributed to the use of the power-law inlet profile with its theoretically infinite vorticity in this region. Referring to Figure 8.9 again, one can also notice that there is a small region of positive streamwise vorticity in the experimental data of Walsh and Gregory-Smith. It is possible that this positive streamwise vorticity identifies the location of the suction side leg of the leading edge horseshoe vortex which has a rotation opposite to that of the passage vortex.

Figure 8.10.a illustrates the dimensionless streamwise vorticity distribution near the blade

Figure 8.10.a Contours of dimensionless streamwise vorticity near the blade passage exit plane of the Walsh and Gregory-Smith cascade (reproduced from Walsh and Gregory-Smith, 1990).

Figure 8.10.b Computed contours of dimensionless streamwise vorticity at the blade passage exit plane for the cascade of Walsh and Gregory-Smith.

trailing edge plane for the experiments of Walsh and Gregory-Smith. The contour plot has been reproduced from Walsh and Gregory-Smith (1990) and, as explained therein, was created by fitting a bicubic spline surface to the velocity data. Values of streamwise vorticity were then obtained by numerical differentiation of the resulting surface. They urge caution in using the absolute values from the plot, especially in regions where high velocity gradients are present. Figure 8.10.b shows the computed dimensionless streamwise vorticity distribution at the blade passage exit plane. As mentioned previously, the secondary flow computation uses the distributed secondary vorticity to calculate the secondary flow velocities and thus ξ^* , is easily available at the exit plane. Figure 8.10.b was created by a piece of graphing software known as EasyPlot™ and the author is unaware of the technique used to generate the contours. Despite this, several similarities between Figures 8.10.a and 8.10.b can be noticed. The region of non-zero streamwise vorticity for both plots is approximately the same size indicating that similar sized vortices were measured and computed. Again, the two plots are spatially equivalent so that spatial features can be directly compared. The convection of fluid with high streamwise vorticity towards the suction side of the passage and up the blade suction surface (indicated by S.S. in Figure 8.10.a) is noticeable and a concentrated region of maximum streamwise vorticity is located in approximately the same location for both plots. The maximum value of streamwise vorticity calculated from the experimental data is about 30% greater than computed by the secondary flows code. This, however, is consistent with the stronger vortex which was measured by Walsh and Gregory-Smith (1990) and presented in Figure 8.7.a.

Looking at the results from the two test cases just discussed it can be concluded that the secondary flow code developed for the present study gives reasonable values of secondary flow velocities. The computations also predict with reasonable accuracy the placement of the passage vortices and thus give a satisfactory radial distribution of secondary velocities from which the convective mixing coefficient, ϵ_c , can be calculated. The next section presents the convective mixing coefficient distributions which were calculated from the secondary flow computations using the two test cases just described.

8.1.1 Convective Mixing Coefficients

Once the secondary flow velocities were calculated for the test cases presented in Section 8.1, spanwise distributions of the convective mixing coefficient, ϵ_c , were calculated using the expression

$$\epsilon_c(x, r) = \left[\left(x + \frac{c_x}{2} \right) \left(1 - \frac{1}{5l} \left(x + \frac{c_x}{2} \right) \right) \right] \left[\left(\frac{w}{u} \right)^2 \left(1 - \left(\frac{x}{5l} \right)^4 \right) \right] \left[\frac{\overline{u_r^2}}{u} \right] \quad (6.15)$$

developed in Section 6.3.2. These distributions were calculated at the exit planes of the cascades and thus a value of $x=0$ was used in evaluating Equation (6.15). When presented in the open literature, values of mixing coefficients, whether convective or turbulent, are usually nondimensionalized by axial stage length, L_s , and mean axial velocity, u . In order to make comparisons with published values, the convective mixing coefficients calculated in the present study have been made dimensionless using twice the corresponding blade axial chord and the mass-averaged axial velocity at the cascade exit. Thus, it has been assumed that an appropriate stage length is approximately equal to twice the blade axial chord. The spanwise distributions of these dimensionless convective mixing coefficients are presented in Figures 8.11.a and 8.11.b using linear and semi-logarithmic scales, respectively.

Examining Figure 8.11.a it is readily noticed that the phenomenon of convective radial mixing due to the passage vortex is significant only in the endwall region of the passage. No doubt the effects of turbulent mixing will dominate in the interior of the flow towards mid-span. It can also be seen that values of the convective mixing coefficient are much greater for the cascade of Walsh and Gregory-Smith (1990) than for the cascade of Marchal and Sieverding (1977). This is obviously the result of the larger flow turning angle, and subsequently larger radial velocities, in the Walsh and Gregory-Smith cascade. This emphasizes the need to include the effects of convective radial mixing in turbine flow calculations where flow turning angles are much larger than those found in compressors. Even the relatively low flow turning angle of the Marchal and Sieverding cascade would be considered quite extreme for a compressor blade. The computed results in Figure 8.11.a are also compared with values measured in a multistage turbine by Lewis (1993(a)). The lines shown represent typical values found in a lightly loaded (LL) and

Figure 8.11.a Comparison of dimensionless convective mixing coefficients for the two cascade test cases (linear axes).

Figure 8.11.b Comparison of dimensionless convective mixing coefficients for the two cascade test cases (semi-log axes).

more heavily loaded (HL) turbines and lie within the range of values computed for the cascades.

Figure 8.11.b presents the same results as those shown in Figure 8.11.a except that the mixing coefficient is plotted on a logarithmic scale. In this figure it can be seen that the value of the convective mixing coefficient for the Marchal and Sieverding cascade actually surpasses that for the Walsh and Gregory-Smith cascade at about $z/l=0.28$. Thus, convective mixing is significant over a greater portion of the blade span for the Marchal and Sieverding cascade than for the cascade of Walsh and Gregory-Smith. This is solely due to the lower aspect ratio of the Marchal and Sieverding cascade and emphasizes the need to include convective radial mixing in turbine flow calculations for machines with low aspect ratios.

8.1.2 Parametric Study for Program BLADE

This section describes how several input parameters to the secondary flow code were systematically varied to see the effects on the results obtained. The test case used in the parametric study was that developed from Marchal and Sieverding (1977). It should be mentioned that program BLADE was compiled using the Sun FORTRAN optimizing compiler and run on a dedicated Sun SPARCstation 2.

The first parameter to be varied in the study was the number of computational regions used to divide the blade passage in the axial direction. As described previously in Chapter 3, each computational region has the same axial length and flow turning angle. Also, each computational region is further divided into four sub-regions also of equal axial length and flow turning angle. Figure 3.6 illustrates both the computational regions and sub-regions in a hypothetical blade passage.

Increasing the number of computational regions should, in theory, make the secondary flow computation more accurate. This is due to the fact that increasing the number of computational regions reduces the flow angle turned by each region. Equation (3.73) was developed under the assumption of a small flow turning angle and is used as a first estimate of the streamwise vorticity at the exit plane to the computational region. Thus using this equation

in regions of lower flow turning should result in more accurate results. The drawback to using more computational regions is, of course, an increase in computation time.

Figure 8.12 illustrates how the computation time increases as the number of computational regions (CRs) is increased. Here, the number of grid points in any particular y - z plane has remained constant with 41 pitchwise points $\times$ 42 spanwise points. As would be expected, the computation time increases linearly with the number of CRs used. Figure 8.13 shows how the pitchwise mass-averaged exit flow angle deviation changes as the number of computational regions is increased. The change in the results from using three computational regions to using thirty is very small and it can be concluded that using three computational regions is sufficiently accurate in this case. It should be noted that the test case used here (Marchal and Sieverding, 1977) involves a fairly moderate flow turning angle and the results are expected to be more dramatic for higher turning cascades.

The other parameter varied in the study was the grid density in the computational planes (y - z planes). The number of grid points in both the y and z -directions was increased from very low values but always in such a way that the grid point spacing in the y and z -directions (Δy and Δz respectively) was approximately equal. This process was also expected to increase the accuracy of the computation, since all derivatives are calculated using various three-point finite-difference formulae, but at the expense of increased computation time.

Figure 8.14 shows how the computation time increased as a result of increasing the number of grid points in the computational planes, while using a fixed number of computational regions. In this case, computation times increased rapidly as a quadratic function of the total number of grid points used. Figure 8.15 illustrates how increasing the number of grid points affected the exit flow deviation angle. This effect was mentioned previously in Section 8.1 as the computational results of this test case were being compared with the experimental results of Marchal and Sieverding (1977). The dramatic difference in the results as the computational grid is refined can be attributed to the increase in the number of grid points that actually lie in the inlet boundary layer. As the number of points in the boundary layer increases, the points used to calculate the boundary layer vorticity become closer together and closer to the endwall. As mentioned before, the boundary layer is approximated with a power-law profile which has infinite

Figure 8.12 Computation time versus number of computational regions (test case 1).

Figure 8.13 The effect of the number of CRs used on exit flow angle deviation (test case 1).

Figure 8.14 Computation time versus number of grid points on intermediate plane (test case 1).

Figure 8.15 The effect of grid point density on exit flow angle deviation (test case 1).

vorticity at the wall. As more grid points approach the wall, larger values of vorticity result from the numerical differentiation of the velocity profile. This results in predicting passage vortices with increasing secondary velocities as the grid is refined. This is clearly illustrated in Figure 8.15 by the increased under- and overturning of the exit flow near the wall. This is not to say that using a better boundary layer profile would eliminate this effect. A more accurate boundary layer profile model will still have very high but finite vorticity at the wall and may reduce the effect seen by increasing the grid density.

Even though the results obtained using the denser computational grids are considered more "accurate", it is the results obtained using less dense grids which seem to better match the experimental results of Marchal and Sieverding. As mentioned previously, this is thought to be the result of numerical inaccuracies in estimating the boundary layer vorticity of the inlet flow. The end result looks similar to a viscous effect, creating smaller secondary velocities. If fluid viscosity was taken into account, the more "accurate" results might better match the experimental measurements. So while using fewer grid points results in reduced computation times and a better data match (for this case at least) there is still no basis for one to choose a coarse grid density in hopes of obtaining a better prediction.

8.2 Results from Annular Duct Flow Computation (DUCT)

This section presents the results obtained from program DUCT in which the flow in an annular diffuser is analyzed. The computed results are compared with those measured in experiments. As well, the same diffuser inlet flow was used in an annular duct of constant cross-sectional area to examine the separate effects of turbulent and convective mixing.

The ideal test case to verify the computations of program DUCT (in its present form) would be an annular flow downstream of a turbine stator blade row. Binder and Romey (1983) published data from such an experiment, however, their data was insufficient to formulate a test case and the original source of their data could not be found. Therefore, the test case used with program DUCT was that which closest resembled the ideal test case and was available in the open literature. The details of this test case and results obtained are presented in the following sub-

sections.

8.2.1 Test Case Description

The test case which was used to verify program DUCT was that of Quest (1990). Quest performed experimental investigations of flows in annular diffusers, the inlets of which were connected to the exit of a single-stage turbine. Consequently, this flow can be considered to be a realistic turbomachinery flow, containing the effects of both secondary flow and turbulence. Flowfield measurements at diffuser inlet and exit planes were carried out for different diffuser geometries and turbine operating points.

The cold-air turbine test rig on which Quest performed the experimental investigations is located at the Propulsion Institute of DFVLR at Cologne and is shown in Figure 8.16. This is the same facility on which Binder and Romey (1983) performed their experiments, only they took their measurements before the rotor had been installed. The test rig has a casing diameter of 400 mm and a variable inner diameter depending on which hub body is used. The turbine inlet conditions are limited by a maximum total pressure of 2 bars and a total temperature of 350 K while conditions at the diffuser exit are always atmospheric. Rotor shaft speed is limited to 8500 rpm by a water-brake allowing a maximum output power of 300 KW.

Quest (1990) presented data for two test cases which use the same diffuser geometry but different operating conditions for the turbine stage. He distinguishes the two by quoting the turbine exit (measuring plane 3=4) swirl angle at mean radius for each case. The test case chosen for the present study had a mean swirl angle of about 10°. The experimental data collected for this test case was carried out using a weakly loaded LP-turbine with a hub-to-tip ratio, v , of 0.525. The stator is of free-vortex design with a flow deflection of 58° at mean radius. The annular diffuser used was one with a constant inner radius of 105 mm and a casing divergence angle of 7°. This leads to a diffuser area ratio (exit area/inlet area) of 3.52 and a dimensionless diffuser length, L (see nomenclature), of 11.63. The inlet duct and diffuser geometry for this case are illustrated in Figure 8.17, where the casing divergence angle is given by θ and the diffuser axial length, denoted L_{ax} , has been calculated to be 1.10 m. The turbine stage operating conditions for

Figure 8.16 Cold-air turbine and diffuser test rig (reproduced from Quest, 1990).

Figure 8.17 Diffuser geometry and measuring planes (reproduced from Quest, 1990).

the test case are given in Table 8.5. In addition to this information, Quest presented tabulated data consisting of p_o^* , p^* , V_x^* , and α versus dimensionless channel height, h^* , for both inlet and exit planes of the diffuser. Here, the total pressure coefficient, p_o^* , is given by

$$p_o^* = \frac{p_o - p_a}{\frac{1}{2}\rho_1 V_1^2} ,$$

the static pressure coefficient, p^* , is given by

$$p^* = \frac{p - p_a}{\frac{1}{2}\rho_1 V_1^2} ,$$

the axial velocity coefficient, V_x^* , is given by

$$V_x^* = \frac{V_x}{V_{x_1}} ,$$

the dimensionless channel height is given by

$$h^* = \frac{r - r_{hub}}{r_{tip} - r_{hub}} ,$$

and the swirl angle, α , is given in degrees measured from the axial direction.

As input, program DUCT requires radial distributions of total pressure, total temperature, and swirl angle at the inlet plane of the diffuser. While total pressure and swirl angle could be directly obtained from the distributions given by Quest, the total temperature had to be estimated using

$$T_o = \frac{(\gamma - 1)V_x^2}{2\gamma R \left[1 - (p/p_o)^{\frac{\gamma-1}{\gamma}} \right] \cos^2 \alpha} \quad (8.2)$$

which assumes negligible radial velocities. In all calculations, the gas constant, R , was taken as 286 J/(kg·K), the specific heat, c_p , was taken as 1004 J/(kg·K), while the ratio of specific heats, γ , was taken as 1.4. Since the turbine exit Mach number was only 0.22, it was expected that

Rotor speed (n)	8576 rpm
Mass flow rate ($\dot{m}$)	7.61 kg/s
Stage pressure ratio (p_{o1}/p_{o4})	1.24
Inlet total pressure (p_{o1})	1.2363×10^5 Pa
Inlet total temperature (T_{o1})	312.7 K
Inlet dynamic pressure ($\rho_1 V_1^2/2$)	2791.51 Pa
Inlet velocity (V_1)	64.2 m/s
Atmospheric pressure (p_a)	0.9928×10^5 Pa
Exit Mach number (M_3)	0.22

Table 8.5 Turbine-stage operating conditions for test case of Quest (1990).

compressibility effects would be small.

When the test case of Quest (1990) was applied to program DUCT, problems with solution stability and divergence were encountered. Efforts to obtain a stable converging solution, using the traditional method of damping changes in streamline position, proved fruitless. Thus, additional damping of other quantities, namely changes in stagnation enthalpy, entropy, and tangential velocity, was necessary to obtain a stable solution. Also, since the axial velocity of the flow decreased significantly from inlet to exit, additional measures had to be taken. On the first iteration through the flowfield the flow along streamlines is assumed to be isentropic (since at this point there is insufficient information to evaluate the mixing equations). Thus, total pressure is constant along streamlines on the first iteration. Because the flow area is increasing along the diffuser, the static pressure along streamlines is steadily increasing. Consequently, there comes a point on streamlines of low total pressure where the static and total pressures are equal. This represents a stagnation point in the flow and results in reverse flow downstream of this point. The streamline curvature method cannot accommodate negative meridional velocities and so the computation must be aborted. In the actual flow this reverse flow does not occur since streamlines with low total pressure experience increasing total pressure in the streamwise direction due to spanwise mixing. To get around this problem, the inlet fluid property profiles (p_o , T_o , and swirl angle) were set to radially constant values (no radial gradients) for the first iteration. These constant values were equal to the average value found in each profile. On subsequent iterations the inlet profiles were slowly brought back to the levels given by Quest. The rate at which the inlet profiles were increased was left as a user input and is discussed further in the program documentation of Appendix F.

8.2.2 Diffuser Inlet Conditions

As mentioned previously, the inlet flow properties necessary to start the computation are radial distributions of total pressure, total temperature, and swirl angle. These properties were calculated from the data given by Quest (1990). For obvious reasons these properties were not available right at the hub and casing surfaces of the diffuser. Since boundary layers were not considered in the computations of the present study, the data of Quest was simply linearly

extrapolated to the diffuser surfaces.

Program DUCT uses simple linear interpolation on the inlet data to find the properties at the intersection points of the streamlines and the inlet plane. Figure 8.18 shows the inlet radial distribution of total pressure coefficient for the data points of Quest along with those interpolated at the intersection points of streamlines and the diffuser inlet plane. Here, a greater number of data points or a better interpolation technique would have made for a smoother inlet profile. Figure 8.19 shows the radial distributions of static pressure coefficient at the diffuser inlet. In this case the data points of Quest were used only in calculating the inlet total temperature profile (using Equation 8.2) while the values of static pressure at the inlet streamlines are obtained from the flow computation. In addition to the calculated profile having a slightly greater average static pressure coefficient, the data of Quest show a negative radial pressure gradient while the computed results show a slight positive pressure gradient. This may reveal some of the simplifications made while developing the governing equations for program DUCT. The flow is assumed to be axisymmetric and in radial equilibrium such that the radial pressure field balances all radial inertial forces that the fluid is experiencing. Since the inlet flow has significant swirl, one would expect a positive radial pressure gradient as indicated by the computed results. Streamline modelling (by cubic spline fit) may also contribute to the discrepancies here as this dictates the meridional flow direction at the inlet. The true diffuser inlet flow (turbine exit flow) is obviously a little more complex than that which is assumed in the flow model of program DUCT. This is again revealed in Figure 8.20 which shows the distributions of axial velocity coefficient across the annulus. Again, the data points of Quest (1950) were used in calculating the total temperature profile while the values at streamline-inlet intersection points were computed. Since the inlet data points of swirl angle were used directly in the computations, the values of α at streamline-inlet intersection points simply vary linearly between the data points, as indicated in Figure 8.21. Considering the simplicity of the flow model used, the computed inlet flow conditions shown in Figures 8.18 to 8.21 are considered to be a reasonable representation of the actual inlet flow.

Figure 8.18 Total pressure coefficient at diffuser inlet.

Figure 8.19 Static pressure coefficient at diffuser inlet.

Figure 8.20 Axial velocity coefficient at diffuser inlet.

Figure 8.21 Swirl angle at diffuser inlet.

8.2.3 Determination of Mixing Coefficients

Recall from Chapter 6 that in order to calculate the eddy viscosity throughout the flowfield it is first necessary to determine the distribution of effective mixing coefficient, ϵ . Recall also that the effective mixing coefficient is composed of a turbulent component, ϵ_t , and a convective component, ϵ_c , due to secondary flow. This section gives the values of the mixing coefficients used in the present study and the methods by which they were determined.

In order to calculate the radial distribution of ϵ_c at the diffuser inlet using the secondary flow code (program BLADE), it was necessary to use some approximations. Program BLADE computes the secondary flow created by the passage vortex in a stationary blade passage, while the secondary flows at the turbine exit examined by Quest (1990) are created by both a stator blade row and a subsequent row of rotor blades. Thus, the convective mixing coefficients calculated here should be considered approximate.

The passage geometry used to compute the secondary flows was taken at the mean radius of the stator. The flow turning angle and inlet boundary layer profiles were then adjusted (by trial-and-error) such that the calculated swirl angles approximately matched those measured at the turbine exit. This resulting swirl angle distribution is shown in Figure 8.22. A closer match was possible at the hub since the turbine rotor tip leakage flow significantly affects the swirl angle at the casing. The corresponding spanwise distribution of convective mixing coefficient from the secondary flow computation is shown in Figure 8.23. Here the results are compared with those of Lewis (1993(a)) which were measured in a lightly loaded (LL) multistage turbine. Values found in the blade tip region agree quite well while those calculated in the hub region are about 40% less than those measured by Lewis. This was the distribution of ϵ_c used at the diffuser inlet.

The value of the turbulent mixing coefficient, ϵ_t , was left as an empirical input to program DUCT. As such its value could be easily varied in order to see the effects on the predicted exit flow. Values consistent with those published in the open literature (see Section 2.3.4) were used. Details are given in the next section.

3 of/de 3

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

Figure 8.22 Swirl angle at diffuser inlet measured by Quest (1990) compared with that calculated by best data fit using the secondary flow computation of the present study.

Figure 8.23 Dimensionless convective mixing coefficient measured by Lewis (1993(a)) compared with that calculated for the turbine of Quest (1990)

8.2.4 Comparison of Diffuser Exit Conditions

This section presents the computed diffuser exit conditions and compares them to those measured by Quest (1990). First, however, the results of a parametric study are presented in which the empirical inputs of the turbulent mixing coefficient, ϵ_t , and the decay power-law exponent, d , of Equations (6.13) and (6.15) respectively, were varied to show their effect on spanwise mixing.

In the parametric study the diffuser inlet conditions described in Section 8.2.2 were used in an annular duct of constant cross-sectional area. The inlet area and axial length of the duct were chosen to be the same as those given by Quest. This amounts to an annular diffuser with $\theta=0$. This duct geometry was chosen so that a wider range of ϵ_t could be used without encountering the problems produced by low mixing levels in a decelerating flow (as described in the later part of Section 8.2.1). Also, in this geometry spanwise mixing is the only process which affects the fluid property profiles thereby making the effects of mixing more apparent.

Figures 8.24, 8.25, and 8.26 show the effect of turbulent mixing on the spanwise distributions of total pressure, axial velocity, and swirl angle, respectively, for values of ϵ_t ranging from 0 m²/s (no mixing) to 0.007 m²/s. Here, values of ϵ_t could not be nondimensionalized in the usual manner because there is no associated stage length. Therefore, values of the diffusive mixing coefficient, ϵ_d , measured in two multistage turbines by Lewis (1993(a)) have been calculated for comparative purposes (dimensionless values are given in Table 2.2). In a lightly-loaded (LL) four stage turbine Lewis measured values of ϵ_d ranging from 0.0022 m²/s to 0.0029 m²/s while in a more heavily loaded (HL) three stage turbine the values of ϵ_d ranged from 0.0047 m²/s to 0.0068 m²/s. Thus, the values of ϵ_t used in the parametric study cover the entire range of values measured by Lewis and so represent realistic levels of turbulent mixing for turbines.

As shown in Figures 8.24 through 8.26, turbulent spanwise mixing acts throughout the annulus height and has a significant effect on the spanwise distribution of properties. The effects are more pronounced here due to the great length of the duct (1.10m) as compared with ducts in typical turbomachines. As would be expected, regions with large radial property gradients at inlet are affected the most while regions with small or no radial gradients remain relatively unchanged.

Figure 8.24 Calculated total pressure coefficient at diffuser exit (turbulent mixing only, $\theta=0^\circ$).

Figure 8.25 Calculated axial velocity coefficient at diffuser exit (turbulent mixing only, $\theta=0^\circ$).

Figure 8.26 Calculated swirl angle at diffuser exit (turbulent mixing only, $\theta=0^\circ$).

Figure 8.27 Calculated total pressure coefficient at diffuser exit (convective mixing only, $\theta=0^\circ$, $d=0.75$).

Consequently, the net effect of turbulent mixing is to produce more uniform fluid properties across the annulus. It should be noted that the profiles of static pressure were omitted because they remained relatively unchanged due to the very slight radial gradient seen in the inlet flow (see Figure 8.19).

Unlike turbulent diffusion, convective transport due to radial secondary flow only affects the flow in the endwall regions of the annulus (at least this is the case using the secondary flow computation developed in the present study). This is evident in Figures 8.27, 8.28, and 8.29 which show spanwise profiles of total pressure, axial velocity, and swirl angle, respectively, at the duct exit when only convective mixing is considered. The distribution of convective mixing coefficient, ϵ_c , used at the duct inlet was calculated using program BLADE and was shown in Figure 8.23. The value of the decay power-law exponent (d in Eqn. 6.15) dictates the manner in which the radial secondary velocities decay downstream of the inlet. Values of d ranging from 0.5 to 1.0 were used in the parametric study but variations seen in the exit profiles for different values were only minor. Thus, a value of $d=0.75$ was used to obtain the results shown in Figures 8.27 to 8.29. As seen in these figures, convective mixing has as great an effect on the spanwise profiles as the highest levels of turbulent mixing while properties in the core of the flow remain unaffected. These results seem even more dramatic when one considers that the mixing due to radial secondary velocities decays to zero within the first 40% of the duct length while turbulent mixing acts over the entire length of the duct.

The exit conditions computed using the same diffuser geometry as Quest (1990) were encouraging. However, due to solution instability the mixing effects of turbulence and secondary flow could not be combined into a single test case. Superimposing both effects on the flow simultaneously results in extremely high mixing levels in the endwall regions of the diffuser inlet. Spanwise mixing has proved to be destabilizing to the streamline curvature method and as such no amount of damping was found to stabilize the solution when the two mixing effects were combined (except in cases where extremely low levels of turbulent mixing were used). As a result the effects of each type of mixing are presented separately, as in the parametric study presented previously. With these separate results one can still draw some useful qualitative conclusions about the combined effects of both mixing mechanisms.

Figure 8.28 Calculated axial velocity coefficient at diffuser exit (convective mixing only, $\theta=0^\circ$, $d=0.75$).

Figure 8.29 Calculated swirl angle at diffuser exit (convective mixing only, $\theta=0^\circ$, $d=0.75$).

The results of turbulent mixing are presented first for a range of ϵ_t values. Figure 8.30 shows the static pressure distributions calculated at the diffuser exit along with measurements made by Quest (1990) at diffuser inlet and exit. The calculated static pressures were slightly greater than those measured albeit by a constant amount along the height of the annulus. The reason for this becomes evident from Figure 8.31. This figure shows the calculated total pressure distributions at the diffuser exit along with those measured. The calculated total pressures were consistently greater than those measured indicating that the flow experienced greater losses in the diffuser than those calculated. Since endwall boundary layers were not considered in the computations it is reasonable to suggest that the inclusion of boundary layer losses would make the computed results more closely match those measured. This would in turn lower the static pressure at the diffuser exit. Another consequence of neglecting boundary layers and the annulus blockage which they impose is a calculated axial velocity distribution that is too low. This can be seen in Figure 8.32. Including boundary layer blockage would increase the axial velocity in the diffuser and would also result in lowering the static pressure. Nevertheless, it should be noted that the inclusion of turbulent spanwise mixing improves the agreement between computed and measured diffuser exit conditions.

The largest discrepancy between computed and measured results appears in the diffuser exit swirl angle. This can be seen in Figure 8.33. While the trend in the computed results is reasonable the swirl angles calculated are much greater than those measured, especially at the hub. No doubt this is the result of the specified boundary conditions on the annulus walls. The walls were assumed to be frictionless and while the axial velocity has decreased due to the change in flow area the tangential components of velocity have only been affected by mixing and change in streamline radius. The actual flow experiences shear forces at the annulus walls causing it to spin down and lose angular momentum. Even if spanwise mixing were infinite in the computation the lack of endwall shear stresses would leave the flow with a tangential velocity profile equivalent to that of solid body rotation (no tangential shear stresses in the fluid).

Thus it can be said that the computed results shown in Figures 8.30 to 8.33 are consistent with the flow model used and that spanwise mixing due to turbulence improves the prediction of the diffuser exit conditions. One could also say that the predicted results could be further improved with additional mixing in the endwall regions. This would bring the total pressure and

Figure 8.30 Calculated static pressure coefficient at diffuser exit (turbulent mixing only, $\theta = 7^\circ$).

Figure 8.31 Calculated total pressure coefficient at diffuser exit (turbulent mixing only, $\theta = 7^\circ$).

Figure 8.32 Calculated axial velocity coefficient at diffuser exit (turbulent mixing only, $\theta=7^\circ$).

Figure 8.33 Calculated swirl angle at diffuser exit (turbulent mixing only, $\theta=7^\circ$).

axial velocity distributions (Figs. 8.31 and 8.32, respectively) more in line with the measurements in the casing region. Additional mixing in the hub region would also improve the computed swirl angle. Convective spanwise mixing due to secondary flow could provide this additional mixing in the endwall region.

Figures 8.34 to 8.37 show the computed conditions at the diffuser exit when only convective spanwise mixing is included. The distribution of convective mixing coefficient at diffuser inlet for this case was shown in Figure 8.23 and a power-law exponent of 0.75 was used for the decay of radial velocities along the length of the diffuser. Computed endwall conditions are similar to those seen when high levels of turbulent mixing are used. It would be inappropriate to say that the two effects could simply be superimposed since the rate of spanwise mixing is also affected by the spanwise gradients within the flow. Fluid property gradients in the two cases are of course different. However, one can still argue that the increased mixing levels caused by radial secondary flow in the endwall regions could improve the computed results obtained by considering turbulent mixing only.

Figure 8.34 Calculated static pressure coefficient at diffuser exit (convective mixing only, $\theta = 7^\circ$, $d=0.75$).

Figure 8.35 Calculated total pressure coefficient at diffuser exit (convective mixing only, $\theta = 7^\circ$, $d=0.75$).

Figure 8.36 Calculated axial velocity coefficient at diffuser exit (convective mixing only, $\theta=7^\circ$, $d=0.75$).

Figure 8.37 Calculated swirl angle at diffuser exit (convective mixing only, $\theta=7^\circ$, $d=0.75$).

9.1 Conclusions

A computer program (BLADE) for calculating the flowfield at the exit plane of a stationary rectangular blade passage was developed. The program predicts the development of two counter-rotating passage vortices resulting from the deflection of an inlet shear flow under inviscid and incompressible flow conditions.

The computed secondary flowfield at the passage exit was compared to experimental measurements taken in two linear turbine cascades. Program BLADE was found to predict secondary flow velocities and passage vortex position with reasonable accuracy. Discrepancies in vortex size and position were attributed to the neglecting of fluid viscosity and the code's inability to predict the formation of a leading edge horseshoe vortex.

A grid refinement study on program BLADE revealed that the computed results varied significantly when the grid point density on computational planes was increased. It was found that results obtained using the sparse grids more closely resembled those measured in experiments than results obtained using more dense grids. The variation in results was attributed to the degree of accuracy with which the inlet boundary layer vorticity could be calculated using finite-difference approximations. The problem may have been aggravated by the infinite value of vorticity found at the endwalls due to the use of power-law profiles for the inlet boundary layers.

A computer program (DUCT) for calculating the compressible axisymmetric flowfield in an annular duct was developed. The code, which uses the streamline curvature method, includes a spanwise mixing model which accounts for the mixing effects of turbulent diffusion and convective secondary flow.

Program DUCT was tested using published results from an annular diffuser located downstream of a single stage turbine. The diffuser exit conditions were predicted reasonably well and spanwise mixing was seen to improve the agreement with the measurements. In general, the discrepancies between computed and measured results were consistent with the simplifications made in developing the flow model, namely the neglecting of annulus wall boundary layers and the specification of frictionless walls.

The value of the turbulent mixing coefficient, ϵ_t , used to get as close a match as possible between the computed and measured results was about $0.007 \text{ m}^2/\text{s}$. This was greater than expected for the flow from a lightly-loaded single-stage turbine and is on the order of the turbulent mixing levels observed by Lewis (1993(a)) in a more heavily-loaded three-stage turbine. It was also found that variation of d , the decay power-law exponent for the secondary flow velocities, between 0.5 and 1.0 (representing square-root and linear decays respectively) had little effect on the computed conditions at the diffuser exit.

9.2 Recommendations for Further Study

In order for the code developed in the present study to become a useful tool for the analysis of turbines with damaged blades, blade rows must first be incorporated into the duct flow computation. This is the most obvious recommendation for the further development of program DUCT and should be a relatively straight-forward task given the highly modular design of the code.

One of the major problems with the streamline curvature method is that of solution stability. Various authors have used special stabilizing techniques to achieve convergence in the basic streamline curvature method (see Section 5.2.3). However, the inclusion of spanwise mixing in the streamline curvature program developed for the present study has proved to be destabilizing and the author had to take special measures (see Section 8.2.1) to achieve converged solutions in many cases. The issue of stability was not dealt with in the published works of others who have included spanwise mixing in streamline curvature programs for compressors. It is therefore recommended that some type of stability analysis be performed on program DUCT in order to

identify sources of instability and to develop some guidelines for stabilizing a streamline curvature solution which includes spanwise mixing.

One of the boundary conditions used in program DUCT was that of frictionless walls. In order to improve the computed results in the near-wall regions, more realistic boundary conditions should be used. The recent work of Howard and Gallimore (1992) describes a method which uses a shear force, consistent with the no-slip condition, on the annulus walls. The method is incorporated into a streamline curvature computation which includes a spanwise mixing model very similar to the one used in the present study. Using this method would incorporate the annulus wall boundary layers directly into the throughflow computation and eliminate the need for empirical blockage factors or an independent boundary layer calculation. In addition, the shear force would generate additional losses at the walls and dissipate the angular momentum of a swirling flow, two effects unseen in the computations of the present study.

At the very least, some type of boundary layer computation should be added to the code in order to calculate appropriate annulus blockage factors. This would simply account for boundary layer blockage and would not create additional losses or dissipate the angular momentum of a swirling flow.

In the present study the value of the turbulent mixing coefficient, ϵ_t , was left as an empirical input to program DUCT. A method for predicting this parameter should be developed. The expression used by Gallimore and Cumpsty (1986) (Eqn. 2.7) for compressors could be adapted for turbine use or the expression developed by Lewis (1993(b)) (Eqn. 2.9) could be used directly.

In order to be able to compute more realistic secondary flow velocities in turbines with lower hub-to-tip ratios, program BLADE should be extended to handle annular blade rows. This is a direct extension of the method used in program BLADE and is described in Gregory-Smith and Okan (1991). Also, a method for computing secondary flows created in rotating passages should be considered. Lewis (1993(b)) suggests a method for doing this.

Last but certainly not least, the complete code should be thoroughly tested on the widest

variety of geometries and flow conditions possible to identify its limitations.

References

- Adkins, G.G., Jr., and Smith, L.H., Jr., 1982, "Spanwise Mixing in Axial-Flow Turbomachines," *ASME Journal of Engineering for Power*, Vol. 104, No. 1, pp. 97-110.
- Akima, H., 1974, "A Method of Bivariate Interpolation and Smooth Surface Fitting Based on Local Procedures," *Communications of the ACM*, Vol. 17, No. 1, pp. 18-31.
- Akima, H., 1978, "A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points," *ACM Transactions on Mathematical Software*, Vol. 4, No. 2, pp. 148-159.
- Bardon, M.F., Moffat, W.C., and Randall, J.L., 1975, "Secondary Flow Effects on Gas Exit Angles in Rectilinear Cascades," *ASME Journal of Engineering for Power*, Vol. 97, No. 1, pp. 93-100.
- Bengtsson, B.E., and Nordbeck, S., 1964, "Construction of isarithms and isarithmic maps by computers," *BIT*, Vol. 4, No. 2, pp. 87-105.
- Binder, A., and Romey, R., 1983, "Secondary Flow Effects and Mixing of the Wake Behind a Turbine Stator," *ASME Journal of Engineering for Power*, Vol. 105, No. 1, pp. 40-46.
- Came, P.M., and Marsh, H., 1974, "Secondary Flows in Cascades: Two Simple Derivations for the Components of Vorticity," *Journal Mechanical Engineering Science*, Vol. 16, No. 6, pp. 391-401.
- Clark, I., 1979, *Practical Geostatistics*, Applied Science Publishers, London.
- Cox, H.J.A., 1976, "Through-Flow Calculation Procedures for Application to High Speed Large Turbines," *AGARD Conference Proceedings* 195.
- Davis, J.C., 1986, *Statistics and Data Analysis in Geology*, Second Edition, John Wiley & Sons, Toronto.

Davis, W.R., and Millar, D.A.J., 1975, "A Comparison of the Matrix and Streamline Curvature Methods of Axial Flow Turbomachinery Analysis, From a User's Point of View," *ASME Journal of Engineering for Power*, Vol. 97, pp. 549-560.

Denton, J.D., 1978, "Throughflow Calculations for Transonic Axial Flow Turbines," *ASME Journal of Engineering for Power*, Vol. 100, pp. 212-218.

De Ruyck, J., and Hirsch, C., 1988, "A Radial Mixing Computation Method," ASME Paper No. 88-GT-68.

De Ruyck, J., Hirsch, C., and Segaert, P., 1990, "Secondary Flows and Radial Mixing in Axial Compressors," AGARD Conference Proceedings 469.

Dring, R.P., 1992, "Radial Transport and Momentum Exchange in an Axial Compressor," ASME Paper No. 92-GT-364.

Eckert, E.R.G. and Drake, R.M., Jr., 1972, *Analysis of Heat and Mass Transfer*, McGraw-Hill, Toronto.

Erkilet, M., and Ucer, S., 1990, "A Study on Secondary Flow and Spanwise Mixing in Axial Flow Compressors," AGARD Conference Proceedings 469.

Gallimore, S.J., and Cumpsty, N.A., 1986, "Spanwise Mixing in Multistage Axial Flow Compressors: Part I-Experimental Investigation," *ASME Journal of Turbomachinery*, Vol. 108, No. 1, pp. 2-9.

Gallimore, S.J., 1986, "Spanwise Mixing in Multistage Axial Flow Compressors: Part II-Throughflow Calculations Including Mixing," *ASME Journal of Turbomachinery*, Vol. 108, No. 1, pp. 10-16.

Glynn, D.R., 1982, "Calculation of Secondary Flow in Cascades Including Effects of Bernoulli Surface Distortion," *Int. J. Heat and Fluid Flow*, Vol. 3, pp. 73-80.

Glynn, D.R., and Marsh H., 1980, "Secondary Flows in Annular Cascades," *Int. J. Heat and Fluid Flow*, Vol. 2, No. 1, pp. 29-33.

Gold, C.M., Charters, T.D., and Ramsden, J., 1977, "Automated contour mapping using triangular element data structures and an interpolant over each irregular triangular domain," *Computer Graphics*, Vol. 11, No. 2, pp. 170-175.

Goto, A., 1991, "Three-Dimensional Flow and Mixing in an Axial Flow Compressor with Different Rotor Tip Clearances," ASME Paper No. 91-GT-89.

Gregory-Smith, D.G., and Cleak, J.G.E., 1992, "Secondary Flow Measurements in a Turbine Cascade With High Inlet Turbulence," *ASME Journal of Turbomachinery*, Vol. 114, No. 1, pp. 173-183.

Gregory-Smith, D.G., and Graves, C.P., 1983, "Secondary Flows and Losses in a Turbine Cascade," AGARD Conference Proceedings 351.

Gregory-Smith, D.G., and Okan, M.B., 1991, "A Simple Method for the Calculation of Secondary Flows in Annular Cascades," *Proc. I. Mech. E., Turbomachinery: Latest Developments in a Changing Scene*, European Conference, pp. 227-234.

Heap, B.R., 1972, "Algorithms for the Production of Contour Maps," NPL Rep. NAC-10, Nat. Phys. Lab., Teddington, England

Hearsey, R.M., 1975, "A Revised Computer Program for Axial Compressor Design," ARL-TR-75-0001, Aerospace Research Laboratories, Wright-Patterson Air Force Base, OH.

Hinze, J.O., 1975, *Turbulence*, Second Edition, McGraw-Hill, Toronto.

Hirsch, Ch., and Denton, J.D., 1981, *Through Flow Calculations in Axial Turbomachines*, AGARD Advisory Report 175.

Hoffmann, K.A., 1989, *Computational Fluid Dynamics for Engineers*, Engineering Education System, Austin, TX.

Horlock, J.H., and Lakshminarayana, B., 1973, "Secondary Flow: Theory, Experiment, and Application in Turbomachinery Aerodynamics," *Annual Review of Fluid Mechanics*, Vol. 5, pp. 247-280.

Howard, M.A., and Gallimore, S.J., 1992, "Viscous Throughflow Modelling for Multi-Stage Compressor Design," ASME Paper No. 92-GT-302.

Jabbari, M.Y., and Goldstein, R.J., 1993, "Vortex Structure and Mass Transfer Near the Base of a Cylinder and a Turbine Blade," AGARD Conference Proceedings 527.

James, M.L., Smith, G.M., and Welford, J.C., 1985, *Applied Numerical Methods for Digital Computation*, Third Edition, Harper & Row, New York.

Kaldellis, J., Douvikas, D., Falchetti, F., and Papailiou, K.D., 1990, "A Secondary Flow Calculation Method for Single-Stage Axial Transonic Flow Compressors, Including Shock-Secondary Flow Interaction," *ASME Journal of Turbomachinery*, Vol. 112, No. 3, pp. 652-668.

Kintley, K.R., Beach, T.A., and Adamczyk, J.J., 1990, "Numerical Analysis of Secondary Flow in a Two-Stage Turbine," AIAA Paper No. AIAA-90-2356.

Klein, W.A., 1993, *Contributions to the Study of the Aerodynamics of Damaged Turbine Blades*, M.Eng. Thesis, Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, Canada.

Kotidis, P.A., and Epstein, A.H., 1991, "Unsteady Radial Transport in a Transonic Compressor Stage," *ASME Journal of Turbomachinery*, Vol. 113, No. 2, pp. 207-218.

Kreyszig, E., 1988, *Advanced Engineering Mathematics*, 6th Ed., John Wiley & Sons, Inc., Toronto.

Lakshminarayana, B. and Horlock, J.H., 1972, "Generalized Expressions for Secondary Vorticity Using Intrinsic Co-ordinates," *Journal of Fluid Mechanics*, Vol. 59, pp. 97-115.

Langston, L.S., 1980, "Crossflows in a Turbine Cascade Passage," *ASME Journal of Engineering for Power*, Vol. 102, No. 4, pp. 866-874.

Langston, L.S., Nice, M.L., and Hooper, R.M., 1977, "Three-Dimensional Flow Within a Turbine Cascade Passage," *ASME Journal of Engineering for Power*, Vol. 99, No. 1, pp. 21-28.

Lewis, K.L., 1993(a), "Spanwise Transport in Axial-Flow Turbines: Part 1 - The Multistage Environment," ASME Paper No. 93-GT-289.

Lewis, K.L., 1993(b), "Spanwise Transport in Axial-Flow Turbines: Part 2 - Throughflow Calculations Including Spanwise Transport," ASME Paper No. 93-GT-290.

Leylek, J.H., and Wisler, D.C., 1991, "Mixing in Axial-Flow Compressors: Conclusions Drawn From Three-Dimensional Navier-Stokes Analyses and Experiments," *ASME Journal of Turbomachinery*, Vol. 113, No. 2, pp. 139-160.

Li, S.M., and Chen, M.Z., 1992, "An Investigation of Spanwise Mixing in Multistage Axial Flow Compressors," ASME Paper No. 92-GT-64.

Li, Y.S., and Cumpsty, N.A., 1991(a), "Mixing in Axial Flow Compressors: Part I - Test Facilities and Measurements in a Four-Stage Compressor," *ASME Journal of Turbomachinery*, Vol. 113, No. 2, pp. 161-165.

Li, Y.S., and Cumpsty, N.A., 1991(b), "Mixing in Axial Flow Compressors: Part II - Measurements in a Single-Stage Compressor and a Duct," *ASME Journal of Turbomachinery*, Vol. 113, No. 2, pp. 166-174.

Marchal, P., and Sieverding, C.H., 1977, "Secondary Flows Within Turbomachinery Bladings," AGARD Conference Proceedings 214.

Marsh, H., 1971, "The Uniqueness of Turbomachinery Flow Calculations Using the Streamline Curvature and Matrix Through-Flow Methods," *Journal of Mechanical Engineering Science*, Vol. 13, No. 6, pp. 376-379.

Marsh, H., 1976, "Through-Flow Calculations in Axial Turbomachinery: A Technical Point of View," AGARD Conference Proceedings 195.

McCullagh, M.J., and Ross, C.G., 1980, "Delaunay triangulation of a random data set for isarithmic mapping," *Cartographic Journal*, Vol. 17, No. 2, pp. 93-99.

Ngo, V.H., and Millar, D.J.A., 1973, *The Design and Performance Prediction of Axial Flow Turbines*, Department of Mechanical and Aeronautical Engineering, Carleton University, Report No. ME 73-7.

Novak, R.A., 1967, "Streamline Curvature Computing Procedures for Fluid-Flow Problems," *ASME Journal of Engineering for Power*, Vol. 89, No. 4, pp. 478-487.

Novak, R.A., 1976, "Flow Field and Performance Map Computation for Axial-Flow Compressors and Turbines," AGARD Lecture Series 83.

Novak, R.A., and Hearsey, R.M., 1977, "A Nearly Three-Dimensional Intrablade Computing System for Turbomachinery," *ASME Journal of Fluids Engineering*, Vol. 99, pp. 154-166.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P., 1992, *Numerical Recipes in FORTRAN: the art of scientific computing*, Second Edition, Cambridge University Press.

Quest, J., 1990, "VII.2 Test Case E/DU-2 Annular Diffuser," AGARD Advisory Report 275.

Schlichting, H., 1979, *Boundary Layer Theory*, Seventh Edition, McGraw-Hill, Toronto.

Shaanan, M.R.A., and Daneshyar, H., 1972, "Methods of Calculating Slope and Curvature of Streamlines in Fluid Flow Problems," *Proc. I. Mech. E.*, Vol. 186.

Shepard, D.A., 1968, "A Two-Dimensional Interpolation Function for Irregularly-Spaced Data," *Proceedings of the ACM National Conference*, pp. 517-524.

Sieverding, C.H., 1985, "Recent Progress in the Understanding of Basic Aspects of Secondary Flows in Turbine Blade Passages," *ASME Journal of Engineering for Gas Turbines and Power*, Vol. 107, No. 2, pp. 248-257.

Smallwood, G.J., 1992, *A Technique for Two-Colour Particle Image Velocimetry*, M.A.Sc. Thesis, Department of Mechanical Engineering, University of Ottawa, Ottawa, Canada.

Smith, D.J.L., 1970, "Computer Solutions of Wu's Equations for the Compressible Flow Through Turbomachines," *Int. Symp. Fluid Mech. Turbomach.* (Pennsylvania State University).

Smith, L.H., Jr., 1955, "Secondary Flow in Axial-Flow Turbomachinery," *Transactions of the ASME*, Vol. 77, pp. 1065-1076.

Spaeth, H., 1974, *Spline Algorithms for Curves and Surfaces*, Utilitas Mathematica Publishing, Winnipeg, 1974.

Walsh, J.A., and Gregory-Smith, D.G., 1990, "Inlet Skew and the Growth of Secondary Losses and Vorticity in a Turbine Cascade," *ASME Journal of Turbomachinery*, Vol. 112, No. 3, pp. 633-642.

Walsh, J.L., Ahlberg, J.H., and Nilson, E.N., 1962, "Best Approximation Properties of the Spline Fit," *Journal of Mathematics and Mechanics*, Vol. 11, No. 2, pp. 225-234.

Wennerstrom, A.J., 1991, "A Review of Predictive Efforts for Transport Phenomena in Axial Flow Compressors," *ASME Journal of Turbomachinery*, Vol. 113, No. 2, pp. 175-179.

White, F.M., 1991, *Viscous Fluid Flow*, Second Edition, McGraw-Hill, Toronto.

Whitten, E.H.T., and Koelling, M.E.V., 1975, "Computation of Bicubic-Spline Surfaces for Irregularly-Spaced Data," Tech. Rep. 3, Dept. of Geol. Sci., Northwestern U., Evanston, Ill.

Wilkinson, D.H., 1970, "Stability, Convergence, and Accuracy of Two-Dimensional Streamline Curvature Methods Using Quasi Orthogonals," *I. Mech. E. Thermodynamics & Fluid Dynamics Convention*, Paper 35.

Wisler, D.C., Bauer, R.C., and Okiishi, T.H., 1987, "Secondary Flow, Turbulent Diffusion, and Mixing in Axial-Flow Compressors," *ASME Journal of Turbomachinery*, Vol. 109, No. 4, pp. 455-482.

Yaras, M.I., 1987, *Flow in the Tip Gap of a Rectangular Turbine Cascade*, M.Eng. Thesis, Department of Mechanical and Aeronautical Engineering, Carleton University, Ottawa, Canada.

Zenisek, A., 1970, "Interpolation Polynomials on the Triangle," *Numerische Mathematik*, Vol. 15, pp. 283-296.

Zlamal, M., 1968, "On the Finite Element Method," *Numerische Mathematik*, Vol. 12, pp. 394-409.

Appendices

Appendix A

Contributions to Experimental Research

This appendix describes the author's contributions to the current program of experimental turbomachinery research at Carleton University. As mentioned in the introductory chapter, the present study was initially to have been composed of both computational and experimental investigations but due to time restrictions and unforeseen computational difficulties, the proposed experimental aspects of the study were withdrawn. The experimental investigation was to have been a study of the effects of blade tip region damage in a large scale planar turbine cascade. Details of the cascade which was to be used may be found in Yaras (1987).

It was proposed that tip damage could be modelled by simply opening up the blade tip gaps to a distance equal to ten and fifteen percent of the blade span. Since the cascade was not originally designed to accommodate such large tip clearances, several parts of the cascade had to be redesigned. These parts included several rectangular and wedge-shaped spacing blocks, an extended length top-wall, and extensions for several clamps which are used in securing the end-wall. For future reference, the technical drawings for these parts are included at the end of this appendix.

In preparation for the experiments, a twelve-port Scanivalve was also made operational. This involved building a housing for the device so that pressure lines and electrical connections could be easily attached or removed. As well, a small interface circuit was designed so that the valve could be controlled by a microcomputer. A circuit diagram for this interface is illustrated in Figure A.1. Details on the use of Scanivalves can be found in Yaras (1987).

Figure A.1 Circuit diagram for Scanivalve interface.

Technical Drawings for Cascade Parts

Dwg. No.	Part ¹
B1	Spacing Block A (10%)
B2	Spacing Block A (15%)
B3	Spacing Block B (10%)
B4	Spacing Block B (15%)
W1	Spacing Wedge (10%)
W2	Spacing Wedge (15%)
P1	Front Cover A
P2	Front Cover B
E1	Clamp Extension

¹Percentages in brackets indicate tip clearance with respect to blade span.

Appendix B

Method of Spanwise Grid Clustering

This appendix outlines the method used to generate grid point clustering in the spanwise direction for the grid used in program BLADE. The rectangular grids generated using this method feature smaller grid spacing near the hub and casing surfaces. Grid clustering was desirable for the secondary flow computations of the present study so that the features of thin inlet boundary layers could be captured without the use of large numbers of grid points.

Consider the lower portion of the blade passage divided into n regions, as illustrated in Figure B.1, such that

$$\frac{l}{2} + \frac{\Delta z_1}{2} = \Delta z_1 + \Delta z_2 + \cdots + \Delta z_{n-1} + \Delta z_n . \quad (\text{B.1})$$

where l is the blade passage height or blade span and n is odd. Also let

$$\Delta z_2 = c \Delta z_1$$

$$\Delta z_3 = c \Delta z_2$$

•
•
•

$$\Delta z_{n-1} = c \Delta z_{n-2}$$

$$\Delta z_n = c \Delta z_{n-1}$$

where c is the ratio of grid spacing in the negative z -direction and is less than 1. Substituting the above set of equations into Equation (B.1) gives

Figure B.1 Notation for spanwise grid clustering.

$$\frac{l}{2} + \frac{\Delta z_1}{2} = \Delta z_1 + c\Delta z_1 + c^2\Delta z_1 + \cdots + c^{n-2}\Delta z_1 + c^{n-1}\Delta z_1$$

which can be written in compact form as

$$\frac{l + \Delta z_1}{2\Delta z_1} = \sum_{i=0}^{n-1} c^i. \quad (\text{B.2})$$

The summation in the above equation can be written as the difference of two infinite series

$$\sum_{i=0}^{n-1} c^i = \sum_{i=0}^{\infty} c^i - \sum_{i=n}^{\infty} c^i$$

and is equivalent to

$$\sum_{i=0}^{n-1} c^i = (1 - c^n) \sum_{i=0}^{\infty} c^i.$$

The infinite series on the right-hand-side of the above expression is a well known geometric series and can be rewritten as

$$\sum_{i=0}^{\infty} c^i = \frac{1}{1-c}, \quad |c| < 1$$

and thus

$$\sum_{i=0}^{n-1} c^i = \frac{1 - c^n}{1 - c}.$$

With the above result, Equation (B.2) can be rewritten

$$\frac{l + \Delta z_1}{2\Delta z_1} = \frac{1 - c^n}{1 - c}$$

and rearranged to give

$$\Delta z_1 = \frac{l(1-c)}{1+c-2c^n}, \quad |c| < 1. \quad (\text{B.3})$$

Since all other grid spacings are related to Δz_1 by

$$\Delta z_i = c^{i-1} \Delta z_1, \quad i = 2, \dots, n, \quad (B.4)$$

then in general

$$\Delta z_i = \frac{l(1-c)c^{i-1}}{1+c-2c^n}, \quad i = 1, \dots, n, \quad |c| < 1.$$

The blade span, l , is known from input data while c and n are calculated from user input. The user is prompted for the total number of passage divisions in the spanwise direction (which must be odd) and the grid spacing ratio in the positive z -direction for the lower portion of the passage (denoted c_l). The total number of spanwise passage divisions is equal to $2n-1$ while c_l is related to c by, $c_l = 1/c$, and is greater than unity. Following user input, the values of Δz_1 and Δz_n are calculated using Equations (B.3) and (B.4) respectively and displayed for reference. The z -coordinates of the grid points are then calculated starting from zero at the hub and continuing towards mid-passage using

$$\Delta z_i = c_l^{n-i} \Delta z_n,$$

where $i = n, n-1, \dots, 3, 2$. The grid is then mirrored about mid-span to fill the upper portion of the blade passage. Some examples of grids generated using this technique are illustrated in Figure B.2. Only divisions in the spanwise direction are shown. Here, the total number of spanwise divisions is only 21 (giving $n=11$) and thus c_l must be increased a fair amount in order to achieve significant grid clustering at hub and casing. However, the clustering effect becomes more dramatic as n is increased and typical values of c_l will be less than those used to generate the grids in Figure B.2. In the secondary flow computations performed for the present study it was found that a value of c_l equal to 1.02 with n equal to 51 gave more than sufficient grid density at the end-walls. The values of c_l and n will always depend on the number of grid points the user wants in the boundary layer and what he/she perceives as acceptable computation times.

Figure B.2 Examples of spanwise grid clustering for $n=11$.

Appendix C

Derivation of the Mixing Equations

This appendix presents detailed derivations of the mixing equations introduced in Section 6.1.2. Equations (6.3), (6.4), and (6.5) are derived from the energy and momentum equations for steady laminar viscous flow and are expressed in terms of the (r, θ, x) cylindrical polar coordinate system. In order to utilize the turbulence model of Section 6.1.1, the molecular viscosity and thermal conductivity in the energy and momentum equations are replaced by the eddy viscosity and eddy thermal conductivity, respectively. The flow is assumed to be axisymmetric such that all tangential gradients of fluid properties are zero (ie. $\partial/\partial\theta=0$). The fluid is also assumed to be an ideal gas in which compressibility effects are accounted for; however, there is no allowance for shocks or flow choking. Axial gradients are also considered to be negligible so that there is no mixing in the x -direction.

The energy equation for steady laminar viscous flow can be expressed in general vector form as

$$\rho \frac{Dh}{Dt} = \frac{Dp}{Dt} + \nabla \cdot (k \nabla T) + \Phi ,$$

where Φ is the dissipation function given by

$$\Phi = \mu \left[2 \left(\left(\frac{\partial v_r}{\partial r} \right)^2 + \left(\frac{1}{r} \frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r} \right)^2 + \left(\frac{\partial v_x}{\partial x} \right)^2 \right) + \left(\frac{1}{r} \frac{\partial v_x}{\partial \theta} + \frac{\partial v_\theta}{\partial x} \right)^2 + \left(\frac{\partial v_r}{\partial x} + \frac{\partial v_x}{\partial r} \right)^2 + \left(\frac{1}{r} \frac{\partial v_r}{\partial \theta} + \frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \right)^2 \right] \\ + \lambda \left(\frac{\partial v_r}{\partial r} + \frac{1}{r} \frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r} + \frac{\partial v_x}{\partial x} \right)^2$$

in the (r, θ, x) coordinate system.

The momentum equations for steady laminar viscous flow are the Navier-Stokes equations and can be written

$$\rho \frac{DV}{Dt} = -\nabla p + \nabla(\lambda \nabla \cdot \mathbf{V}) + \mu(\nabla^2 \mathbf{V} + \nabla(\nabla \cdot \mathbf{V})) + 2(\nabla \mu \cdot \nabla) \mathbf{V} + \nabla \mu \times (\nabla \times \mathbf{V}) + \rho \mathbf{F}$$

in general vector form.

Also, the following operations are defined for the (r, θ, x) coordinate system:

$$\nabla f = \frac{\partial f}{\partial r} \mathbf{u} + \frac{1}{r} \frac{\partial f}{\partial \theta} \mathbf{v} + \frac{\partial f}{\partial x} \mathbf{w} ;$$

$$\nabla \cdot \mathbf{g} = \frac{1}{r} \left(\frac{\partial}{\partial r} (r g_r) + \frac{\partial g_\theta}{\partial \theta} + r \frac{\partial g_x}{\partial x} \right) ;$$

$$\nabla \times \mathbf{g} = \frac{1}{r} \begin{vmatrix} \mathbf{u} & r\mathbf{v} & \mathbf{w} \\ \frac{\partial}{\partial r} & \frac{\partial}{\partial \theta} & \frac{\partial}{\partial x} \\ g_r & r g_\theta & g_x \end{vmatrix} ;$$

$$\nabla^2 f = \frac{1}{r} \left[\frac{\partial}{\partial r} \left(r \frac{\partial f}{\partial r} \right) + \frac{1}{r} \frac{\partial^2 f}{\partial \theta^2} + r \frac{\partial^2 f}{\partial x^2} \right] ;$$

and

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + (\mathbf{V} \cdot \nabla f) ;$$

where f is a general scalar function, $\mathbf{g}$ is a general vector function given by

$$\mathbf{g} = g_r \mathbf{u} + g_\theta \mathbf{v} + g_x \mathbf{w} .$$

and $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ are orthogonal unit vectors in the r , θ , and x -directions, respectively. Since, during the course of the derivations, there are repeated references made to Figure 5.2, the figure is reproduced on the following page.

Figure 5.2 Notation for meridional plane.

Now, considering the energy equation and the assumptions stated previously,

$$\nabla \cdot (k \nabla T) = \frac{1}{r} \left[\frac{\partial}{\partial r} \left(r k_r \frac{\partial T}{\partial r} \right) \right]$$

and

$$\Phi = \mu_r \left[2 \left(\left(\frac{\partial v_r}{\partial r} \right)^2 + \left(\frac{v_r}{r} \right)^2 \right) + \left(\frac{\partial v_z}{\partial r} \right)^2 + \left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \right)^2 \right]. \quad (\text{C.1})$$

Also, referring to Figure 5.2, it can be seen that

$$\frac{D}{Dt} = v_m \frac{\partial}{\partial m} \quad (\text{C.2})$$

so that the energy equation becomes

$$\rho \frac{\partial h}{\partial m} = \frac{\partial p}{\partial m} + \frac{1}{r v_m} \left[\frac{\partial}{\partial r} \left(r k_r \frac{\partial T}{\partial r} \right) \right] + \frac{\Phi}{v_m}.$$

Using the thermodynamic relation referred to as Gibb's equation,

$$\frac{\partial S}{\partial m} = \frac{1}{T} \frac{\partial e}{\partial m} + \frac{p}{T} \frac{\partial v}{\partial m},$$

and the definition of specific enthalpy,

$$h = e + p v,$$

(where v is specific volume) the energy equation can be further manipulated to give

$$\frac{\partial S}{\partial m} = \frac{1}{r \rho T v_m} \left[\frac{\partial}{\partial r} \left(r k_r \frac{\partial T}{\partial r} \right) \right] + \frac{\Phi}{\rho T v_m}, \quad (6.3)$$

where Φ is given by Equation (C.1).

Under the assumptions stated previously, the Navier-Stokes equations can be simplified and written in component form for the (r, θ, z) coordinate system as

$$\rho \left(\frac{Dv_r}{Dt} - \frac{v_\theta^2}{r} \right) = \rho F_r - \frac{\partial p}{\partial r} + E_r, \quad (\text{C.3.a})$$

$$\rho \left(\frac{Dv_\theta}{Dt} + \frac{v_r v_\theta}{r} \right) = \rho F_\theta + E_\theta \quad (\text{C.3.b})$$

$$\rho \left(\frac{Dv_x}{Dt} \right) = \rho F_x - \frac{\partial p}{\partial x} + E_x \quad (\text{C.3.c})$$

where E_r , E_θ , and E_x are given by

$$E_r = \frac{2}{r} \left[\frac{\partial}{\partial r} \left(\mu_t r \frac{\partial v_r}{\partial r} \right) - \frac{\mu_t v_r}{r} \right],$$

$$E_\theta = \frac{\partial}{\partial r} \left[\mu_t \left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \right) \right] + \frac{2\mu_t}{r} \left[\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \right],$$

and

$$E_x = \frac{1}{r} \frac{\partial}{\partial r} \left[\mu_t r \frac{\partial v_x}{\partial r} \right],$$

respectively. Referring to Figure 5.2, it can easily be seen that

$$v_r = v_m \sin \zeta,$$

where

$$\sin \zeta = \frac{\partial r}{\partial m}.$$

Using this, in addition to Equation (C.2), the tangential momentum equation (Eqn. (C.3.b)) can be written

$$v_m \frac{\partial v_\theta}{\partial m} + \frac{1}{r} v_m v_\theta \frac{\partial r}{\partial m} = F_\theta + \frac{E_\theta}{\rho} .$$

Multiplying the above equation by v_θ/v_m , neglecting the force term, and simplifying gives

$$\frac{\partial}{\partial m}(r v_\theta) = \frac{r}{v_m} \left(\frac{E_\theta}{\rho} \right) . \quad (6.4)$$

Resolving the radial and axial momentum equations (Equations (C.3.a) and (C.3.c) respectively) in the streamwise direction gives

$$v_m \frac{\partial v_r}{\partial m} \sin \zeta - \frac{v_\theta^2}{r} \sin \zeta = F_r \sin \zeta - \frac{1}{\rho} \frac{\partial p}{\partial r} \sin \zeta + \frac{E_r}{\rho} \sin \zeta \quad (C.4.a)$$

and

$$v_m \frac{\partial v_x}{\partial m} \cos \zeta = F_x \cos \zeta - \frac{1}{\rho} \frac{\partial p}{\partial x} \cos \zeta + \frac{E_x}{\rho} \cos \zeta , \quad (C.4.b)$$

respectively. Again, referring to Figure 5.2 it can be seen that

$$F_r = F_m \frac{\partial r}{\partial m} ,$$

$$F_x = F_m \frac{\partial x}{\partial m} ,$$

and

$$v_m \frac{\partial v_m}{\partial m} = v_r \frac{\partial v_r}{\partial m} + v_x \frac{\partial v_x}{\partial m} .$$

Using these relations and adding Equations (C.4.a) and (C.4.b) gives

$$v_m \frac{\partial v_m}{\partial m} = \frac{v_\theta^2}{r} \frac{\partial r}{\partial m} + F_m - \frac{1}{\rho} \frac{\partial p}{\partial m} + \frac{E_r}{\rho} \sin \zeta + \frac{E_x}{\rho} \cos \zeta . \quad (C.5)$$

Now, consider total specific enthalpy

$$h_o = h + \frac{v_m^2 + v_o^2}{2} ,$$

from which we get

$$\frac{\partial h_o}{\partial m} = \frac{\partial h}{\partial m} + v_m \frac{\partial v_m}{\partial m} + v_o \frac{\partial v_o}{\partial m} .$$

Substituting for $\partial h/\partial m$, the second law of thermodynamics,

$$\frac{\partial h}{\partial m} = T \frac{\partial S}{\partial m} + \frac{1}{\rho} \frac{\partial p}{\partial m} ,$$

results in

$$\frac{\partial h_o}{\partial m} = T \frac{\partial S}{\partial m} + \frac{1}{\rho} \frac{\partial p}{\partial m} + v_m \frac{\partial v_m}{\partial m} + v_o \frac{\partial v_o}{\partial m} .$$

Finally, substituting Equation (C.5) in the above expression for $v_m \frac{\partial v_m}{\partial m}$, neglecting the force term, and simplifying gives

$$\frac{\partial h_o}{\partial m} = T \frac{\partial S}{\partial m} + \frac{v_o}{r} \frac{\partial}{\partial m} (r v_o) + \frac{E_r}{\rho} \sin \zeta + \frac{E_z}{\rho} \cos \zeta . \quad (6.5)$$

Appendix D

Numerical Approximation of Derivatives

This appendix presents the finite-difference approximations of derivatives used in both programs DUCT and BLADE. All finite-difference approximations presented are three-point formulae capable of handling non-uniform point spacing. All formulae are written in terms of $f(x)$ and are presented here because most numerical methods texts deal only with derivative approximations for uniformly spaced points.

Consider the second-order polynomial,

$$f(x) = ax^2 + bx + c \quad , \quad (D.1)$$

passing through the three points $(x_{i-1}, f(x_{i-1}))$, $(x_i, f(x_i))$, and $(x_{i+1}, f(x_{i+1}))$ as illustrated in Figure D.1.a). The value of $f(x)$ at these locations is f_{i-1} , f_i , and f_{i+1} respectively, while the corresponding x values are $-\Delta x$, 0, and $\alpha\Delta x$, respectively. Since these points must satisfy Equation (D.1),

$$f_{i-1} = a(-\Delta x)^2 + b(-\Delta x) + c \quad ,$$

$$f_i = c \quad ,$$

and

$$f_{i+1} = a(\alpha\Delta x)^2 + b(\alpha\Delta x) + c.$$

Solving these equations for the coefficients a , b , and c gives

$$a = \frac{f_{i+1} - (\alpha + 1)f_i + \alpha f_{i-1}}{\alpha(\alpha + 1)(\Delta x)^2} \quad ,$$

Figure D.1 Using second-order polynomial approximations to develop: a) central-; b) forward-; and c) backward-difference expressions.

$$b = \frac{f_{i+1} + (\alpha^2 - 1)f_i - \alpha^2 f_{i-1}}{\alpha(\alpha + 1)(\Delta x)} ,$$

and

$$c = f_i .$$

Differentiating Equation (D.1) gives

$$\frac{\partial f(x)}{\partial x} = 2ax + b ,$$

which at point (x, f_i) reduces to

$$\left. \frac{\partial f(x)}{\partial x} \right|_i = b ,$$

and thus,

$$\left. \frac{\partial f(x)}{\partial x} \right|_i = \frac{f_{i+1} + (\alpha^2 - 1)f_i - \alpha^2 f_{i-1}}{\alpha(\alpha + 1)(\Delta x)} .$$

This is the three-point central-finite-difference approximation of the first derivative of $f(x)$ at the point (x, f_i) .

Differentiating Equation (D.1) a second time gives

$$\frac{\partial^2 f(x)}{\partial x^2} = 2a ,$$

and thus,

$$\left. \frac{\partial^2 f(x)}{\partial x^2} \right|_i = \frac{2[f_{i+1} - (\alpha + 1)f_i + \alpha f_{i-1}]}{\alpha(\alpha + 1)(\Delta x)^2} .$$

This is the three-point central-finite-difference approximation of the second derivative of $f(x)$ at the point (x, f_i) .

In a similar manner, it is possible to derive an expression for the three-point forward-finite-difference approximation of the first derivative of $f(x)$ at point (x, f_i) . With reference to

Figure D.1.b), the forward-difference expression is given by

$$\frac{\partial f(x)}{\partial x} \Big|_i = \frac{-f_{i+2} + (\alpha+1)f_{i+1} - \alpha(\alpha+2)f_i}{\alpha(\alpha+1)(\Delta x)} .$$

A three-point backward-finite-difference expression for the first derivative of $f(x)$ can also be derived. With reference to Figure D.1.c) this is given by

$$\frac{\partial f(x)}{\partial x} \Big|_i = \frac{f_{i-2} - (\alpha+1)f_{i-1} + \alpha(\alpha+2)f_i}{\alpha(\alpha+1)(\Delta x)} .$$

It should be noted that all finite-difference approximations presented in this appendix have truncation errors associated with them which are of the order $(\Delta x)^2$ and are thus referred to as second-order approximations.

Appendix E

Numerical Integration Using Non-Uniform Step-Size

This appendix presents the derivation of an expression which can be used to numerically integrate a function which can only be evaluated at discrete points having unequally spaced abscissa. The method can be thought of as a generalized form of Simpson's one-third rule because it uses a series of quadratic curves to approximate a single curve which could be passed through the discrete points. Here, the expression is derived in terms of points which lie in the x - y plane where values of y can be found for discrete values of x .

Consider a second-order curve of the form, $y = ax^2 + bx + c$, which is passed through the three points (x_i, y_i) , (x_{i+1}, y_{i+1}) , and (x_{i+2}, y_{i+2}) . Integrating the quadratic expression from $x = x_i$ to $x = x_{i+2}$ gives the area under the curve as indicated in Figure E.1. This area, which is composed of two strips of unequal width, is given by

$$Area_j = \int_{x=x_i}^{x=x_{i+2}} (a_j x^2 + b_j x + c_j) dx = \left[\frac{a_j}{3} x^3 + \frac{b_j}{2} x^2 + c_j x \right]_{x=x_i}^{x=x_{i+2}}$$

which, after substitution of the limits, gives

$$Area_j = \frac{a_j}{3} (x_{i+2}^3 - x_i^3) + \frac{b_j}{2} (x_{i+2}^2 - x_i^2) + c_j (x_{i+2} - x_i) . \quad (E.1)$$

The subscript j on the coefficients denotes that they are specific to $Area_j$. Now, since the three points (x_i, y_i) , (x_{i+1}, y_{i+1}) , and (x_{i+2}, y_{i+2}) must satisfy the quadratic,

$$y_i = a_j x_i^2 + b_j x_i + c_j ,$$

$$y_{i+1} = a_j x_{i+1}^2 + b_j x_{i+1} + c_j ,$$

and

Figure E.1 Area in two strips of unequal width under a quadratic.

$$y_{i+2} = a_j x_{i+2}^2 + b_j x_{i+2} + c_j .$$

Solving these equations for a_j , b_j , and c_j gives,

$$a_j = \frac{(y_{i+2} - y_{i+1})(x_{i+1} - x_i) - (y_{i+1} - y_i)(x_{i+2} - x_{i+1})}{(x_{i+2}^2 - x_{i+1}^2)(x_{i+1} - x_i) - (x_{i+1}^2 - x_i^2)(x_{i+2} - x_{i+1})} ,$$

$$b_j = \frac{(y_{i+2} - y_{i+1})(x_{i+1}^2 - x_i^2) - (y_{i+1} - y_i)(x_{i+2}^2 - x_{i+1}^2)}{(x_{i+1}^2 - x_i^2)(x_{i+2} - x_{i+1}) - (x_{i+2}^2 - x_{i+1}^2)(x_{i+1} - x_i)} ,$$

and

$$c_j = y_i - a_j x_i^2 - b_j x_i .$$

Equation (E.1) can now be applied to the area under a curve which passes through $n+1$ points, where n is even. This area consists of n strips of unequal width and is illustrated in Figure E.2. The area of each pair of strips is given by

$$Area_1 = \frac{a_1}{3}(x_3^3 - x_1^3) + \frac{b_1}{2}(x_3^2 - x_1^2) + c_1(x_3 - x_1)$$

$$Area_2 = \frac{a_2}{3}(x_5^3 - x_3^3) + \frac{b_2}{2}(x_5^2 - x_3^2) + c_2(x_5 - x_3)$$

$$Area_3 = \frac{a_3}{3}(x_7^3 - x_5^3) + \frac{b_3}{2}(x_7^2 - x_5^2) + c_3(x_7 - x_5)$$

•
•
•

$$Area_{n/2} = \frac{a_{n/2}}{3}(x_{n+1}^3 - x_{n-1}^3) + \frac{b_{n/2}}{2}(x_{n+1}^2 - x_{n-1}^2) + c_{n/2}(x_{n+1} - x_{n-1}) .$$

Figure E.2 Area under a curve divided into an even number of strips of unequal width.

Summing all the areas gives

$$\int_{x_1}^{x_{n+1}} f(x) dx = \sum_{j=1}^{n/2} Area_j = \sum_{j=1}^{n/2} \frac{a_j}{3} (x_{2j+1}^3 - x_{2j-1}^3) + \frac{b_j}{2} (x_{2j+1}^2 - x_{2j-1}^2) + c_j (x_{2j+1} - x_{2j-1}) .$$

This expression can be used to integrate a function consisting of an odd number of discrete points with unequally spaced abscissa. Here, a_j , b_j , and c_j are expressed in terms of j (since $i = 2j - 1$) and are given by

$$a_j = \frac{(y_{2j+1} - y_{2j})(x_{2j} - x_{2j-1}) - (y_{2j} - y_{2j-1})(x_{2j+1} - x_{2j})}{(x_{2j+1}^2 - x_{2j}^2)(x_{2j} - x_{2j-1}) - (x_{2j}^2 - x_{2j-1}^2)(x_{2j+1} - x_{2j})} ,$$

$$b_j = \frac{(y_{2j+1} - y_{2j})(x_{2j}^2 - x_{2j-1}^2) - (y_{2j} - y_{2j-1})(x_{2j+1}^2 - x_{2j}^2)}{(x_{2j}^2 - x_{2j-1}^2)(x_{2j+1} - x_{2j}) - (x_{2j+1}^2 - x_{2j}^2)(x_{2j} - x_{2j-1})} ,$$

and

$$c_j = y_{2j-1} - a_j x_{2j-1}^2 - b_j x_{2j-1} .$$

It should be noted that if the points (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , ... , (x_{n+1}, y_{n+1}) have equally spaced abscissa, then the familiar Simpson's one-third rule expression

$$\int_{x_1}^{x_{n+1}} f(x) dx = \frac{\Delta x}{3} \left(y_1 + 4 \sum_{j=2,4,6}^n y_j + 2 \sum_{j=3,5,7}^{n-1} y_j + y_{n+1} \right)$$

is obtained, where Δx denotes the difference in abscissa of adjacent points.

Appendix F

Program Documentation

This appendix gives detailed documentation for all programs, subroutines, and user defined functions developed in the present study. It is meant to give users and programmers specific details about such things as input and output file formats, user prompts and messages, the values of hard-coded constants, and any numerical peculiarities which may be associated with a particular piece of code. It also directs the reader to sections which contain pertinent information about a particular piece of code.

Program **BLADE**

This is the main program in the secondary flow computation. This program calls subroutines **BLI**, **GRADNT**, **INTERP**, and **SOLPSI**. The theory and numerical procedures behind the secondary flow computation were presented in Chapter 3 and Chapter 4 respectively. When program **BLADE** is run the user is first prompted with

```
Enter file name containing inlet flow data:
```

This file contains the inlet boundary layer data and takes the form

```
LASTPT          <number of data points>
RHO              <fluid density>
RIN(1),VIN(1),POIN(1)
RIN(2),VIN(2),POIN(2)
      .
      .
      .
RIN(LASTPT-1),VIN(LASTPT-1),POIN(LASTPT-1)
RIN(LASTPT),VIN(LASTPT),POIN(LASTPT)
```

Next the user is prompted with

Enter file name containing blade data:

This file contains the geometric data for the blade passage and should be of the form

WIDTH	<passage width or blade pitch>
XCHORD	<blade axial chord>
INMANG	<inlet flow angle [deg]>
EXMANG	<exit flow angle [deg]>

The final prompt for a file name is for the raw output data. Here the data format is of no concern since this file will be read by the data reduction program REDUCE1. The next prompt presented to the user is

Enter the number of computational regions:

The concept of the computational region was introduced in Section 3.1 and further discussed in Section 3.5. In general, the greater the flow turning produced in the passage, the greater number of computational regions should be used to maintain accuracy. The user is next prompted with

Enter number of cross-passage divisions (EVEN, MAX: 200):

Here the program also displays a suggested value for the number of cross-passage divisions. This suggested value is calculated based on the number of computational regions chosen. The greater number of computational regions chosen the greater number of divisions are suggested. The user should consider this suggested value as an upper limit. In a similar manner the user is next prompted for the number of divisions in the spanwise direction. Here, however, the user is also asked about grid point clustering. The prompt

Enter clustering parameter (≥ 1.0):

is displayed asking the user to enter a value of c_l , the grid spacing ratio for the lower half of the blade passage (as explained in Appendix B). Once this information is entered by the user the values of Δx , Δy , and Δz are displayed. The user is then given the option to change any of the

blade passage divisions or to continue. As the computation continues the number of the computation region and subregion on which calculations are being performed is displayed on the screen so the user is aware of the status of the computations.

During execution of program BLADE a message such as

```
STREAMLINE HAS PASSED BEYOND BLADE SURFACE
INCREASE NUMBER OF COMPUTATIONAL REGIONS
Y-COORDINATE OF POINT: 0.101
MIN Y-COORDINATE : 0.0
MAX Y-COORDINATE : 0.10
```

may appear on the screen. This indicates that a streamline intersection point with an intermediate plane has been calculated to lie beyond the bounds of the blade surfaces. This can occur when the number of divisions in the pitchwise direction is too great and/or the number of computational regions is too low. If this occurs the computation is aborted and the user must run program BLADE again with fewer pitchwise divisions and/or a greater number of computational regions. A similar message occurs if a streamline passes beyond the hub or casing surfaces.

Program DUCT

This is the main program for the annular duct flow computation described in Chapters 5, 6, and 7. This program calls subroutines ADJUST, FINDIF, MIX, NEWVM, and SPLINE. When the user runs program BLADE he/she is presented with the following prompts:

Enter file name containing inlet and annulus data:

This file contains information about the inlet flow conditions and the annulus geometry and takes the form,

CP, RGAS, GAM	$\langle c_p, R, \gamma \rangle$
QOMAX	$\langle \text{total number of QOs} \rangle$
X(1,1), R(1,1)	
X(1,SLMAX), R(1,SLMAX)	
X(2,1), R(2,1)	

```

X(2,SLMAX),R(2,SLMAX)
.
.
.
X(QOMAX-1,1),R(QOMAX-1,1)
X(QOMAX-1,SLMAX),R(QOMAX-1,SLMAX)
X(QOMAX,1),R(QOMAX,1)
X(QOMAX,SLMAX),R(QOMAX,SLMAX)
DATPTS
RADIN(1),TOIN(1),POIN(1),SWIRL(1)
RADIN(2),TOIN(2),POIN(2),SWIRL(2)
.
.
.
RADIN(DATPTS-1),TOIN(DATPTS-1),POIN(DATPTS-1),SWIRL(DATPTS-1)
RADIN(DATPTS),TOIN(DATPTS),POIN(DATPTS),SWIRL(DATPTS)
MASS

```

<endpoint coordinates of QOs>

<number of inlet data points>

< r, T_o, p_o, Ω >

<mass flow rate>

The user is next prompted to enter file names for output. One output file is for viewing the computed results and a second output file is meant to hold raw data. The output file for viewing gives the radial distributions of various properties at the annulus inlet, axial midpoint, and exit. The output file for raw data is processed by program REDUCE2 and will be discussed at a later point.

After being prompted for file names the user then decides whether he/she wants to include mixing in the computation and if so which types. If the user selects turbulent mixing then he/she will be prompted for a value of the turbulent mixing coefficient, ϵ_t . If convective mixing is selected then the user must enter the name of a file which contains convective mixing data produced by the secondary flow computation (program BLADE). This file is created by program REDUCE1 and will be discussed at a later point. The user is also prompted for a scaling factor for the convective data and the decay power-law exponent d . If the user selects a computation which includes spanwise mixing, a damping factor between 0 and 1 must also be selected. A choice of 1 will provide no damping whereas a choice of 0 will not allow any mixing to occur. In the present study damping factors of 0.3-0.4 were used to maintain solution stability. The prompt

Number of iterations to reach full mixing?

allows the user to slowly raise the level of mixing as the computation proceeds. This can aid in

stability in some cases. The user is then prompted to enter the number of streamtubes desired for the calculation as well as the method used to calculate streamline slope and curvature. These methods were discussed in Section 7.2. If the rational cubic spline is chosen to estimate streamline slope and curvature then the user is also prompted for a spline damping factor q . The user is then prompted to enter a damping factor for streamline shifts as well as a first guess of the axial velocity in the annulus. To aid in stability the user has the option to smooth the inlet fluid property profiles. Entering a value of 0 for the percentage of inlet smoothing leaves the inlet profiles untouched while entering a value of 1 removes all gradients from the inlet profiles and gives an average profile value over the annulus inlet. Under normal circumstances a value of 0 should be used for this parameter. Program DUCT does not as yet have any convergence criteria to indicate that the solution has converged. As such, the user simply enters the number of iterations to be performed. As a further aid to solution stability when mixing is being used, the inlet profiles can be averaged (no spanwise gradients) to start the computation and slowly brought to their original shapes. This can be done in steps and the user is prompted for the number of steps required and the number of iterations during each step. The user is also prompted for the number of iterations to be performed after the profiles are up to their correct levels.

Program REDUCE1

This program is used to reduce the raw output data produced by program BLADE. The user has the option to create reduced data for the entire blade passage or either the upper or lower halves of the passage. If a symmetric inlet flow profile was used in the secondary flow computation then the exit flow will also be symmetric about the mid-passage and only half of the data need be considered. The user is next prompted for file names which will belong to files containing various quantities to be graphed. These files reflect quantities which were of interest in the present study. The only file name prompt of general interest is

Enter output filename for u^2 and $(w/u)^2$ vs r :

which will be the file to contain convective mixing data for program DUCT. Its format is not important since it is read directly by program DUCT with no need for user intervention. As a

matter of interest the file contains data which is used to evaluate Equation (6.15) and thereby produce a distribution of ϵ_c for the meridional plane.

Program REDUCE2

This program is used to reduce the raw output data produced by program DUCT. It prompts for file names and produces four data files containing p^* , p_o^* , V_x^* , and α versus h^* . These quantities were of interest in the present study and some examples of graphs containing these quantities can be found in Section 8.2.4.

Subroutine ADJUST

This subroutine is used in the annular duct flow computation to adjust the streamline-QO intersection points such that all streamtubes have the same mass flow rate. After all streamlines are adjusted, values of density and axial velocity are interpolated at the new intersection points. Details of these procedures are given in Section 7.4. The step-size (DR) for the mass flow integration in this subroutine was arbitrarily set to a value equal to one-hundredth of the average streamtube height. This value may be decreased in order to increase the accuracy of the integration (at the expense of longer execution times). This subroutine is called from program DUCT.

Subroutine BLI

This subroutine is used to interpolate properties (e.g. total pressure, axial velocity) from points on a rectangular grid to points on a non-rectangular unstructured grid. The method used for this bivariate interpolation can be found in Section 4.2. This subroutine is called by program BLADE.

Subroutine FINDIF

This subroutine serves the same purpose as subroutine SPLINE in that it is used to calculate streamline slope and curvature. The method used to do this is simply based on using finite-difference equations to approximate the first and second derivatives of r with respect to x . Details can be found in Section 7.2.2. This subroutine calls subroutine GRADNT and is called by program DUCT.

Subroutine GRADNT

This subroutine is simply used to find derivatives for a set of points which lie along a straight line (such as a QO). It coordinates the use of functions DER1FW, DER1V1, and DER1BW using the appropriate finite-difference formula for the appropriate points. This subroutine is called by programs BLADE and DUCT and subroutines FINDIF, MIX, and NEWVM.

Subroutine INOUT

This subroutine is used in the bivariate interpolation scheme described in Section 4.3.4. This subroutine is called by subroutine INTERP and in turn calls subroutine SIDE. It coordinates the use of subroutine SIDE to determine if a point (y,z) lies inside the triangle formed by (y_1,z_1) , (y_2,z_2) , and (y_3,z_3) .

Subroutine INTERP

This subroutine is used to interpolate points from an unstructured grid to a rectangular grid using the method described in Section 4.3.4. This subroutine is called by program BLADE and in turn calls subroutines INOUT and PLANE.

Subroutine MFLOW

This subroutine is used in the annular duct flow computation to calculate the mass flow rate crossing a particular quasi-orthogonal. This subroutine is called by subroutine NEWVM and details of the method used to calculate the mass flow rate are given in Section 7.3.

Subroutine MIX

This subroutine is used to incorporate radial mixing in the annular duct flow computation. The mixing equations (6.3, 6.4, and 6.5) are solved here for the entire flow field using the technique described in Section 7.5. Once the mixing equations are solved, the properties of temperature, pressure, density, and Mach number are updated for the flow field. This subroutine makes numerous calls to subroutine GRADNT and is called by program DUCT.

Subroutine NEWVM

It is in this subroutine where the radial equilibrium equation (Eqn. 5.3) and the continuity equation (Eqn. 5.5) are solved iteratively using the method described in Section 7.3. Convergence problems usually develop here if solution damping is insufficient. A convergence problem usually manifests itself as a divergence of the calculated mass flow rate away from the required mass flow rate. This eventually leads to negative values of meridional and/or axial velocity at some point along a QO. A negative value of velocity may be encountered when the velocity profile along the QO is being adjusted by an amount Δv_x in order to obtain the correct mass flow rate across the QO (see Sect. 7.3). If this should occur the message

```
*** Negative Vx encountered ***
***   in adjusting profile   ***
***      PROGRAM ABORTED      ***
```

is displayed to the user and program execution is stopped. The program must be aborted here

because having the correct mass flow rate across the QO is necessary in order to properly adjust the streamlines. The user must re-run program DUCT using some different values for the various damping factors (see documentation for Program DUCT).

Subroutine PLANE

This subroutine is used to find the interpolated value of $P(y,z)$ which lies in a triangular plane defined by three points of the form (y,z,P) . It is one of the many subroutines used in the bivariate interpolation of points from an unstructured grid to a rectangular grid (see Section 4.3.4). It evaluates the algebraic equations given in Section 4.3.4 and is called by subroutine INTERP.

Subroutine SIDE

This subroutine is used in the bivariate interpolation of points from an unstructured grid to a rectangular grid. It determines on which side of a line through (y_1,z_1) and (y_2,z_2) a point (y,z) lies. This subroutine is called by subroutine INOUT.

Subroutine SOLPSI

This subroutine is used in the secondary flow computation to solve the secondary stream function equation (Eqn. (3.6)). The method used to solve this equation can be found in Section 4.1. The value of variable ERRMAX was set to an arbitrarily small value of 0.01. Too small a value for ERRMAX could prevent the solution from being considered converged when it actually is (due to numerical errors), causing the subroutine to perform the maximum number of 999 iterations (also arbitrarily set). This subroutine is called from program BLADE.

Subroutine SOLVE

This subroutine is used to solve systems of simultaneous algebraic equations using Cholesky's method. This method is known by other names including Crout's method, the method of matrix decomposition, the method of matrix factorization, and LU decomposition. The code in this subroutine has been adapted from that given in James et al. (1985) with only minor modifications.

Subroutine SPLINE

This subroutine is used in the annular duct flow computation to fit a rational cubic spline to the two-dimensional points which define a streamline. Streamline slope and curvature are also calculated within this subroutine. Details of the rational cubic spline are given in Section 7.2.1 while details on calculating slope and curvature are given in Section 7.2.2. The second derivatives at the endpoints of the spline should ideally be set to zero as a boundary condition; however, this creates a "Division by zero" error when the streamline radius of curvature is calculated (the radius of curvature should be infinite). It is for this reason that any second derivatives equalling zero are given an arbitrarily small value of 1.0×10^{-10} . This subroutine is called by program DUCT and calls subroutine SOLVE.

Function DERIV1

This function returns the value of the first derivative of some quantity with respect to another. The method used to approximate the derivative is a three-point central-difference and details are given in Appendix D. This function is used in subroutine GRADNT.

Function DER1FW

This function returns the value of the first derivative of some quantity with respect to

another. The method used to approximate the derivative is a three-point forward-difference and details are given in Appendix D. This function is used in subroutine GRADNT.

Function DERIBW

This function returns the value of the first derivative of some quantity with respect to another. The method used to approximate the derivative is a three-point backward-difference and details are given in Appendix D. This function is used in subroutine GRADNT.

Function LINTRP

This function returns a value found by linearly interpolation of a set of two-dimensional data points. If the point to be interpolated lies outside of the range of the data set then the user is informed and program execution is stopped. This function is used in program DUCT and subroutine MIX.

Appendix G

Program Listings

This appendix contains all of the source code developed for the present study. All code is written in standard FORTRAN 77. Programs are listed first, then subroutines and functions; all being presented in alphabetical order within these three categories (the same order as documented in Appendix F).


```

C Prompt user for file names
WRITE(*,*) ' '
WRITE(*,*) '**** SECONDARY FLOWS CALCULATION ****'
WRITE(*,*) ' '
READ(*,*) INFILE
WRITE(*,*) 'Enter file name containing inlet flow data:'
WRITE(*,*) ' '
READ(*,*) BLFILE
WRITE(*,*) 'Enter file name containing blade data:'
WRITE(*,*) ' '
READ(*,*) BLFILE
WRITE(*,*) 'Enter file name for output data:'
READ(*,*) EXFILE

C Open input files
OPEN(UNIT=7, FILE=INFILE, STATUS='OLD')
OPEN(UNIT=8, FILE=BLFILE, STATUS='OLD')

C Read inlet flow data
READ(7,*) LASTPT
READ(7,*) RHO
DO 5 I=1, LASTPT
  READ(7,*) RIN(I), VIN(I), POIN(I)
5 CONTINUE

C Read blade passage geometry
SPAN=RIN(LASTPT)-RIN(1)
RHUB=RIN(1)
READ(8,*) WIDTH
READ(8,*) XCHORD
READ(8,*) INNANG
READ(8,*) EXWANG

C Convert angles to radians
INWANG=INWANG*PI/180.0
EXWANG=EXWANG*PI/180.0
ALPHA=INNANG

C Input no. of computational regions, subregions, and regular grid
15 WRITE(*,*) ' '
WRITE(*,*) 'Enter number of computational regions:'
READ(5,*) CRS
IF(CRS.LT.1) GO TO 15

C No. of subregions set to 4
SUBRS=4
DX=XCHORD/CRS/SUBRS
DALPHA=(EXWANG-INWANG)/CRS/SUBRS

C Approximate number of passage divisions in the y and z directions
NOY=2*INT(WIDTH/DX/2.)
NOZ=2*INT(SPAN/DX/2.)+1

25 WRITE(*,*) ' '
WRITE(*,*) 'Enter number of cross-passage divisions (EVEN, MAX: 20'
+0)'. '
WRITE(*,*) ' [Suggested value: NOY]'
READ(5,*) NOY
IF(NOY.LT.6.OR.NOY.GT.200) GO TO 25

NOY=2*INT(NOY/2)
INAX=NOY+1
DY=WIDTH/NOY

30 WRITE(*,*) ' '
WRITE(*,*) 'Enter number of spanwise divisions (ODD, MAX: 199):'
WRITE(*,*) ' [Suggested value: NOZ]'
READ(5,*) NOZ
IF(NOZ.LT.5.OR.NOZ.GT.199) GO TO 30
NOZ=2*INT(NOZ/2)+1
JMAX=NOZ+1

31 WRITE(*,*) ' '
WRITE(*,*) 'Enter clustering parameter (>=1):'
READ(5,*) RLOW
IF(RLOW.LT.1.0) GO TO 31

C Calculate parameters for non-uniform spanwise grid spacing
RUP=1./RLOW
IF(RLOW.NE.1.0) THEN
  NHALF=JMAX/2
  DZCEN=SPAN*(1.0-RUP)/(1.0-RUP-2.0*RUP*NHALF)
  DZWALL=RUP*(NHALF-1)*DZCEN
ELSE
  DZCEN=SPAN/NOZ
  DZWALL=DZCEN
ENDIF

WRITE(*,*) ' '
WRITE(*,*) 'DX is: ', DX, ' [m]'
WRITE(*,*) 'DY is: ', DY, ' [m]'
WRITE(*,*) 'Dz at mid-span is: ', DZCEN, ' [m]'
WRITE(*,*) 'Dz at wall is: ', DZWALL, ' [m]'
32 WRITE(*,*) ' '
WRITE(*,*) 'Are these values acceptable? (y or n):'
READ(*,*) (A)
ANS
IF (ANS.EQ. 'n'.OR.ANS.EQ. 'N') THEN
  GO TO 15
ELSEIF (ANS.EQ. 'y'.OR.ANS.EQ. 'Y') THEN
  GO TO 33
ELSE
  GO TO 32
ENDIF

C Set up array of Simpson's coefficients for numerical integration
33 SIMPCO(1)=1./3.
DO 39 K=2, SUBRS, 2
  SIMPCO(K)=4./3.
39 CONTINUE
DO 40 K=3, SUBRS-1, 2
  SIMPCO(K)=2./3.
40 CONTINUE
SIMPCO(SUBRS+1)=1./3.

C Set up rectangular grid at blade row inlet
C Divide passage in pitchwise direction (uniform spacing)
DO 50 I=1, INAX
  DO 45 J=1, JMAX
    Y(I,J)=(J-1)*DY
  END DO
50 CONTINUE

```

```

      RECTY(I,J)=Y(I,J)
45  CONTINUE
50  CONTINUE

C Divide passage in spanwise direction (clustering towards walls)
  Z(1,1)=0.0
  RECT2(1,1)=0.0
  Z(1,JMAX)=SPAN
  RECT2(1,JMAX)=SPAN
  DO 51 J=2,JMAX/2
    Z(1,J)=Z(1,J-1)+2*SMALL*FLOW** (J-2)
    RECT2(1,J)=Z(1,J)
  Z(1,JMAX-J+1)=SPAN-Z(1,J)
  RECT2(1,JMAX-J+1)=Z(1,JMAX-J+1)
51  CONTINUE
  DO 52 I=2,IMAX
    DO 53 J=1,JMAX
      Z(I,J)=Z(1,J)
      RECT2(I,J)=Z(1,J)
53  CONTINUE
52  CONTINUE

C Interpolate new V, Po, and U arrays for calculation of derivatives
C at rectangular grid points
  COSTMP=COS(ALPHA1)
  SINTMP=SIN(ALPHA1)
  DO 54 J=1,JMAX
    POS=RHUB*RECT2(I,J)
    V(J)=XINTRP(POS,RIN,VIN,LA5ET)
    PO(J)=XINTRP(POS,RIN,POIN,LA5ET)
    U(I,J)=V(J)*COSTMP
54  CONTINUE

C Calculate streamwise and total vorticity and speed at regular grid
  points
  DO 55 J=1,JMAX
    PRPTMP(C)=T(C)
    POSTMP(C)=RECT2(I,J)
55  CONTINUE
    CALL GRANT(POSTMP,PRPTMP,DERMP,JMAX)
    DO 56 J=1,JMAX
      COSTMP=COS(ALPHA1)
      SINTMP=SIN(ALPHA1)
56  CONTINUE
      TO=V(J)*JMAX-1
      DZ=RECT2(I,J)-RECT2(I,J-1)
      IF (J.EQ.1) THEN
        SUB=FLOW
      ELSE
        SUB=1
      ENDIF
      END=1
      CALL GRANT(ERL,VALC,DZ,VIN,VALC,END)
      XIN=V(J)*SINTMP
      XIN=V(J)*SINTMP*XIN+CEOS(ALPHA1)*RHO*V(J)*C
      XIN=V(J)*XIN+COSTMP*XIN*SINTMP
      XIN=V(J)*XIN+COSTMP*XIN*SINTMP
60  CONTINUE

C Since the components of vorticity are indexed at the wall (i=0)
  they are set equal to the values next to the wall
  XIN(1,1)=XIN(1,2)

```

```

XINI(1,1)=XINI(1,2)
XISI(1,JMAX)-XISI(1,JMAX-1)
XINI(1,JMAX)-XINI(1,JMAX-1)

C Spread i-D distribution of properties to 2-D (across passage) at
C inlet to blade row
  DO 70 I=2,IMAX
    DO 65 J=1,JMAX
      XISI(1,J)=XISI(1,J)
      XINI(1,J)=XINI(1,J)
      PO(1,J)=PO(1,J)
      U(1,J)=U(1,J)
      DPOD2(1,J)=DPOD2(1,J)
65  CONTINUE
70  CONTINUE

C Calculate integrand of integral equation at blade row inlet
  COSTMP=COS(ALPHA1)
  DO 80 I=2,IMAX-1
    DO 75 J=2,JMAX-1
      INTSUM(I,J)=SIMPCO(1,J)*DPOD2(I,J)*COSTMP*COSTMP
75  CONTINUE
80  CONTINUE

C Set up initial guess of stream function at inlet to blade passage
  DO 88 I=1,IMAX
    DO 87 J=1,JMAX
      PSI(1,J)=0.0
87  CONTINUE
88  CONTINUE

C Solve for stream function at inlet to blade passage
  CALL SOLPSI(PSI,RLOW,ALPHA1,XISI,DY,SPAN,IMAX,JMAX)

C Calculate z derivative of stream function at inlet to blade row
  DO 100 I=2,IMAX
    DO 95 J=1,JMAX
      PRPTMP(I)=PSI(I,J)
      POSTMP(I)=RECT2(I,J)
95  CONTINUE
      CALL GRANT(POSTMP,PRPTMP,DERMP,JMAX)
      DO 96 J=1,JMAX
        PSID2(I,J)=DERMP(J)
        PSID2(I,J)=DERMP(J)
96  CONTINUE
100  CONTINUE

C Calculate y derivative of stream function at inlet to blade row
  DO 105 I=1,IMAX
    DO 100 J=1,JMAX
      PRPTMP(I)=PSI(I,J)
      POSTMP(I)=RECT2(I,J)
105  CONTINUE
      CALL GRANT(POSTMP,PRPTMP,DERMP,JMAX)
      DO 110 I=1,IMAX
        PSIDY(I,J)=DERMP(I)
        PSIDY(I,J)=DERMP(I)
110  CONTINUE
115  CONTINUE

C Start calculation of secondary flows in computational region. C

```

```

DO 310 CR=1,CRS
  PRINT *, 'Computational region:', CR
  FACT1=CR/CRS
  FACT2=1.-FACT1
  DO 35 I=1,IMAX
    DO 36 J=1,JMAX
      DPR2DY(I,J)=FACT1*DPSIDY(I,J)+FACT2*DPSIDY(I,J)
      DPR2DZ(I,J)=FACT1*DPSIDZ(I,J)+FACT2*DPSIDZ(I,J)
    CONTINUE
  CONTINUE
36
35
C Transfer y and z derivatives to other arrays for interpolation from
  rectangular grid to distorted grid (plane 2)
DO 156 I=1,IMAX
  DO 157 J=1,JMAX
    DPR2DY(I,J)=DPR2DY(I,J)
    DPR2DZ(I,J)=DPR2DZ(I,J)
  CONTINUE
157
156
C Interpolate dps1/dz onto distorted grid at plane 2
  CALL BLI(Y,Z,DPS2DZ,IMAX,JMAX)
C Interpolate dps1/dy onto distorted grid at plane 2
  CALL BLI(Y,Z,DPS2DY,IMAX,JMAX)
C Calculate convection of streamlines from plane 1 to plane 2 and
  iterate until converged to some margin of error (only interior)
FRSUM=0.0
DO 165 I=2,IMAX-1
  DO 160 J=2,JMAX-1
    DELX=(0.5*DPSIDZ(I,J)+0.5*DPS2DZ(I,J))*DX/U(I,J)
    DELZ=(0.5*DPSIDY(I,J)+0.5*DPS2DY(I,J))*DZ/U(I,J)
    YNEW=Y(I,J)+DELX
    ERRSUM=ERRSUM+ABS(YNEW-Y(I,J))
    Y(I,J)=YNEW
    ZNEW=Z(I,J)+DELZ
    ERRSUM=ERRSUM+ABS(ZNEW-Z(I,J))
    Z(I,J)=ZNEW
  CONTINUE
C Check if any streamlines have moved beyond passage limits
  IF (YNEW.LT.0.0.OR.ZNEW.GT.SPAN) THEN
    PRINT *, 'STREAMLINE HAS PASSED BEYOND HUB OR CASING'
    PRINT *, 'INCREASE NUMBER OF COMPUTATIONAL REGIONS'
    PRINT *, 'INCREASE NUMBER OF COMPUTATIONAL REGIONS'
    PRINT *, 'Y-COORDINATE OF POINT: ', YNEW
    PRINT *, 'MIN Y-COORDINATE : 0.0'
    PRINT *, 'MAX Y-COORDINATE : ', RECTY(IMAX,1)
  STOP
ELSEIF (ZNEW.LT.0.0.OR.ZNEW.GT.SPAN) THEN
  PRINT *, 'STREAMLINE HAS PASSED BEYOND HUB OR CASING'
  PRINT *, 'INCREASE NUMBER OF COMPUTATIONAL REGIONS'
  PRINT *, 'INCREASE NUMBER OF COMPUTATIONAL REGIONS'
  PRINT *, 'Z-COORDINATE : ', ZNEW
  PRINT *, 'MIN Z-COORDINATE : 0.0'
  PRINT *, 'MAX Z-COORDINATE : ', SPAN
  STOP
ENDIF
160
165
C If the iteration error is greater than a specified amount,
  perform another iteration
  IF (ERRSUM.GT.0.01) GO TO 156
C Interpolate  $\rho$  onto rectangular grid at plane 2
  DO 175 I=2,IMAX-1

```

```

DO 310 CR=1,CRS
  PRINT *, 'Computational region:', CR
  FACT1=CR/CRS
  FACT2=1.-FACT1
  ALPHA=FACT1*INWANG+FACT2*EXWANG
C Estimate streamwise vorticity at exit plane to c.r. (rect grid)
  CONST1=COS(ALPHA1)/COS(ALPHA)
  CONST2=1/(ALPHA-ALPHA1)+0.5*(SIN(2.*ALPHA)-SIN(2.*ALPHA1))/
    /COS(ALPHA1)/COS(ALPHA)
  DO 125 I=2,IMAX-1
    DO 120 J=2,JMAX-1
      XISE(I,J)=CONST1*XISE(I,J)+MINI(I,J)*CONST2
    CONTINUE
  CONTINUE
120
125
C Solve stream function at exit plane to c.r. using current psi
  C values as an initial guess, and estimated value of streamwise
  vorticity
  CALL SOLPSI(PSI, RLOW, ALPHA, XISE, DI, SPAN, IMAX, JMAX)
C Calculate dps1/dz at exit plane to c.r.
  DO 140 I=1,IMAX
    DO 130 J=1,JMAX
      PRPTMP(I)=PSI(I,J)
      POSTMP(J)=RECTZ(I,J)
    CONTINUE
  CALL GRANT(POSTMP, PRPTMP, DERTMP, JMAX)
130
  DO 135 J=1,JMAX
    DPS2DZ(I,J)=DERTMP(J)
  CONTINUE
135
140
C Calculate dps1/dy at exit plane to c.r.
  DO 155 J=1,JMAX
    DO 145 I=1,IMAX
      PRPTMP(I)=PSI(I,J)
      POSTMP(I)=RECTY(I,J)
    CONTINUE
  CALL GRANT(POSTMP, PRPTMP, DERTMP, IMAX)
145
  DO 150 I=1,IMAX
    DPS2DY(I,J)=DERTMP(I)
  CONTINUE
150
155
C Calculate convection of streamlines from one intermediate plane
  C to the next (plane 1 to plane 2)
DO 235 SUBR=1,SUBRS
  PRINT *, 'Exit to subregion:', SUBR
C Store grid point coordinates of plane 1 (no boundary points)
  DO 110 I=2,IMAX-1
    DO 117 J=2,JMAX-1
      Y1(I,J)=Y(I,J)
      Z1(I,J)=Z(I,J)
    CONTINUE
117
110
C Interpolate (linear, univariate) the values of dps1/dy and dps1/dz
  C at plane 2 (rect. grid) using values at inlet and exit to c.r.

```

```

DO 170 J=2, JMAX-1
  PORECT(I,J)=PO(I,J)
  CONTINUE
175 CALL INTERP(Y,2,PORECT,IMAX,JMAX)

C Calculate values of dfo/dz on rectangular grid at plane 2
DO 200 I=1, IMAX
  DO 190 J=1, JMAX
    POSTMP(J)=RECT2(I,J)
    PRPTMP(J)=PORECT(I,J)
    CONTINUE
190 CALL GRADNT(POSTMP,PRPTMP,DERTMP,JMAX)
DO 195 J=1, JMAX
  DP0D2(I,J)=DERTMP(J)
  CONTINUE
200 CONTINUE

C If calculating at exit plane to computational region, store dfo/dz on
C rectangular grid for later use
IF (SUBR.EQ.SUBRSTHEN)
  DO 210 I=1, IMAX
    DO 205 J=1, JMAX
      DP0DZ(I,J)=DP0D2(I,J)
    CONTINUE
210 CONTINUE
  ENDDF

C Interpolate values of dfo/dz back onto distorted grid at plane 2
CALL PLI(Y,2,DP0DZ,IMAX,JMAX)

C Calculate integral of integral equation of to plane 2
ALPHA=FACT*ALPHA*FACT*ALPHA
COSTMP=COS(ALPHA)
DO 220 I=2, JMAX-1
  DO 215 J=2, JMAX-1
    INTSUM(I,J)=INTSUM(I,J)+SIMPCC(SUBR+1)*DP0DZ(I,J)*COSTMP
  CONTINUE
220 CONTINUE
  ENDDF

C Interpolate stream function derivatives to new distorted grid (Y,2)
C to be used in plane 1 of next subroutine
DO 230 I=1, IMAX
  DO 225 J=1, JMAX
    DPS2DY(I,J)=DERTMP(I,J)
    DP0DZ(I,J)=DP0DZ(I,J)
  CONTINUE
230 CONTINUE
  CALL PLI(Y,2,DPS2DY,IMAX,JMAX)
  CALL PLI(Y,2,DP0DZ,IMAX,JMAX)
  ENDDF

C Calculate streamwise vorticity distribution at c.r. exit plane using
C integral equation evaluated by Simpson's Rule (this is the second
C more accurate estimation of streamwise vorticity at c.r. exit)
COSTMP=COS(ALPHA)
CONST=COS(ALPHA)*COSTMP

```

```

DO 250 I=4, IMAX-1
  DO 245 J=2, JMAX-1
    XISE(I,J)=CONST*XISE(I,J)+2*(U(I,J)/RHO/COSTMP*INTSUM
      (I,J)*DALPHA
  CONTINUE
245 CONTINUE
250 CONTINUE

C Interpolate streamwise vorticity onto rectangular grid at exit plane
CALL INTERP(Y,2,XISE,IMAX,JMAX)

C Solve for stream function at exit plane using last PSI values as
C an initial guess
CALL SOLPSI(PSI,ALOW,ALPHA,XISE,DY,SPAN,IMAX,JMAX)

C Calculate y derivative of stream function at exit of c.r.
DO 265 J=1, JMAX
  DO 255 I=1, IMAX
    PRPTMP(I)=PSI(I,J)
    POSTMP(I)=RECT1(I,J)
    CONTINUE
255 CALL GRADNT(POSTMP,PRPTMP,DERTMP,IMAX)
DO 260 I=1, IMAX
  DPS2DY(I,J)=DERTMP(I)
  CONTINUE
265 CONTINUE

C Calculate z derivative of stream function at exit of c.r.
DO 280 I=1, IMAX
  DO 270 J=1, JMAX
    PRPTMP(J)=PSI(I,J)
    POSTMP(J)=RECT2(I,J)
    CONTINUE
270 CALL GRADNT(POSTMP,PRPTMP,DERTMP,JMAX)
DO 275 J=1, JMAX
  DPS2DZ(I,J)=DERTMP(J)
  CONTINUE
280 CONTINUE

C Calculate x (axial) component of velocity at exit plane of
C rectangular grid (since v is constant along streamlines)
CALL INTERP(Y,2,U,IMAX,JMAX)

C If not at the exit of last computational region, then transfer all
C necessary properties to the inlet of next computational region,
C and calculate normal component of inlet vorticity (all properties
C are now on a rectangular grid)
IF (CR.NE.CRSTHEN)
  DO 305 I=1, IMAX
    DO 300 J=1, JMAX
      DPS2DY(I,J)=DPS2DY(I,J)
      DPS2DZ(I,J)=DPS2DZ(I,J)
      DP0DZ(I,J)=DP0DZ(I,J)
      POSTMP(I,J)=POSTMP(I,J)
      XISE(I,J)=XISE(I,J)
      IF (EQ.1.OR.J.EQ.IMAX.OR.J.EQ.1.EQ.JMAX) GO TO 305
      INTSUM(I,J)=INTSUM(I,J)+DP0DZ(I,J)*COSTMP
      XISE(I,J)=XISE(I,J)
      Z(I,J)=RECT2(I,J)
      XISE(I,J)=COSTMP/U(I,J)*(DP0DZ(I,J)/RHO-DPS2DZ
        (I,J)*COSTMP*XISE(I,J))
    CONTINUE
  CONTINUE

```



```
C X(,): x-coordinates of streamline-QQ intersection points
C XCHORD: Blade axial chord (for use in mixing calculation)
C Z(): Spanwise coordinate from secondary flows code
C ZETA(,): Streamline slope angle
C .....
PROGRAM DUCT
CHARACTER*1 ANS
CHARACTER*20 INFILE, EXFILE, MXFILE, RMFILE
INTEGER I,J, COMAX, STMAX, DATPTS, ITER, SLNMX, N, MIXPTS, MIXANS, FITANS,
      MIXKN
REAL X,R,RHO,T,S,H,O,V,T,VM,P1,CP,RCAS,GAM,RHUB,DA,DR,ROUT,TO,PC,
      ZETA,KCARCA,RADIN(20),POIN(20),PHI,VX,LINTRP,
      SWIRL(20),MASS,FV,MFRAC,Q,KAPPA,VXIN,CUMAR,IMANG,DAMP1,
      MIXCOT,Z(20),NUZ(2),XCHORD,D,VSO,VNOLD(50,50),
      ERROR,MSO,PROFRC,TOSUM,POSUM,SHSUM,TONVG,POAVG,SNMVG,
      TOTPR(20),POTNP(20),SWTMP(20),MXSCL,MXDMP
COMMON X(50,50),R(50,50),RHOF(50,50),T(50,50),S(50,50),HO(50,50),
      VT(50,50),VM(50,50),P1,CP,RCAS,GAM,KAPPA(50),VX(50,50),
      ZETA(50,50),RC(50,50),PHI(50,50),PI50(50),VR(50,50),
      MSC(50,50),STEPS,IISTEP,ITAFI,STEP,ILLUM
external myhandler
integer :: see_handler('set','control','myhandler')
C SUB CONSTANTS
E1=4.0*A*XAN(1.0)
C Prompt user for input and output files
WRITE(*,*) ***** ANNUAL FLOW COMPUTATION *****
WRITE(*,*)
READ(*,*) INFILE
WRITE(*,*) 'Enter filename containing inlet and annulus data:'
READ(*,*) EXFILE
WRITE(*,*) RMFILE
READ(*,*) ANS
IF (.NOT.(ANS.EQ.'N'.OR.ANS.EQ.'Y'.OR.ANS.EQ.'Y')) GO TO 7
GO TO 1
C TRANSFER Y OR A.N.S.E.Q.'Y' THEN
WRITE(*,*) '*****'
C Read user file data; mixing .s to be included in the calculation
WRITE(*,*) 'Input gas properties'
WRITE(*,*) 'Include spanwise mixing in the calculation?'
WRITE(*,*) ANS
IF (.NOT.(ANS.EQ.'N'.OR.ANS.EQ.'Y'.OR.ANS.EQ.'Y')) GO TO 7
GO TO 1
C TRANSFER Y OR A.N.S.E.Q.'Y' THEN
WRITE(*,*) '*****'
WRITE(*,*) 'Enter damping factor for charges in mixes'
WRITE(*,*) 'Properties V, S, and HO (0.0-1.0)'
READ(*,*) DAMPF
C .....
WRITE(*,*) '*****'
WRITE(*,*) 'Choose'
WRITE(*,*) '1) Turbulent mixing only'
WRITE(*,*) '2) Convective mixing only'
WRITE(*,*) '3) Both mixing mechanisms'
READ(*,*) MIXANS
IF (MIXANS.EQ.1) GO TO 215
IF (MIXANS.EQ.2) GO TO 205
IF (MIXANS.EQ.3) GO TO 205
GO TO 200
WRITE(*,*) 'Enter filename containing convective mixing data:'
READ(*,*) MXFILE
WRITE(*,*)
WRITE(*,*) 'Enter scaling factor for data (0.0-1.0) :'
READ(*,*) PMSCSL
WRITE(*,*)
WRITE(*,*) 'Enter decay power-law exponent (0.5-1.0):'
READ(*,*) D
C Open file containing convective mixing data (from BLADE) and read
OPEN (UNIT=12, FILE=MXFILE, STATUS='OLD')
READ (12,105) MIXPTS
DO 4 J=1,MIXPTS
      READ(12,100) Z(J),MUZ(J)
      MUZ(J)=MUZ(J)*PMSCSL
      CONTINUE
      READ(12,110) XCHORD
      MIXCOT=0.0
      IF (MIXANS.EQ.2) GO TO 220
      WRITE(*,*) '*****'
      WRITE(*,*) 'Enter turbulent mixing coefficient [m^2/s]:'
      READ(*,*) MIXCOT
      IF (MIXANS.EQ.1) MIXPTS=0
      WRITE(*,*) '*****'
      GO TO 220
      WRITE(*,*) 'Number of iterations to reach full mixing?'
      READ(*,*) MIXN
      ENDF
C Open file containing inlet flow and annulus data
OPEN (UNIT=10, FILE=INFILE, STATUS='OLD')
C Input gas properties
READ(10,*) CP,RCAS,GAM
C Prompt user for number of streamtubes
      WRITE(*,*)
      WRITE(*,*) 'Enter number of streamtubes (EVEN, MIN: 4):'
      READ(10,*) STMAX
      IF (STMAX.LE.4) GO TO 3
      STMAX=2*INT(STMAX/2)
      STMAX=STMAX+1
C Input number of QQs, QQ endpoint coords, and calculate epsilon angles
      REAC(10,*) COMAX
      DO 5 I=1,COMAX
        READ(10,*) X(I),Z(1),P(I,2)
        READ(10,*) KICLSUMAX,R(I,SLMXX)
        KAPPA(I)=ATAN(X(I)-X(1,SLMXX))/(R(I,SLMXX)+P(I,2))
      5 CONTINUE
C Input flow data (To, Fo, swirl, angle) at first QQ (must cover ENTIRE
      ANNULUS STARTING AT HUB AND ENDING AT CASING)
      READ(10,*) DATPTS
      DO 6 J=1,DATPTS
        C .....

```

```

      READ(10,*) RADIN(J),TOIN(J),POIN(J),SWIRL(J)
      SWIRL(J)=SWIRL(J)*PI/180.
      7 CONTINUE

C Input total mass flow rate and calculate mass flow fraction for
C each streamline
      READ(10,*) MASS
      MFRAC=MASS/STMAX

C Prompt user to choose method for calculating slope and curvature
      8 WRITE(*,*) 'Choose method for calculating streamline slope and cur
+vature:'
      WRITE(*,*) '1) Rational cubic spline'
      WRITE(*,*) '2) Finite Difference Approximations'
      READ(10,*) FITANS
      IF (FITANS.EQ.1.OR.FITANS.EQ.2) GO TO 9
      GO TO 8
      9 IF (FITANS.EQ.2) GO TO 10
      WRITE(*,*) 'Enter damping factor Q for spline (0-infinity):'
      WRITE(*,*) 'Q=0: No damping'
      WRITE(*,*) 'Q=infinity: Line segments'
      READ(10,*) Q

10 WRITE(*,*) 'Enter damping factor for streamline shift (0-1):'
      WRITE(*,*) 'Q=0: Total damping (no shift)'
      WRITE(*,*) 'Q=1: No damping (full shift)'
      READ(10,*) DAMP

C Prompt user for some parameters which will help in stabilizing
C the converging solution
      WRITE(*,*) 'Enter percentage of inlet profile smoothing desired'
      WRITE(*,*) '0.0 - no smoothing'
      WRITE(*,*) '1.0 - total smoothing (averaged)'
      READ(10,*) PROFRC

      WRITE(*,*) 'Enter number of steps required to achieve this profile'
      WRITE(*,*) '0 - no steps'
      READ(10,*) STEPS

      WRITE(*,*) 'Enter number of iterations during each step'
      WRITE(*,*) '0 - no iterations'
      READ(10,*) ITSTEP

      WRITE(*,*) 'Enter number of iterations afterwards?'
      WRITE(*,*) '0 - no iterations'
      READ(10,*) ITAFT

C Compute total number of iterations
      N-STEPS=ITSTEP+ITAFT

C Compute profile average values (To, Po, swirl angle)
      TOSUM=0.0
      POSUM=0.0
      SMSUM=0.0
      DO 300 J=1,DATPTS
        TOSUM=TOSUM+TOIN(J)
        POSUM=POSUM+POIN(J)
        SMSUM=SMSUM+SWIRL(J)
      300 CONTINUE
      TOAVG=TOSUM/DATPTS
      POAVG=POSUM/DATPTS
      SMAVG=SMSUM/DATPTS

C Compute smoothed inlet profiles (To, Po, swirl angle)
      DO 310 J=1,DATPTS
        TOIN(J)=TOIN(J)+PROFRC*(TOAVG-TOIN(J))
        POIN(J)=POIN(J)+PROFRC*(POAVG-POIN(J))
        SWIRL(J)=SWIRL(J)+PROFRC*(SMAVG-SWIRL(J))
      310 CONTINUE

C Fill transient profile arrays with initial (average) values
      DO 320 J=1,DATPTS
        TOTMP(J)=TOAVG
        POTMP(J)=POAVG
        SWTMP(J)=SMAVG
      320 CONTINUE

C***** SET UP INITIAL CONDITIONS OF FLOW *****
C Calculate points of intersection of QOs and initial streamlines
C based on streamtubes of equal cross-sectional area
      DO 12 I=1,QOVAR
        RHUB=R(I,1)
        AREA=PI*(R(I,1,SIMAX)**2-RHUB**2)/STMAX
        DX=X(I,SIMAX)-X(I,1)
        DR=R(I,SIMAX)-RHUB
        DO 15 J=1,STMAX-1
          CUMAR=AREA/J
          ROUT=SQRT(CUMAR/PI+RHUB**2)
          R(I,J+1)=ROUT
          X(I,J+1)=X(I,1)+(ROUT-RHUB)/DR*DX
        15 CONTINUE
      12 CONTINUE

C Compute radius of curvature and slope of streamlines
      IF (FITANS.EQ.1) THEN
        DO 20 J=1,SIMAX
          CALL SPLINE(QOVAR,Q,J)
        20 CONTINUE
      ELSE
        DO 21 J=1,SIMAX
          CALL FINDIF(QOVAR,J)
        21 CONTINUE
      ENDIF

C Fill Vx array and calculate radial and meridional velocities.
      DO 22 I=1,QOVAR
        DO 23 J=1,SIMAX

```



```

DO 15 I=1,IMAX
  READ(7,106) Y(I,2),Z(I,2),U(I,2),VP(I,2),W(I,2),XIS(I,2)
15 CONTINUE
16 CONTINUE

C Calculate Simpson's coefficients for cross passage averaging
SIMPCO(1)=1.
DO 20 K=2,IMAX-1,2
  SIMPCO(K)=4.
20 CONTINUE
DO 25 K=3,IMAX-2,2
  SIMPCO(K)=2.
25 CONTINUE
SIMPCO(IMAX)=1.

C Compute required dimensions
DY=Y(2,1)-Y(1,1)
PITCH=Y(IMAX,1)-Y(1,1)

C Compute mass-average axial velocity for entire exit plane
C (Arithmetic average of pitch-wise mass-averages)
SUM1=0.0
DO 26 J=1,JMAX
  SUM2=0.0
  SUMDEN=0.0
  DO 27 I=1,IMAX
    SUM2=SUM2+SIMPCO(I)*U(I,2)*U(I,2)
    SUMDEN=SUMDEN+SIMPCO(I)*U(I,2)
27 CONTINUE
  SUM1=SUM1+SUM2/SUMDEN
26 CONTINUE
UAVG=SUM1/JMAX

IF (ANS.EQ.2) JINIT=INT(JMAX/2)
IF (ANS.EQ.3) JMAX=INT(JMAX/2)

C Calculate cross-passage mass-average flow properties
COSTMP=CGS(EAXGIL)
DO 30 J=JINIT,JY
  SUM1=0.0
  SUM2=0.0
  SUM3=0.0
  SUM4=0.0
  SUMDEN=0.0
  DO 35 I=1,IMAX
    SUM1=SUM1+SIMPCO(I)*U(I,2)*ATAN(VP(I,2)*COSTMP/U(I,1))
    SUM2=SUM2+SIMPCO(I)*U(I,2)*U(I,2)*U(I,2)
    SUM3=SUM3+SIMPCO(I)*W(I,2)*W(I,2)*U(I,2)
    SUM4=SUM4+SIMPCO(I)*U(I,2)*XIS(I,2)
    SUMDEN=SUMDEN+SIMPCO(I)*U(I,2)
35 CONTINUE

C Calculate flow angle deviation (in degrees, -ve = cverturning)
HAVANG=(SUM1/SUNDEN)*180.0/PI
IF (CANGLE.LT.0.0) HAVANG=-1.0*HAVANG
WRITE(8,108) Z(1,J)/SPAN,HAVANG+MANGLE

C Calculate mixing parameters
HAVG2=SUM2/SUNDEN
HAVG3=SUM3/SUNDEN
WRITE(9,110) Z(1,J),HAVG2,HAVG3

C Calculate mixing coefficient at blade passage exit (non-dimension)
MKCOEF=(KCHORD/2.-(KCHORD/2.+(20.*SPAN))*HAVG3/2.-(KCHORD/2.-(KCHORD/2.
WRITE(12,122) Z(1,2)/SPAN,MKCOEF
C Calculate non-dimensional streamwise vorticity (pitch-average)
NOCVOR=SUM4/SUNDEN*PITCH/VINCL
WRITE(13,124) Z(1,2)/SPAN,NOCVOR
30 CONTINUE

C Write output file for non-dimensional secondary vorticity plot
C (for use with EasyPlot TW)
WRITE(11,*) '//cx c 1'
WRITE(11,*) '//cy c 0.5'
WRITE(11,*) '//d z'
WRITE(11,*) '//columns',IMAX
DO 47 J=JMAX,JINIT,-1
  DO 48 I=1,IMAX
    WRITE(11,120) XIS(I,2)*PITCH/VINCL
48 CONTINUE
47 CONTINUE

C Write .dxf file for secondary velocity vector plot (for AutoCAD)
WRITE(10,112) 0,2,0
RADIUS=DY/SPAN/15.
C Draw small circle for base of vector
DO 50 J=JINIT,JMAX
  DO 51 I=1,IMAX
    WRITE(10,114) 0,0,10,Y(I,2)/SPAN,20,Z(I,2)/SPAN,
    30,0,0,40,RADIUS,0
51 CONTINUE
50 CONTINUE
C Draw vector (line) from centre of circles
DO 52 J=JINIT,JMAX
  DO 53 I=1,IMAX
    WRITE(10,116) 0,0,10,Y(I,2)/SPAN,20,Z(I,2)/SPAN,
    30,0,0,11,Y(I,2)/SPAN+VP(I,2)/VINCL/SCALE,21,
    Z(I,2)/SPAN+W(I,2)/VINCL/SCALE,31,0,0,0
53 CONTINUE
52 CONTINUE
C Draw reference vector (inlet centerline velocity) at bottom
WRITE(10,116) 0,0,10,0,0,20,-0.2,30,0,0,11,1,0/SCALE,21,
+ -0.2,31,0,0,0
WRITE(10,118) 0

C Close all files
CLOSE(7,STATUS='KEEP')
CLOSE(8,STATUS='KEEP')
CLOSE(9,STATUS='KEEP')
CLOSE(10,STATUS='KEEP')
CLOSE(11,STATUS='KEEP')
CLOSE(12,STATUS='KEEP')
CLOSE(13,STATUS='KEEP')

C Format statements
100 FORMAT(13,1X,13)
102 FORMAT(F10.5)
104 FORMAT(F10.7,1X,F7.5,1X,F7.5)
106 FORMAT(F10.7,1X,F10.7,1X,F10.5,1X,F10.5,1X,F10.5,1X,F10.2)
108 FORMAT(F7.5,*,*,F10.5)
110 FORMAT(F7.5,*,*,F10.4,*,*,F10.6)
112 FORMAT(13,*,*,SECTION,*,13,*,*,ENTITIES',*,13)
114 FORMAT('CIRCLE',/13,11,13,/,F6.4,/,13,/,F3.1,/,
+ 13,/,F6.5,/,13)
116 FORMAT('LINE',/13,11,13,/,F6.4,/,13,/,F3.1,/,
+ 13,/,F3.1,/,

```



```

C      3:4159...
C      POSTMP(I): Temporary storage of r values
C      PRPTMP(I): " " f(r) values
C      QOMAX: Number of QCs
C      R(I): Radial coordinates of grid points
C      RC(I): Radius of curvature
C      STLINE: Row counter (streamline counter)
C      X(I): Axial coordinates of grid points
C      ZETA(I): Streamline slope angle
C.....
C      SUBROUTINE FINDER(QOMAX,STLINE)
C
C      INTEGER I,QOMAX,STLINE
C      REAL R,X,PRPTMP(50),POSTMP(50),DRDX(50),D2RDX2(50),KAPPA,PHI,
C      * ZETA,RC,PI
C      COMMON X(50,50),R(50,50),RHO(50,50),T(50,50),S(50,50),HO(50,50),
C      * VT(50,50),VN(50,50),PI,CP,RCAS,GAM,KAPPA(50),VX(50,50),
C      * ZETA(50,50),RC(50,50),PHI(50,50),P(50,50),VR(50,50),
C      * MSQ(50,50)
C
C      Calculate dr/dx at every grid point along the streamline
C      DO 10 I=1,QOMAX
C      PRPTMP(I)=R(I,STLINE)
C      POSTMP(I)=X(I,STLINE)
C      10 CONTINUE
C      CALL GRADNT(POSTMP,PRPTMP,DRDX,QOMAX)
C
C      Calculate d(dr/dx)/dx at every grid point along the streamline
C      CALL GRADNT(POSTMP,DRDX,D2RDX2,QOMAX)
C
C      Calculate streamline slope and radius of curvature
C      DO 20 I=1,QOMAX
C      ZETA(I,STLINE)=ATAN(DRDX(I))
C      PHI(I,STLINE)=PI/2.-KAPPA(I)-ZETA(I,STLINE)
C      Safeguard against infinite radius of curvature
C      IF (D2RDX2(I).EQ.0.0) D2RDX2(I)=1.0E-10
C      RC(I,STLINE)=1/(COS(ZETA(I,STLINE))**.3*D2RDX2(I))
C      20 CONTINUE
C
C      RETURN
C      END
C.....
C.....
C.....
C      SUBROUTINE USED TO CALCULATE THE FIRST DERIVATIVES OF SOME PROPERTY
C      WITH RESPECT TO SOME LINEAR COORDINATE FOR A 1-D ARRAY
C.....
C      LAST REVISION: APRIL 16, 1992
C.....
C      VARIABLES:
C.....
C      ALPHA: Factor used in calculation of derivatives
C      DER1BW: Value of derivative returned from function DER1BW
C      DER1FW: Value of derivative returned from function DER1FW
C      DER1V1: Value of derivative returned from function DER1V1
C      DERTV(I): Contains values of derivatives
C      DS: Grid spacing used in calculation of derivatives
C.....

```

```

C      GO TO 3
C      ENDDIF
C      ENDDIF
C      GO TO 1
C
C      Find which two columns of the rectangular grid pc (Y,2) lies between
C
C      3 COL=1
C      2 IF (Y.GT.RECTY(COL,1)) THEN
C      COL=COL+1
C      ELSE
C      IF (COL.EQ.1) THEN
C      UC=2
C      LC=1
C      GO TO 5
C      ELSE
C      UC=COL
C      LC=COL-1
C      GO TO 5
C      ENDDIF
C      ENDDIF
C      GO TO 2
C
C      Perform double linear interpolation
C
C      5 RATIO1=(Y-RECTY(LC,UR))/(RECTY(UC,UR)-RECTY(LC,UR))
C      RATIO2=(2-RECTY(LC,UR))/(RECTY(LC,UR)-RECTY(LC,UR))
C      UPRP=PROPTY(LC,UR)+RATIO1*(PROPTY(UC,UR)-PROPTY(LC,UR))
C      LOWRP=PROPTY(LC,LR)+RATIO1*(PROPTY(UC,LR)-PROPTY(LC,LR))
C      PRPTR(I,J)=LOWRP+RATIO2*(UPRP-LOWRP)
C
C      20 CONTINUE
C      10 CONTINUE
C
C      Transfer interpolated values to output array
C
C      DO 30 J=2,JMAX-1
C      DO 35 I=2,IMAX-1
C      PROPTY(I,J)=PRPTR(I,J)
C      35 CONTINUE
C      30 CONTINUE
C
C      RETURN
C      END
C.....
C.....
C.....
C      SUBROUTINE TO CALCULATE STREAMLINE SLOPE AND RADIUS OF CURVATURE
C      USING SIMPLE FINITE DIFFERENCE EXPRESSIONS TO OBTAIN THE FIRST
C      AND SECOND DERIVATIVES OF r WITH RESPECT TO x
C.....
C      LAST REVISION: JULY 13, 1993
C.....
C      VARIABLES:
C.....
C      DRDX(I): dr/dx
C      D2RDX2(I): d(dr/dx)/dx
C      I: Column counter (QO counter)
C      KAPPA(I): QO lean angle
C      PHI(I): Angle between the x and q directions
C.....

```



```

C Points on hub and endwall
DO 75 JJ=1,JMAX,JMAX-1
  DO 80 II=2,IMAX-1
    DMNSQ1=100.
    PROP1=0.
    DMNSQ2=101.
    PROP2=0.
    DO 85 I=2,IMAX-1
      DO 90 J=2,JMAX-1
        DELX=RECTY(II,JJ)-IRREGY(I,J)
        DELZ=RECTZ(II,JJ)-IRREGZ(I,J)
        DSQ=DELX*DELX+DELZ*DELZ
        IF (DSQ.LE.DMNSQ1) THEN
          DMNSQ1=DMNSQ
          PROP1=PROP1
          DMNSQ2=DSQ
          PROP2=PROP2
        ELSEIF (DSQ.LE.DMNSQ2) THEN
          DMNSQ2=DSQ
          PROP2=PROP2
        ENDIF
      CONTINUE
    CONTINUE
  DMIN1=SQRT(DMNSQ1)
  DMIN2=SQRT(DMNSQ2)
  PROPT(II,JJ)=(DMIN2*PROP1+DMIN1*PROP2)/(DMIN1+DMIN2)
CONTINUE
80 CONTINUE
75 CONTINUE

C Points in passage
C neighbours wall points
  DY=RECTY(2,1)-RECTY(1,1)
  DZ=RECTZ(1,2)-RECTZ(1,1)
  PROPT(1,1)=(DY*PROPT(1,2)+DZ*PROPT(2,1))/(DY+DZ)
  PROPT(1,JMAX)=(DY*PROPT(1,1)+DZ*PROPT(JMAX,1))/(DY+DZ)
  PROPT(JMAX,1)=(DY*PROPT(JMAX,2)+DZ*PROPT(1,JMAX))/(DY+DZ)
  PROPT(JMAX,JMAX)=(DY*PROPT(JMAX,1)+DZ*PROPT(1,JMAX))/(DY+DZ)
*

C Interpolate interior points (using triangular planes)
DO 60 II=2,IMAX-1
  DO 55 JJ=2,JMAX-1
    C Set coordinates of point to be interpolated
    Y=RECTY(II,JJ)
    Z=RECTZ(II,JJ)
    C Find local grid spacing in the z-direction and calculate maximum
    C radius of search area around point Y,Z
    DZ=RECTZ(1,JJ)-RECTZ(1,JJ-1)
    DMAXSQ=9.*(DZ+DZ+DY*DY)
    C Initialize variables
    DMNSQ1=1000.
    PROP1=0.
    Y1=0.

```

```

C Points on blade surfaces
DO 20 II=1,IMAX,IMAX-1
  DO 15 JJ=2,JMAX-1
    DMNSQ1=100.
    PROP1=0.
    DMNSQ2=101.
    PROP2=0.
    DO 10 I=2,IMAX-1
      DO 5 J=2,JMAX-1
        DELX=RECTY(II,JJ)-IRREGY(I,J)
        DELZ=RECTZ(II,JJ)-IRREGZ(I,J)
        DSQ=DELX*DELX+DELZ*DELZ
        IF (DSQ.LE.DMNSQ1) THEN
          DMNSQ1=DMNSQ
          PROP1=PROP1
          DMNSQ2=DSQ
          PROP2=PROP2
        ELSEIF (DSQ.LE.DMNSQ2) THEN
          DMNSQ2=DSQ
          PROP2=PROP2
        ENDIF
      CONTINUE
    CONTINUE
  DMIN1=SQRT(DMNSQ1)
  DMIN2=SQRT(DMNSQ2)
  PROPT(II,JJ)=(DMIN2*PROP1+DMIN1*PROP2)/(DMIN1+DMIN2)
CONTINUE
15 CONTINUE

```

```

C Approximate boundary values by finding two closest interior points to
C each boundary point and using inverse distance weighting
C Points on blade surfaces
DO 20 II=1,IMAX,IMAX-1
  DO 15 JJ=2,JMAX-1
    DMNSQ1=100.
    PROP1=0.
    DMNSQ2=101.
    PROP2=0.
    DO 10 I=2,IMAX-1
      DO 5 J=2,JMAX-1
        DELX=RECTY(II,JJ)-IRREGY(I,J)
        DELZ=RECTZ(II,JJ)-IRREGZ(I,J)
        DSQ=DELX*DELX+DELZ*DELZ
        IF (DSQ.LE.DMNSQ1) THEN
          DMNSQ1=DMNSQ
          PROP1=PROP1
          DMNSQ2=DSQ
          PROP2=PROP2
        ELSEIF (DSQ.LE.DMNSQ2) THEN
          DMNSQ2=DSQ
          PROP2=PROP2
        ENDIF
      CONTINUE
    CONTINUE
  DMIN1=SQRT(DMNSQ1)
  DMIN2=SQRT(DMNSQ2)
  PROPT(II,JJ)=(DMIN2*PROP1+DMIN1*PROP2)/(DMIN1+DMIN2)
CONTINUE
15 CONTINUE

```

```

C Points on blade surfaces
DO 20 II=1,IMAX,IMAX-1
  DO 15 JJ=2,JMAX-1
    DMNSQ1=100.
    PROP1=0.
    DMNSQ2=101.
    PROP2=0.
    DO 10 I=2,IMAX-1
      DO 5 J=2,JMAX-1
        DELX=RECTY(II,JJ)-IRREGY(I,J)
        DELZ=RECTZ(II,JJ)-IRREGZ(I,J)
        DSQ=DELX*DELX+DELZ*DELZ
        IF (DSQ.LE.DMNSQ1) THEN
          DMNSQ1=DMNSQ
          PROP1=PROP1
          DMNSQ2=DSQ
          PROP2=PROP2
        ELSEIF (DSQ.LE.DMNSQ2) THEN
          DMNSQ2=DSQ
          PROP2=PROP2
        ENDIF
      CONTINUE
    CONTINUE
  DMIN1=SQRT(DMNSQ1)
  DMIN2=SQRT(DMNSQ2)
  PROPT(II,JJ)=(DMIN2*PROP1+DMIN1*PROP2)/(DMIN1+DMIN2)
CONTINUE
15 CONTINUE

```

```

65      CONTINUE
70      CONTINUE

C Pass planes through 2 different triangles which contain point
C at which property is to be interpolated. The two results are
C then averaged
      CALL INOUT(Y1,Z1,Y2,Z2,Y3,Z3,Y,Z,IN)
      IF (IN.EQ.1) THEN
        CALL PLANE(Y1,Z1,PROP1,Y2,Z2,PROP2,Y3,Z3,PROP3,Y,Z,
          PRIN1)
      +
      ELSE
        CALL PLANE(Y1,Z1,PROP1,Y3,Z3,PROP3,Y4,Z4,PROP4,Y,Z,
          PRIN1)
      +
      ENDIF
      CALL INOUT(Y1,Z1,Y2,Z2,Y4,Z4,Y,Z,IN)
      IF (IN.EQ.1) THEN
        CALL PLANE(Y1,Z1,PROP1,Y2,Z2,PROP2,Y4,Z4,PROP4,Y,Z,
          PRIN2)
      +
      ELSE
        CALL PLANE(Y2,Z2,PROP2,Y3,Z3,PROP3,Y4,Z4,PROP4,Y,Z,
          PRIN2)
      +
      ENDIF
      PRTEMP (11,JJ)=(PRIN1+PRIN2)/2.
55      CONTINUE
60      CONTINUE

C Transfer interpolated interior points to output matrix
      DO 30 I=2,IMAX-1
        DO 25 J=2,JMAX-1
          PROPT(I,J)=PRTEMP(I,J)
25      CONTINUE
30      CONTINUE
      RETURN
      END

```

SUBROUTINE FOR CALCULATING MASS FLOW RATE ACROSS QQ: INTEGRATION
OF THE CONTINUITY EQUATION USING A GENERALIZED FORM OF "SIMPSON'S
ONE-THIRD RULE" FOR NON-UNIFORM STEP-SIZE

LAST REVISION: JULY 8, 1993

VARIABLES:

A: local quadratic coefficient
B: "
C: "
DEN: Denominator of expressions for A and -B
EQN: (Density * X * Z) at point 2K
F2M1: "
F2M2: "
F2M3: "
F2M4: "
F2M5: "
F2M6: "
F2M7: "
F2M8: "
F2M9: "
F2M10: "
F2M11: "
F2M12: "
F2M13: "
F2M14: "
F2M15: "
F2M16: "
F2M17: "
F2M18: "
F2M19: "
F2M20: "
F2M21: "
F2M22: "
F2M23: "
F2M24: "
F2M25: "
F2M26: "
F2M27: "
F2M28: "
F2M29: "
F2M30: "
F2M31: "
F2M32: "
F2M33: "
F2M34: "
F2M35: "
F2M36: "
F2M37: "
F2M38: "
F2M39: "
F2M40: "
F2M41: "
F2M42: "
F2M43: "
F2M44: "
F2M45: "
F2M46: "
F2M47: "
F2M48: "
F2M49: "
F2M50: "
F2M51: "
F2M52: "
F2M53: "
F2M54: "
F2M55: "
F2M56: "
F2M57: "
F2M58: "
F2M59: "
F2M60: "
F2M61: "
F2M62: "
F2M63: "
F2M64: "
F2M65: "
F2M66: "
F2M67: "
F2M68: "
F2M69: "
F2M70: "
F2M71: "
F2M72: "
F2M73: "
F2M74: "
F2M75: "
F2M76: "
F2M77: "
F2M78: "
F2M79: "
F2M80: "
F2M81: "
F2M82: "
F2M83: "
F2M84: "
F2M85: "
F2M86: "
F2M87: "
F2M88: "
F2M89: "
F2M90: "
F2M91: "
F2M92: "
F2M93: "
F2M94: "
F2M95: "
F2M96: "
F2M97: "
F2M98: "
F2M99: "
F2M100: "
F2M101: "
F2M102: "
F2M103: "
F2M104: "
F2M105: "
F2M106: "
F2M107: "
F2M108: "
F2M109: "
F2M110: "
F2M111: "
F2M112: "
F2M113: "
F2M114: "
F2M115: "
F2M116: "
F2M117: "
F2M118: "
F2M119: "
F2M120: "
F2M121: "
F2M122: "
F2M123: "
F2M124: "
F2M125: "
F2M126: "
F2M127: "
F2M128: "
F2M129: "
F2M130: "
F2M131: "
F2M132: "
F2M133: "
F2M134: "
F2M135: "
F2M136: "
F2M137: "
F2M138: "
F2M139: "
F2M140: "
F2M141: "
F2M142: "
F2M143: "
F2M144: "
F2M145: "
F2M146: "
F2M147: "
F2M148: "
F2M149: "
F2M150: "
F2M151: "
F2M152: "
F2M153: "
F2M154: "
F2M155: "
F2M156: "
F2M157: "
F2M158: "
F2M159: "
F2M160: "
F2M161: "
F2M162: "
F2M163: "
F2M164: "
F2M165: "
F2M166: "
F2M167: "
F2M168: "
F2M169: "
F2M170: "
F2M171: "
F2M172: "
F2M173: "
F2M174: "
F2M175: "
F2M176: "
F2M177: "
F2M178: "
F2M179: "
F2M180: "
F2M181: "
F2M182: "
F2M183: "
F2M184: "
F2M185: "
F2M186: "
F2M187: "
F2M188: "
F2M189: "
F2M190: "
F2M191: "
F2M192: "
F2M193: "
F2M194: "
F2M195: "
F2M196: "
F2M197: "
F2M198: "
F2M199: "
F2M200: "
F2M201: "
F2M202: "
F2M203: "
F2M204: "
F2M205: "
F2M206: "
F2M207: "
F2M208: "
F2M209: "
F2M210: "
F2M211: "
F2M212: "
F2M213: "
F2M214: "
F2M215: "
F2M216: "
F2M217: "
F2M218: "
F2M219: "
F2M220: "
F2M221: "
F2M222: "
F2M223: "
F2M224: "
F2M225: "
F2M226: "
F2M227: "
F2M228: "
F2M229: "
F2M230: "
F2M231: "
F2M232: "
F2M233: "
F2M234: "
F2M235: "
F2M236: "
F2M237: "
F2M238: "
F2M239: "
F2M240: "
F2M241: "
F2M242: "
F2M243: "
F2M244: "
F2M245: "
F2M246: "
F2M247: "
F2M248: "
F2M249: "
F2M250: "
F2M251: "
F2M252: "
F2M253: "
F2M254: "
F2M255: "
F2M256: "
F2M257: "
F2M258: "
F2M259: "
F2M260: "
F2M261: "
F2M262: "
F2M263: "
F2M264: "
F2M265: "
F2M266: "
F2M267: "
F2M268: "
F2M269: "
F2M270: "
F2M271: "
F2M272: "
F2M273: "
F2M274: "
F2M275: "
F2M276: "
F2M277: "
F2M278: "
F2M279: "
F2M280: "
F2M281: "
F2M282: "
F2M283: "
F2M284: "
F2M285: "
F2M286: "
F2M287: "
F2M288: "
F2M289: "
F2M290: "
F2M291: "
F2M292: "
F2M293: "
F2M294: "
F2M295: "
F2M296: "
F2M297: "
F2M298: "
F2M299: "
F2M300: "
F2M301: "
F2M302: "
F2M303: "
F2M304: "
F2M305: "
F2M306: "
F2M307: "
F2M308: "
F2M309: "
F2M310: "
F2M311: "
F2M312: "
F2M313: "
F2M314: "
F2M315: "
F2M316: "
F2M317: "
F2M318: "
F2M319: "
F2M320: "
F2M321: "
F2M322: "
F2M323: "
F2M324: "
F2M325: "
F2M326: "
F2M327: "
F2M328: "
F2M329: "
F2M330: "
F2M331: "
F2M332: "
F2M333: "
F2M334: "
F2M335: "
F2M336: "
F2M337: "
F2M338: "
F2M339: "
F2M340: "
F2M341: "
F2M342: "
F2M343: "
F2M344: "
F2M345: "
F2M346: "
F2M347: "
F2M348: "
F2M349: "
F2M350: "
F2M351: "
F2M352: "
F2M353: "
F2M354: "
F2M355: "
F2M356: "
F2M357: "
F2M358: "
F2M359: "
F2M360: "
F2M361: "
F2M362: "
F2M363: "
F2M364: "
F2M365: "
F2M366: "
F2M367: "
F2M368: "
F2M369: "
F2M370: "
F2M371: "
F2M372: "
F2M373: "
F2M374: "
F2M375: "
F2M376: "
F2M377: "
F2M378: "
F2M379: "
F2M380: "
F2M381: "
F2M382: "
F2M383: "
F2M384: "
F2M385: "
F2M386: "
F2M387: "
F2M388: "
F2M389: "
F2M390: "
F2M391: "
F2M392: "
F2M393: "
F2M394: "
F2M395: "
F2M396: "
F2M397: "
F2M398: "
F2M399: "
F2M400: "
F2M401: "
F2M402: "
F2M403: "
F2M404: "
F2M405: "
F2M406: "
F2M407: "
F2M408: "
F2M409: "
F2M410: "
F2M411: "
F2M412: "
F2M413: "
F2M414: "
F2M415: "
F2M416: "
F2M417: "
F2M418: "
F2M419: "
F2M420: "
F2M421: "
F2M422: "
F2M423: "
F2M424: "
F2M425: "
F2M426: "
F2M427: "
F2M428: "
F2M429: "
F2M430: "
F2M431: "
F2M432: "
F2M433: "
F2M434: "
F2M435: "
F2M436: "
F2M437: "
F2M438: "
F2M439: "
F2M440: "
F2M441: "
F2M442: "
F2M443: "
F2M444: "
F2M445: "
F2M446: "
F2M447: "
F2M448: "
F2M449: "
F2M450: "
F2M451: "
F2M452: "
F2M453: "
F2M454: "
F2M455: "
F2M456: "
F2M457: "
F2M458: "
F2M459: "
F2M460: "
F2M461: "
F2M462: "
F2M463: "
F2M464: "
F2M465: "
F2M466: "
F2M467: "
F2M468: "
F2M469: "
F2M470: "
F2M471: "
F2M472: "
F2M473: "
F2M474: "
F2M475: "
F2M476: "
F2M477: "
F2M478: "
F2M479: "
F2M480: "
F2M481: "
F2M482: "
F2M483: "
F2M484: "
F2M485: "
F2M486: "
F2M487: "
F2M488: "
F2M489: "
F2M490: "
F2M491: "
F2M492: "
F2M493: "
F2M494: "
F2M495: "
F2M496: "
F2M497: "
F2M498: "
F2M499: "
F2M500: "
F2M501: "
F2M502: "
F2M503: "
F2M504: "
F2M505: "
F2M506: "
F2M507: "
F2M508: "
F2M509: "
F2M510: "
F2M511: "
F2M512: "
F2M513: "
F2M514: "
F2M515: "
F2M516: "
F2M517: "
F2M518: "
F2M519: "
F2M520: "
F2M521: "
F2M522: "
F2M523: "
F2M524: "
F2M525: "
F2M526: "
F2M527: "
F2M528: "
F2M529: "
F2M530: "
F2M531: "
F2M532: "
F2M533: "
F2M534: "
F2M535: "
F2M536: "
F2M537: "
F2M538: "
F2M539: "
F2M540: "
F2M541: "
F2M542: "
F2M543: "
F2M544: "
F2M545: "
F2M546: "
F2M547: "
F2M548: "
F2M549: "
F2M550: "
F2M551: "
F2M552: "
F2M553: "
F2M554: "
F2M555: "
F2M556: "
F2M557: "
F2M558: "
F2M559: "
F2M560: "
F2M561: "
F2M562: "
F2M563: "
F2M564: "
F2M565: "
F2M566: "
F2M567: "
F2M568: "
F2M569: "
F2M570: "
F2M571: "
F2M572: "
F2M573: "
F2M574: "
F2M575: "
F2M576: "
F2M577: "
F2M578: "
F2M579: "
F2M580: "
F2M581: "
F2M582: "
F2M583: "
F2M584: "
F2M585: "
F2M586: "
F2M587: "
F2M588: "
F2M589: "
F2M590: "
F2M591: "
F2M592: "
F2M593: "
F2M594: "
F2M595: "
F2M596: "
F2M597: "
F2M598: "
F2M599: "
F2M600: "
F2M601: "
F2M602: "
F2M603: "
F2M604: "
F2M605: "
F2M606: "
F2M607: "
F2M608: "
F2M609: "
F2M610: "
F2M611: "
F2M612: "
F2M613: "
F2M614: "
F2M615: "
F2M616: "
F2M617: "
F2M618: "
F2M619: "
F2M620: "
F2M621: "
F2M622: "
F2M623: "
F2M624: "
F2M625: "
F2M626: "
F2M627: "
F2M628: "
F2M629: "
F2M630: "
F2M631: "
F2M632: "
F2M633: "
F2M634: "
F2M635: "
F2M636: "
F2M637: "
F2M638: "
F2M639: "
F2M640: "
F2M641: "
F2M642: "
F2M643: "
F2M644: "
F2M645: "
F2M646: "
F2M647: "
F2M648: "
F2M649: "
F2M650: "
F2M651: "
F2M652: "
F2M653: "
F2M654: "
F2M655: "
F2M656: "
F2M657: "
F2M658: "
F2M659: "
F2M660: "
F2M661: "
F2M662: "
F2M663: "
F2M664: "
F2M665: "
F2M666: "
F2M667: "
F2M668: "
F2M669: "
F2M670: "
F2M671: "
F2M672: "
F2M673: "
F2M674: "
F2M675: "
F2M676: "
F2M677: "
F2M678: "
F2M679: "
F2M680: "
F2M681: "
F2M682: "
F2M683: "
F2M684: "
F2M685: "
F2M686: "
F2M687: "
F2M688: "
F2M689: "
F2M690: "
F2M691: "
F2M692: "
F2M693: "
F2M694: "
F2M695: "
F2M696: "
F2M697: "
F2M698: "
F2M699: "
F2M700: "
F2M701: "
F2M702: "
F2M703: "
F2M704: "
F2M705: "
F2M706: "
F2M707: "
F2M708: "
F2M709: "
F2M710: "
F2M711: "
F2M712: "
F2M713: "
F2M714: "
F2M715: "
F2M716: "
F2M717: "
F2M718: "
F2M719: "
F2M720: "
F2M721: "
F2M722: "
F2M723: "
F2M724: "
F2M725: "
F2M726: "
F2M727: "
F2M728: "
F2M729: "
F2M730: "
F2M731: "
F2M732: "
F2M733: "
F2M734: "
F2M735: "
F2M736: "
F2M737: "
F2M738: "
F2M739: "
F2M740: "
F2M741: "
F2M742: "
F2M743: "
F2M744: "
F2M745: "
F2M746: "
F2M747: "
F2M748: "
F2M749: "
F2M750: "
F2M751: "
F2M752: "
F2M753: "
F2M754: "
F2M755: "
F2M756: "
F2M757: "
F2M758: "
F2M759: "
F2M760: "
F2M761: "
F2M762: "
F2M763: "
F2M764: "
F2M765: "
F2M766: "
F2M767: "
F2M768: "
F2M769: "
F2M770: "
F2M771: "
F2M772: "
F2M773: "
F2M774: "
F2M775: "
F2M776: "
F2M777: "
F2M778: "
F2M779: "
F2M780: "
F2M781: "
F2M782: "
F2M783: "
F2M784: "
F2M785: "
F2M786: "
F2M787: "
F2M788: "
F2M789: "
F2M790: "
F2M791: "
F2M792: "
F2M793: "
F2M794: "
F2M795: "
F2M796: "
F2M797: "
F2M798: "
F2M799: "
F2M800: "
F2M801: "
F2M802: "
F2M803: "
F2M804: "
F2M805: "
F2M806: "
F2M807: "
F2M808: "
F2M809: "
F2M810: "
F2M811: "
F2M812: "
F2M813: "
F2M814: "
F2M815: "
F2M816: "
F2M817: "
F2M818: "
F2M819: "
F2M820: "
F2M821: "
F2M822: "
F2M823: "
F2M824: "
F2M825: "
F2M826: "
F2M827: "
F2M828: "
F2M829: "
F2M830: "
F2M831: "
F2M832: "
F2M833: "
F2M834: "
F2M835: "
F2M836: "
F2M837: "
F2M838: "
F2M839: "
F2M840: "
F2M841: "
F2M842: "
F2M843: "
F2M844: "
F2M845: "
F2M846: "
F2M847: "
F2M848: "
F2M849: "
F2M850: "
F2M851: "
F2M852: "
F2M853: "
F2M854: "
F2M855: "
F2M856: "
F2M857: "
F2M858: "
F2M859: "
F2M860: "
F2M861: "
F2M862: "
F2M863: "
F2M864: "
F2M865: "
F2M866: "
F2M867: "
F2M868: "
F2M869: "
F2M870: "
F2M871: "
F2M872: "
F2M873: "
F2M874: "
F2M875: "
F2M876: "
F2M877: "
F2M878: "
F2M879: "
F2M880: "
F2M881: "
F2M882: "
F2M883: "
F2M884: "
F2M885: "
F2M886: "
F2M887: "
F2M888: "
F2M889: "
F2M890: "
F2M891: "
F2M892: "
F2M893: "
F2M894: "
F2M895: "
F2M896: "
F2M897: "
F2M898: "
F2M899: "
F2M900: "
F2M901: "
F2M902: "
F2M903: "
F2M904: "
F2M905: "
F2M906: "
F2M907: "
F2M908: "
F2M909: "
F2M910: "
F2M911: "
F2M912: "
F2M913: "
F2M914: "
F2M915: "
F2M916: "
F2M917: "
F2M918: "
F2M919: "
F2M920: "
F2M921: "
F2M922: "
F2M923: "
F2M924: "
F2M925: "
F2M926: "
F2M927: "
F2M928: "
F2M929: "
F2M930: "
F2M931: "
F2M932: "
F2M933: "
F2M934: "
F2M935: "
F2M936: "
F2M937: "
F2M938: "
F2M939: "
F2M940: "
F2M941: "
F2M942: "
F2M943: "
F2M944: "
F2M945: "
F2M946: "
F2M947: "
F2M948: "
F2M949: "
F2M950: "
F2M951: "
F2M952: "
F2M953: "
F2M954: "
F2M955: "
F2M956: "
F2M957: "
F2M958: "
F2M959: "
F2M960: "
F2M961: "
F2M962: "
F2M963: "
F2M964: "
F2M965: "
F2M966: "
F2M967: "
F2M968: "
F2M969: "
F2M970: "
F2M971: "
F2M972: "
F2M973: "
F2M974: "
F2M975: "
F2M976: "
F2M977: "
F2M978: "
F2M979: "
F2M980: "
F2M981: "
F2M982: "
F2M983: "
F2M984: "
F2M985: "
F2M986: "
F2M987: "
F2M988: "
F2M989: "
F2M990: "
F2M991: "
F2M992: "
F2M993: "
F2M994: "
F2M995: "
F2M996: "
F2M997: "
F2M998: "
F2M999: "
F2M1000: "
F2M1001: "
F2M1002: "
F2M1003: "
F2M1004: "
F2M1005: "
F2M1006: "
F2M1007: "
F2M1008: "
F2M1009: "
F2M1010: "
F2M1011: "
F2M1012: "
F2M1013: "
F2M1014: "
F2M1015: "
F2M1016: "
F2M1017: "
F2M1018: "
F2M1019: "
F2M1020: "
F2M1021: "
F2M1022: "
F2M1023: "
F2M1024: "
F2M1025: "
F2M1026: "
F2M1027: "
F2M1028: "
F2M1029: "
F2M1030: "
F2M1031: "
F2M1032: "
F2M1033: "
F2M1034: "
F2M1035: "
F2M1036: "
F2M1037: "
F2M1038: "
F2M1039: "
F2M1040: "
F2M1041: "
F2M1042: "
F2M1043: "
F2M1044: "
F2M1045: "
F2M1046: "
F2M1047: "
F2M1048: "
F2M1049: "
F2M1050: "
F2M1051: "
F2M1052: "
F2M1053: "
F2M1054: "
F2M1055: "
F2M1056: "
F2M1057: "
F2M1058: "
F2M1059: "
F2M1060: "
F2M1061: "
F2M1062: "
F2M1063: "
F2M1064: "
F2M1065: "
F2M1066: "
F2M1067: "
F2M1068: "
F2M1069: "
F2M1070: "
F2M1071: "
F2M1072: "
F2M1073: "
F2M1074: "
F2M1075: "
F2M1076: "
F2M1077: "
F2M1078: "
F2M1079: "
F2M1080: "
F2M1081: "
F2M1082: "
F2M1083: "
F2M1084: "
F2M1085: "
F2M1086: "
F2M1087: "
F2M1088: "
F2M1089: "
F2M1090: "
F2M1091: "
F2M1092: "
F2M1093: "
F2M1094: "
F2M1095: "
F2M1096: "
F2M1097: "
F2M1098: "
F2M1099: "
F2M1100: "
F2M1101: "
F2M1102: "
F2M1103: "
F2M1104: "
F2M1105: "
F2M1106: "
F2M1107: "
F2M1108: "
F2M1109: "
F2M1110: "
F2M1111: "
F2M1112: "
F2M1113: "
F2M1114: "
F2M1115: "
F2M1116: "
F2M1117: "
F2M1118: "
F2M1119: "
F2M1120: "
F2M1121: "
F2M1122: "
F2M1123: "
F2M1124: "
F2M1125: "
F2M1126: "
F2M1127: "
F2M1128: "
F2M1129: "
F2M1130: "
F2M1131: "
F2M1132: "
F2M1133: "
F2M1134: "
F2M1135: "
F2M1136: "
F2M1137: "
F2M1138: "
F2M1139: "
F2M1140: "
F2M1141: "
F2M1142: "
F2M1143: "
F2M1144: "
F2M1145: "
F2M1146: "
F2M1147: "
F2M1148: "
F2M1149: "
F2M1150: "
F2M1151: "
F2M1152: "
F2M1153: "
F2M1154: "
F2M1155: "
F2M1156: "
F2M1157: "
F2M1158: "
F2M1159: "
F2M1160: "
F2M1161: "
F2M1162: "
F2M1163: "
F2M1164: "
F2M1165: "
F2M1166: "
F2M1167: "
F2M1168: "
F2M1169: "
F2M1170: "
F2M1171: "
F2M1172: "

```

C TOTAL ENTHALPY, AND ANGULAR MOMENTUM
C
C LAST REVISION: JULY 31, 1993
C
C VARIABLES:
C
C CHPK1() : Equal to r.Vt*K1
C CP : Specific heat at constant pressure
C D : Decay power-law exponent
C DRCOM : Average value of rho/dm over interval DM
C DISFUN : Value of the dissipation function
C DM() : Streamwise distance between QO's
C DMVRDR() : d/dr[mu.r.dVr/dr]
C DMVDR() : d/dr[mu.r.dVr/dr]
C DMVADR() : d/dr[mu.r.dVa/dr]
C DRKTRD() : d/dr[r.kt.dr/dr]
C DSDM() : ds/dm at QO i-1
C DSDM1 : ds/dm at QO 1
C DTRD() : dt/dr
C DTRDR() : dVr/dr
C DTRDR() : dVt/dr
C DTRDR() : dVt/dr
C EPSIL() : Effective mixing coefficient
C ERI : Radial stress term at QO i-1
C ERI : Radial stress term at QO 1
C ERI : Tangential stress term at QO 1
C EX() : Axial stress term at QO i-1
C EX() : Axial stress term at QO 1
C HO() : Total or stagnation enthalpy
C J : Column counter
C J : Row counter
C K1() : Eddy thermal conductivity
C K1() : Angular momentum equation evaluated at QO i-1 times dm
C K2() :
C K2() :
C LINTRP : Name of linear interpolation function subroutine
C MIXCOC : Local convective mixing coefficient
C MIXCOC : Global turbulent mixing coefficient
C MSG() : Local Mach number squared
C MUT() : Eddy viscosity
C P() : Pressure
C POSTM() : Stores linear coordinate for calculation of derivatives
C PRTPRP() : Total number of QO's
C QOMAX : Total number of QO's
C R() : Radial coordinates for grid points
C RGAS : Gas constant
C RHO() : Density
C RINT : Value of R used to interpolate convective mixing params
C S() : Entropy
C SPAN : Total number of streamlines
C SPAN : Blade span (or annulus height at inlet)
C T() : Temperature
C VM() : Meridional velocity
C VR() : Radial velocity
C VSQ : Total velocity squared
C VT() : Tangential velocity
C VX() : Axial velocity
C WU2() : (Radial/Axial velocity) squared (from program BLADE)
C WU2E : (Radial/Axial) velocities squared at inlet (from BLADE)
C X() : Axial coordinate for grid points
C XCHORD : Blade axial chord (from program BLADE)
C Z() : Spanwise coordinate for mixing parameters WU2 and U2

```

```

C
C N : Number of streamlines (must be even)
C P : 3.14159...
C R() : Radial coordinate at q-o-streamline intersection point
C R2K : Radius at point 2K
C R2K1 : " " " 2K-1
C R2K2 : " " " 2K+1
C RHO() : Density
C SMAX : Total number of streamlines in flow
C VX() : Axial velocity

```

SUBROUTINE MFLOR(SLMAX,MSUM,I)

```

C
C INTEGER N,SLMAX,I,N,K2,K2M1,K2P1
C REAL RHO,A,MSUM,P1,VX,A,B,C,R2KM1,R2K,R2KP1,F2KM1,F2K,F2KP1,DEN
C COMMON X(50,50),R(50,50),RHO(50,50),Z(50,50),S(50,50),HO(50,50),
C VT(50,50),VM(50,50),P,CP,RGAS,GAM,KAPPA(50),VX(50,50),
C ZETA(50,50),RC(50,50),PHI(50,50),P(50,50),VR(50,50),
C MSG(50,50)

```

```

C
C N=SLMAX-1
C MSUM=0.0

```

```

C DO 10 K=1,N/2

```

```

C Store needed array variables as simple variables

```

```

C K2M1=K*2-1
C K2=K*2
C K2P1=K*2+1
C R2KM1=R(I,K2M1)
C R2KR=R(I,K2)
C R2KP1=R(I,K2P1)
C F2KM1=RHO(I,K2M1)*VX(I,K2M1)*R(I,K2M1)
C F2KR=RHO(I,K2)*VX(I,K2)*R(I,K2)
C F2KP1=RHO(I,K2P1)*VX(I,K2P1)*R(I,K2P1)
C DEN=(R2KP1**2-R2K**2)*(R2K-R2KM1)-(R2K**2-R2KM1**2)*
C (R2KP1-R2K)

```

```

C Calculate local quadratic coefficients

```

```

C A=((F2KP1-F2K)*(R2K-R2KM1)-(F2K-F2KM1)*(R2KP1-R2K))/DEN
C B=((F2KP1-F2K)*(R2K**2-R2KM1**2)-(F2K-F2KM1)*
C (R2KP1**2-R2K**2))/DEN
C C=F2KM1-A*R2KM1**2-B*R2KM1

```

```

C Add local integral to sum
C MSUM=MSUM+A/3.*R2KP1**3-R2KM1**3)+B/2.*(R2KP1**2-R2KM1**2)
C + C*(R2KP1-R2KM1)

```

```

C 10 CONTINUE

```

```

C MSUM=2.*P1*MSUM

```

```

C RETURN
C END

```

```

C
C
C
C SUBROUTINE WHICH INCORPORATES RADIAL MIXING INTO THE STREAMLINE
C CURVATURE PROGRAM BY PRODUCING STREAMWISE GRADIENTS OF ENTROPY,

```

```

C      ZETA(I,J): Slope angle of streamline
C      .....
C      SUBROUTINE MIX (SIMAX,MIXCOT,MIXPTS,Z,WU2,D,XCHORD,QOMAX,ITER,
+      MIXN,MXDMP)
C      .....
C      INTEGER I,J,QOMAX,SIMAX,MIXPTS,MIXN,ITER
C      REAL MUX(50,50),EPSIL(50,50),KT(50,50),PRPTMP(201),POSTMP(201),CP,
+      DVDR(201),DVXDR(201),DVTD(201),DTRD(201),DRKTR(201),DHODM,
+      DMVDR(201),DMVDRD(201),DMVTRD(201),DSOM(50),ER(50),EX(50),I,
+      DN(50),CHIPK1(50),N1(50),K2(50),VX,VT,RHO,DISFUN,VM,DSOM1,HO,
+      R,VR,S,ERI,EXI,ZETA,VSD,P,RGAS,X,MIXCOT,2(201),ETI,DAMP,
+      WU2(201),D,SPAN,XCHORD,MIXCOC,MSQ,WUZE,LINTRP,PRINT,MXDMP
C      COMMON X(50,50),R(50,50),RHO(50,50),I(50,50),S(50,50),HO(50,50),
+      VT(50,50),VM(50,50),PI,CP,RGAS,GAM,KAPPA(50),VX(50,50),
+      ETA(50,50),RC(50,50),PHI(50,50),P(50,50),VR(50,50),
+      MSG(50,50)
C      .....
C      Calculate the effective mixing coefficient for all grid points
C      :F(MIXPTS,NE,0) THEN
C      Convective mixing only or both types of mixing
C      SPAN=R(1,SIMAX)-R(1,1)
C      DO 2 J=2,SIMAX-1
C      RINT=R(1,J)-R(1,1)
C      WUZE=LINTRP(RINT,Z,WU2,MIXPTS)
C      DO 4 I=1,QOMAX
C      :F(X(I,J),I,E,(5.*SPAN-XCHORD/2.)) THEN
C      MIXCOC=((X(I,J)+XCHORD/2.)*(1.-1.)/(5.*SPAN)*(X(I,J)-XCHORD
+      /2.))*WUZE*(1.-(X(I,J)/(5.*SPAN))*D))*VX(I,J)/
+      VX(I,J)
C      ELSE
C      MIXCOC=0.0
C      ENDIF
C      EPSIL(I,J)=MIXCOT-MIXCOC
C      CONTINUE
C      CONTINUE
C      Convective mixing coefficient is zero at walls (no radial flow)
C      Thus only turbulent mixing coefficient is used
C      DO 11 I=1,QOMAX
C      EPSIL(I,J)=MIXCOT
C      EPSIL(I,SIMAX)=MIXCOT
C      CONTINUE
C      CONTINUE
C      Calculate damped values of effective mixing coefficient
C      Calculate damped values based on iteration number
C      ITER=0
C      DO 6 J=1,SIMAX
C      DO 8 I=1,QOMAX
C      EPSIL(I,J)=MIXCOT
C      CONTINUE
C      CONTINUE
C      CONTINUE
C      DO 33 J=1,SIMAX
C      DO 35 I=1,QOMAX
C      EPSIL(I,J)=MIXCOT
C      CONTINUE
C      CONTINUE
C      CONTINUE
C      Calculate damped values
C      DO 12 J=1,SIMAX
C      DO 13 I=1,QOMAX
C      EPSIL(I,J)=MIXCOT*EPSIL(I,J)
C      CONTINUE
C      CONTINUE
C      Calculate eddy viscosity and eddy thermal conductivity for all grid
C      DO 10 I=1,QOMAX
C      DO 15 J=1,SIMAX
C      MUX(I,J)=RHO(I,J)*EPSIL(I,J)
C      KT(I,J)=MUT(I,J)*CP
C      CONTINUE
C      CONTINUE
C      Calculate necessary derivatives at inlet QO
C      Calculate dVr/dr @ inlet
C      PRPTMP(J+1)=VR(1,J)
C      POSTMP(J+1)=R(1,J)
C      CONTINUE
C      Impose boundary conditions using phantom streamlines beyond
C      hub and casing surfaces
C      PRPTMP(1)=PRPTMP(2)
C      POSTMP(1)=2.*POSTMP(2)-POSTMP(3)
C      PRPTMP(SIMAX-2)=PRPTMP(SIMAX-1)
C      POSTMP(SIMAX-2)=2.*POSTMP(SIMAX-1)-POSTMP(SIMAX)
C      CALL GRADNT(POSTMP,PRPTMP,DVDR,DVDR,SIMAX-2)
C      DO 22 J=1,SIMAX
C      DVDR(J)=DVDR(J+1)
C      CONTINUE
C      Calculate dVx/dr @ inlet
C      DO 25 J=1,SIMAX
C      PRPTMP(J+1)=VX(1,J)
C      CONTINUE
C      Impose boundary conditions
C      PRPTMP(1)=PRPTMP(2)
C      PRPTMP(SIMAX-2)=PRPTMP(SIMAX-1)
C      CALL GRADNT(POSTMP,PRPTMP,DVDR,DVDR,SIMAX-2)
C      DO 27 J=1,SIMAX
C      DVDR(J)=DVDR(J+1)
C      CONTINUE
C      Calculate dVt/dr @ inlet
C      DO 30 J=1,SIMAX
C      PRPTMP(J+1)=VT(1,J)
C      CONTINUE
C      Impose boundary conditions
C      PRPTMP(1)=PRPTMP(2)
C      PRPTMP(SIMAX-2)=PRPTMP(SIMAX-1)
C      CALL GRADNT(POSTMP,PRPTMP,DVDR,DVDR,SIMAX-2)
C      DO 33 J=1,SIMAX
C      DVDR(J)=DVDR(J+1)
C      CONTINUE
C      Calculate dT/dr @ inlet
C      DO 35 J=1,SIMAX
C      PRPTMP(J+1)=T(1,J)
C      CONTINUE
C      CONTINUE
C      Calculate damped values
C      DO 12 J=1,SIMAX
C      DO 13 I=1,QOMAX
C      EPSIL(I,J)=MIXCOT*EPSIL(I,J)
C      CONTINUE
C      CONTINUE
C      Calculate eddy viscosity and eddy thermal conductivity for all grid

```



```

      PRPTMP(SLMAX+2)=PRPTMP(SLMAX+1)
      CALL GRADNT(POSTMP,PRPTMP,DTDR,SLMAX+2)
      DO 37 J=1,SLMAX
        DTDR(J)=DTDR(J+1)
      37 CONTINUE

C Discontinue use of phantom streamlines
DO 39 J=1,SLMAX
  POSTMP(J)=R(I,J)
39 CONTINUE

C Calculate d/dr [mu.r.dvx/dr] @ inlet
DO 40 J=1,SLMAX
  PRPTMP(J)=R(I,J)*KT(I,J)*DTDR(J)
40 CONTINUE
  CALL GRADNT(POSTMP,PRPTMP,DRKTR,SLMAX)

C Calculate d/dr [mu.r.dvx/dr] @ inlet
DO 45 J=1,SLMAX
  PRPTMP(J)=MCT(I,J)*R(I,J)*DVXDR(J)
45 CONTINUE
  CALL GRADNT(POSTMP,PRPTMP,DVXDR,SLMAX)

C Calculate d/dr [mu.r.dvr/dr] @ inlet
DO 50 J=1,SLMAX
  PRPTMP(J)=MUT(I,J)*R(I,J)*DVRDR(J)
50 CONTINUE
  CALL GRADNT(POSTMP,PRPTMP,DVDRDR,SLMAX)

C Calculate d/dt, Er. and Ex at inlet QO
DO 60 J=1,SLMAX
  DISFNM=MUT(I,J)*(2.*(DVRDR(J)**2*(VR(I,J)/R(I,J))**2)+DVXDR(J)
    *2.*(DVRDR(J)-VT(I,J)/R(I,J))**2)
  DSDM(J)=(DRKTR(J)/R(I,J)+DISFNM)/RHO(I,J)/I(I,J)/VM(I,J)
  ER(J)=2.*R(I,J)*IDVDRDR(J)-MCT(I,J)*VR(I,J)/R(I,J)
  EX(J)=DVXDR(J)/R(I,J)
60 CONTINUE

C Calculate changes in Vt, S, and h0 along streamlines
DO 120 I=2,QOMAX

C Calculate dm for each streamline between QO's i-1 and i
DO 123 J=1,SLMAX
  DM(J)=SQRT((X(I,J)-X(I-1,J))**2+(R(I,J)-R(I-1,J))**2)
123 CONTINUE

C Calculate change in angular momentum r.Vt between QO's i-1
  and i using a second order Runge-Kutta method
DO 130 J=1,SLMAX
  PRPTMP(J)=MUT(I-1,J)*(DVRDR(J)-VT(I-1,J)/R(I-1,J))
130 CONTINUE
  CALL GRADNT(POSTMP,PRPTMP,DVDRDR,SLMAX)

C Calculate k1 and derivative necessary for k2 along QO i
DO 135 J=1,SLMAX
  K1(J)=DM(J)*R(I-1,J)/RHO(I-1,J)/VM(I-1,J)*
    (DMVDRDR(J)+2.*MUT(I-1,J)/R(I-1,J)*(DVRDR(J)
    -VT(I-1,J)/R(I-1,J)))
    +
    CHIPK1(J)-R(I-1,J)*VT(I-1,J)*K1(J)
    +
    PRPTMP(J+1)-VR(I,J)
    +
    PRPTMP(J+1)-VX(I,J)
    +
    PRPTMP(J+1)-CHIPK1(J)/R(I,J)
    +
    POSTMP(J+1)-R(I,J)
    +
    PRPTMP(I)=PRPTMP(2)
    +
    POSTMP(I)=2.*POSTMP(2)-POSTMP(3)
    +
    PRPTMP(SLMAX+2)=PRPTMP(SLMAX+1)
    +
    POSTMP(SLMAX+2)=2.*POSTMP(SLMAX+1)-POSTMP(SLMAX)
    +
    CALL GRADNT(POSTMP,PRPTMP,DVDRDR,SLMAX+2)
    +
    DO 140 J=1,SLMAX
      DVDRDR(J)=DVDRDR(J+1)
    140 CONTINUE

C Discontinue use of phantom streamlines
DO 147 J=1,SLMAX
  POSTMP(J)=R(I,J)
147 CONTINUE

C Calculate derivative necessary for k2
DO 145 J=1,SLMAX
  PRPTMP(J)=MUT(I,J)*(DVRDR(J)-CHIPK1(J)/R(I,J)**2)
145 CONTINUE
  CALL GRADNT(POSTMP,PRPTMP,DVDRDR,SLMAX)

C Calculate k2 along QO i
DO 150 J=1,SLMAX
  K2(J)=DM(J)*R(I,J)/RHO(I,J)/VM(I,J)*(DMVDRDR(J)
    +2.*MUT(I,J)/R(I,J)*(DVRDR(J)-CHIPK1(J)/R(I,J)**2))
150 CONTINUE

C Calculate Vt at QO i
DO 155 J=1,SLMAX
  VT(I,J)=VT(I,J)+MCDMP*((R(I-1,J)*VT(I-1,J)+
    (K1(J)+K2(J))/2.)/R(I,J))-VT(I,J)
155 CONTINUE

C Calculate change in entropy between QO's i-1 and i using second-
  order Runge-Kutta method
C Calculate necessary derivatives at QO-i
C Calculate dvT/dr @ QO-i
DO 160 J=1,SLMAX
  PRPTMP(J+1)=VR(I,J)
  POSTMP(J+1)=R(I,J)
160 CONTINUE
  CALL GRADNT(POSTMP,PRPTMP,DVDRDR,SLMAX+2)
  DO 163 J=1,SLMAX
    DVDRDR(J)=DVDRDR(J+1)
  163 CONTINUE

C Calculate dvX/dr @ QO-i
DO 165 J=1,SLMAX
  PRPTMP(J+1)=VX(I,J)

```



```

C *****
C DMSUM: Difference in DMSUM from iteration to iteration
C DR: Difference in r-coordinates of adjacent grid points
C DRVTDRI:  $d(r^2 \cdot Vt^2)/dr$  along current QO
C ISUR(I): ds/dr along current QO
C DVMDM(I,J):  $dV/dm$  for entire grid
C DX: Axial velocity added to all grid points along QO
C DXK: Difference in x-coordinates of adjacent grid points
C F2K: (density  $\times r$ ) at point 2K
C F2K1: " " " " 2K-1
C F2K2: " " " " 2K+1
C HG(I,J): Stagnation enthalpy
C I: Array column counter
C J: Array row counter
C K: Counter for numerical integration
C K2: Equal to K-2
C K2M1: " " " " K-2-1
C K2F1: " " " " K-2+1
C K1RK: First order term in Runge-Kutta formula
C K2RK: Second order " " " "
C MASS: Specified total mass flow rate through the annulus
C MDIR(I): Coordinates in the m-direction for streamline
C MSUM: Total mass flow across QO using current conditions
C MSUMOL: Value of mass flow from previous iteration along QO
C N: Number of streamtubes
C PI: 3.14159...
C QOMAX: Total number of QOs
C R(I,J): Radial coordinates of grid points
C RHO(I,J): Radius of curvature
C RHO(I,J): Density
C R2K: Radius at point 2K
C R2K1: " " " " 2K-1
C R2K2: " " " " 2K+1
C S(I,J): Entropy
C SLMAX: Total number of streamlines (including boundaries)
C T(I,J): Temperature
C VM(I,J): Meridional velocity
C VT(I,J): Tangential velocity
C VX(I,J): Axial velocity
C X(I,J): Axial coordinates of grid points
C ZETA(I,J): Streamline slope angle
C *****
C SUBROUTINE NEWVM(SLMAX,MASS,QOMAX)
C
C INTEGER I,SLMAX,J,QOMAX,N,K2,K2M1,K2F1,K
C REAL X,R,RHO,T,S,HO,VT,VM,PI,VA,ZETA,RC,PHI,SUM,MASS,MSUM,DX,DR,
C * MDIR(201),PRPTMP(201),DERTMP(201),DVMDM(50,50),DVX,K1RK,K2RK,
C * POSTMP(201),DHODR(201),DSDR(201),DRVTDRI(201),A,B,C,DEN,
C * R2KM1,R2K,K2KP1,F2KM1,F2K,F2KP1,DMSUM,DMSUMOL,MSUMOL,VNDAMP
C * COMMON VT(50,50),R(50,50),RHO(50,50),T(50,50),S(50,50),HO(50,50),
C * ZETA(50,50),VM(50,50),PI,CP,RCAS,GAN,KAPPA(50),VX(50,50),
C * VZ(50,50),PC(50,50),PHI(50,50),P(50,50),VR(50,50),
C * MSQ(50,50)
C
C Calculate dVm/dm for entire grid
C MDIR(I)=0.0
C DO 10 J=1,SLMAX
C DO 15 I=1,QOMAX
C DX=X(I,J)-X(I-1,J)
C DR=R(I,J)-R(I-1,J)
C *****
C MDIR(I)=MDIR(I-1)*SQRT(DX**2+DR**2)
C CONTINUE
C DO 20 I=1,QOMAX
C PRPTMP(I)=VM(I,J)
C CONTINUE
C CALL GRADNT(MDIR,PRPTMP,DERTMP,QOMAX)
C DO 25 I=1,QOMAX
C DVMDM(I,J)=DERTMP(I)
C CONTINUE
C *****
C Calculate new Vm (Vm) distribution at each QO
C DO 30 I=1,QOMAX
C *****
C Calculate necessary radial derivatives along QO
C Calculate dho/dr
C DO 35 J=1,SLMAX
C PRPTMP(J)=HO(I,J)
C POSTMP(J)=R(I,J)
C CONTINUE
C CALL GRADNT(POSTMP,PRPTMP,DHODR,SLMAX)
C Calculate ds/dr
C DO 40 J=1,SLMAX
C PRPTMP(J)=S(I,J)
C CONTINUE
C CALL GRADNT(POSTMP,PRPTMP,DSDR,SLMAX)
C Calculate d(rvt)/dr
C DO 45 J=1,SLMAX
C PRPTMP(J)=R(I,J)*VT(I,J)
C CONTINUE
C CALL GRADNT(POSTMP,PRPTMP,DRVTDRI,SLMAX)
C *****
C Use second-order Runge-Kutta method to solve REE along QO using
C Vt at the hub (from previous iteration)
C DO 50 J=2,SLMAX
C DR=R(I,J)-R(I,J-1)
C K1RK=DR*(DHODR(J-1)-T(I,J-1)*DSDR(J-1)-VT(I,J-1)/R(I,J-1)*
C * DRVTDRI(J-1)/VM(I,J-1)+VM(I,J-1)*K1RK)/RC(I,J-1)*SIN(PHI(I,J-1))
C * -DVMDM(I,J-1)*COS(PHI(I,J-1)))
C K2RK=DR*(DHODR(J)-T(I,J)*DSDR(J)-VT(I,J)/R(I,J)*DRVTDRI(J)/
C * (VM(I,J-1)+K1RK)+(VM(I,J-1)+K1RK)/RC(I,J)*SIN(PHI(I,J))
C * -DVMDM(I,J)*COS(PHI(I,J)))
C Calculate new value of Vm
C VM(I,J)=VM(I,J-1)+0.5*(K1RK+K2RK)
C VX(I,J)=VM(I,J)*COS(ZETA(I,J))
C CONTINUE
C *****
C Compare calculated mass flow with specified and adjust Vm profile
C accordingly by a constant amount dvx over entire profile
C CALL MFLOW(SLMAX,MSUM,I)
C *****
C ** NOTE: If it is too strict a tolerance between MASS and MSUM,
C ** the program will get stuck in an infinite loop. If this
C ** should occur increase tolerance below.

```

```

C C A PLANE DEFINED BY THREE POINTS OF THE FORM (Y,Z,P)
C C LAST REVISION: FEBRUARY 17th, 1993
C C .....
C C VARIABLES:
C C
C C A: y-coefficient in plane equation
C C B: z- " " " " "
C C C: P " " " " "
C C D: Constant in plane equation
C C P: Property value at point Y,Z
C C P1: Property value at point 1 on plane
C C P2: " " " " 2 "
C C P3: " " " " 3 "
C C U1: First component of vector from point 1 to 2
C C U2: Second " " " " "
C C U3: Third " " " " "
C C V1: First component of vector from point 1 to 3
C C V2: Second " " " " "
C C V3: Third " " " " "
C C Y: y-coordinate of point to be interpolated
C C Y1: y-coordinate of point 1 on plane
C C Y2: " " " " 2 "
C C Y3: " " " " 3 "
C C Z: z-coordinate of point of interest
C C Z1: z-coordinate of point 1 on plane
C C Z2: " " " " 2 "
C C Z3: " " " " 3 "
C C .....
C C SUBROUTINE PLANE (Y1,Z1,P1,Y2,Z2,P2,Y3,Z3,P3,Y,Z,P)
C C REAL Y1,Z1,P1,Y2,Z2,P2,Y3,Z3,P3,Y,Z,P,P1,P2,P3,Y1,Z1,Y2,Z2,Y3,Z3,P1,P2,P3
C C Determine components of vectors U and V
C C
C C U=Y2-Y1
C C U2=Z2-Z1
C C U3=P2-P1
C C V=Y3-Y1
C C V2=Z3-Z1
C C V3=P3-P1
C C
C C Compute coefficients of equation AY-BZ-CP=D
C C
C C A=U2*V3-U3*V2
C C B=U3*V1-U1*V3
C C C=U1*V2-U2*V1
C C D=(A*Y1+B*Z1+C*P1)
C C
C C Compute the value of P at point X,Z
C C
C C P=(D-A*Y-B*Z)/C
C C
C C RETURN
C C END
C C .....
C C SUBROUTINE WHICH CALCULATES THE VALUE OF P AT (Y,Z) WHICH LIES ON

```

```

C C IF (ABS (MASS-MSUM) .LE. 0.01*MS) GO TO 30
C C N=SUMAX-1
C C SUM=0.0
C C
C C DO 65 K=1,N/2
C C
C C Store needed array variables as simple variables
C C K2M1=K*2-1
C C K2=K*2
C C K2P1=K*2+1
C C R2KM1=R(1,K2M1)
C C R2K=R(1,K2)
C C R2KP1=R(1,K2P1)
C C F2KM1=RHO(1,K2M1)*R(1,K2M1)
C C F2K=RHO(1,K2)*R(1,K2)
C C F2KP1=RHO(1,K2P1)*R(1,K2P1)
C C DEN=(R2KP1**2-R2K**2)*(R2K-R2KM1)-(R2K**2-R2KM1**2)*
C C + (R2KP1-R2K)
C C
C C Calculate local quadratic coefficients
C C A=((F2KP1-F2K)*(R2K-R2KM1)-(F2K-F2KM1)*(R2KP1-R2K))/DEN
C C B=((F2KP1-F2K)*(R2K**2-R2KM1**2)-(F2K-F2KM1)*
C C + (R2KP1**2-R2K**2))/DEN
C C C=F2KM1-A*R2KM1**2-B*R2K
C C
C C Add local integral to sum
C C SUM=SUM+A/3.*(R2KP1**3-R2KM1**3)-B/2.*(R2KP1**2-R2KM1**2)
C C + C*(R2KP1-R2KM1)
C C
C C CONTINUE
C C
C C Calculate necessary change in VX across array as (to get correct
C C mass flow rate)
C C DVX=(MASS-MSUM) / (PI*SUM)
C C
C C DO 70 J=1,MSUM
C C
C C VX(J)=VX(J)+DVX
C C IF (VX(J) .LE. 0.0) THEN
C C PRINT *, 'Error: is detected if negative value of VX occurs'
C C PRINT *, ' '
C C PRINT *, '*** Negative VX encountered ***'
C C PRINT *, '*** If adjusting digital ***'
C C PRINT *, '*** PROGRAM ABORTED ***'
C C PRINT *, ' '
C C STOP
C C END
C C VX(J)=VX(J)+ABS(VX(J))
C C CONTINUE
C C
C C DO 70 J=1,MSUM
C C
C C CONTINUE
C C RETURN
C C END
C C .....
C C SUBROUTINE WHICH CALCULATES THE VALUE OF P AT (Y,Z) WHICH LIES ON

```



```

C
C      A: Grid spacing ratio
C      DS: Reference difference in ordinates
C      FI: Abscissa of middle point
C      FIM1: Abscissa of first point (point with least ordinate)
C      FIP1: Abscissa of last point (point with greatest ordinate)
C
C-----
      REAL FUNCTION DERLFW(A,FIP2,FIP1,FI,DS)
      REAL A,FIP2,FIP1,FI,DS
      DERLFW=(-FIP2*(1.+A)**2-FIP1-A*(A+2.)*FI)/A/(1.+A)/DS
      RETURN
      END
C-----

```

```

      RP=(BI-AI*DI*(Q+3.))/DELX(I-1)
      ZETA(I,STLINE)=ATAN(RP)
      PHI(I,STLINE)=PI/2.+KAPPA(I)-ZETA(I,STLINE)
      RC(I,STLINE)=1./(COS(ZETA(I,STLINE))**3*RPP(I))
      RETURN
      END
C-----
C-----
C      FUNCTION FOR FIRST DERIVATIVE (3 POINT, CENTRAL, UNEQUAL GRID)
C-----
C      VARIABLES:
C      A: Grid spacing ratio
C      DS: Reference difference in ordinates
C      FI: Abscissa of middle point
C      FIM1: Abscissa of first point (point with least ordinate)
C      FIP1: Abscissa of last point (point with greatest ordinate)
C-----

```

```

      REAL FUNCTION DERLVA(A,FIP1,FI,FIM1,DS)
      REAL A,FIP1,FI,FIM1,DS
      DERLVA=(FIP1*(A+2-1.)*FI-A**2-FIM1)/A/(A+1.)/DS
      RETURN
      END
C-----

```

```

C-----
C      FUNCTION FOR FIRST DERIVATIVE (3 POINT, BACKWARD, UNEQUAL GRID)
C-----
C      VARIABLES:
C      A: Grid spacing ratio
C      DS: Reference difference in ordinates
C      FI: Abscissa of middle point
C      FIM1: Abscissa of first point (point with least ordinate)
C      FIP1: Abscissa of last point (point with greatest ordinate)
C-----

```

```

      REAL FUNCTION DERLWM(A,FIP2,FIP1,FI,DS)
      REAL A,FIP2,FIP1,FI,DS
      DERLWM=(-FIP2*(1.+A)**2-FIP1-A*(A+2.)*FI)/A/(1.+A)/DS
      RETURN
      END
C-----
C      FUNCTION FOR FIRST DERIVATIVE (3 POINT, FORWARD, UNEQUAL GRID)
C-----
C      VARIABLES:
C-----

```


END

0 9 - 1 1 - 9 4

FIN