Chapter 3

Related Literature

In this chapter we review some relevant literature in welding metallurgy which the author had studied during the course of his research. We start with a brief discussion of the heat source models discussed by Rosenthal [1], Goldak *et. al.* [2] and Pavelic [3]. The model for cementite dissolution and homogenization of austenite, originally presented by Li, Easterling and Ashby [4] is also discussed. We conclude this chapter with a discussion on the austenite transformations as modeled by Watt *et. al.* [5] describing the decomposition of austenite into its daughter products ferrite, pearlite, bainite and martensite.

3.1 Heat Source Models

The nature of the heat source is an important factor governing heat flow during welding. Efficiency and energy-distribution of the heat source can directly affect heat flow during welding in general. A detailed description of the heat source is not intended here, however, a brief study of the basic models developed in this area is presented. Point heat source and line heat source models were chosen for study because they are idealizations requiring more assumptions than other more comprehensive models. These models are widely used in research and practise because of

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

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.

A DESIGN AND ANALYSIS ENVIRONMENT FOR WELDING

by

Arunkumar Jonnalagadda

B.Eng., Osmania University, Hyderabad, India, 1986

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

Master of Engineering

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

December 21, 1991

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

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.

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 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.

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 0-315-75965-8

Canada

The undersigned recommend to
The Faculty of Graduate Studies and Research
acceptance of the thesis
A Design and Analysis Environment for Welding
submitted by
Arunkumar Jonnalagadda
in partial fulfillment of the requirements for
the degree of Master of Engineering

Thesis Supervisor

Chairman, Department of Mechanical and Aerospace Engineering
Carleton University
Ottawa, Ontario
Canada

December 21, 1991

Abstract

Welding engineers are currently not in a position to simulate the physics of the welding process or of the welded structure. Weld designs are based on codes, experience and laboratory trials. These are expensive, slow and have limited robustness. The hypothesis of this thesis is that the capability to simulate and hence analyze the physics of the welding process and of the welded structure would reduce cost and improve quality. Equally important, it would increase robustness of design.

The objective of this thesis is to develop a software system that enables a welding engineer to simulate the physics of the welding process and of the welded structure. Specifically, the transient temperature field and the evolution of microstructure in the weld heat affected zone will be simulated for single pass welds in t-joints and butt welds and multipass welds as in girth welds on straight pipes.

Dedication

To my grandfather,
the Late Sri Gorti Kameswara Rao.

*Your lessons on hard work and
respect for the printed word
are fondly remembered.*

Acknowledgements

Sincere thanks to my supervisor, Prof. John Goldak, for his excellent guidance and enthusiastic supervision throughout my Masters program. I wish to thank my parents for their inspiration and encouragement. My solidarity of thanks are due to my sister, Mrs. Natana Valiveti and brother-in-law, Dr. Radhakrishna Valiveti for their suggestion that I come to Carleton University for a graduate degree; I am indebted to them for everything they provided me with; the long list is unending. My siblings in India, elder sister Sharada and younger brother Ramesh - thanks to both of you, you gave a lot to this.

The financial assistance provided by Carleton University in the form of teaching and research assistantship is appreciated.

I would like to thank Mr. Naci Dai, Mr. Dan Downey, Dr. Akbar Mashaie and Dr. Alan Oddy (in alphabetical order) in clarifying many issues and giving to me the benefit of their experience in managing a variety of software that was integrated in this thesis. Chapters 1 and 2 were proof-read by Chantal Lamontagne; Sherif Hegazy proof-read chapter 3; their suggestions for improvement are duly acknowledged. Finally, I would like to thank my colleagues Robert (Rob) Tjoa, David (Dave) Whitehouse and Micheal (Mike) New for their timely assistance and moral support in preparing for the defense.

I will ever be grateful to my colleague Naci Mehmet Dai for helping me structure my program and to Prof. Goldak for his patience in training me to write this thesis. Whatever merit there is in this work is due in large measure to their concepts and ideas; the remaining deficiencies are my own responsibility.

Table of Contents

Title	i
Acceptance Sheet	ii
Abstract	iii
Dedication	iv
Acknowledgements	v
Table Of Contents	vi
List of Figures	ix
List of Symbols	xi
1 Introduction	1
1.1 Creating the Framework	1
1.1.1 Overview	2
1.2 Scope of this Thesis	3
1.3 The Advantage of Objects	4
1.3.1 Incentives	6
1.4 Layout Of the Thesis	6
1.5 Bibliography	8

2	The Welding Environment	9
2.1	Introduction	9
2.2	The Environment	10
2.3	Base Metal Composition	11
2.4	Geometry of the Welded Structure	12
2.4.1	The T-Joint	16
2.4.2	Four Pass Girth Welds	17
2.4.3	Single Pass Butt Welds between Plates	18
2.5	Weld Process Selection	22
2.6	Welding Procedure	23
2.7	Weld Pool Representation and Internal Fluid Specification	26
2.7.1	Weld Pool Modeling	26
2.7.2	Internal Fluid Selection and Specification	28
2.8	Simulating the Physics	30
2.9	Conclusions and Future Work	30
2.9	Bibliography	32
3	Related Literature	33
3.1	Heat Source Models	33
3.1.1	The Point Heat Source Model	34
3.1.2	The Disc Heat Source Model	35
3.1.3	The Hemispherical Heat Source Model	37
3.1.4	The Ellipsoidal Heat Source Model	38
3.1.5	The Double Ellipsoid Heat Source Model	39
3.2	The Li-Easterling-Ashby Model	42
3.2.1	Precipitate Dissolution	42
3.2.2	Homogenization of Austenite	46
3.3	The Watt-Coon Algorithm	48

3.3.1 Austenite Decomposition	48
3.4 Bibliography	54
Appendix A	56
Appendix B	74
Appendix C	84

List of Figures

2.1	The Welding Environment.	10
2.2	Base Metal Composition.	13
2.3	Taxonomy of Steels.	14
2.4	Geometry of the Welded Structure.	15
2.5	T - Joint Finite Element Mesh.	17
2.6	(a) T-Joint Editor (b) Girth Weld Editor.	19
2.7	Sectioned Finite Element Mesh for Butt Joint between Plates	20
2.8	(a) Edge Preparation for Girth Weld (b to e) Sectioned Girth Weld Meshes.	21
2.9	(a) Edge Preparation for Butt Weld (b) Butt Weld Editor	22
2.10	Arc Welding Process Hierarchy.	23
2.11	Arc Welding Process Selection.	24
2.12	Welding Process Editors.	25
2.13	Double Ellipsoid Modeling.	27
2.14	Heat Source Shape Editor.	27
2.15	Internal Fluid Selection and Specification.	29
3.1	Schematic Set-up for Welding.	34
3.2	Circular Disk Heat Source Model.	36
3.3	Coordinate System for the Disc Heat Source Model.	37
3.4	Volumetric Specific Heat and Latent Heat Variation with Temperature.	41
3.5	Thermal Conductivity Variation with Temperature.	41
3.6	Concentration Variation as per Invariant Size Approximation.	43
3.7	Iron Rich Lower Section of the $Fe - C$ Equilibrium Diagram.	44
3.8	Cell Model for Analyzing Fe_3C Dissolution.	44
A.1	Weld World Abstraction Level.	62

A.2	Part/Piping Assembly Abstraction Level.	63
A.3	Primitives Abstraction Level.	63
A.4	Steel Selection Abstraction Level.	65
A.5	Hierarchy Of Steel Selection.	66
A.6	Process Specification Abstraction Level.	66
A.7	Arc Welding Process Abstraction Level.	67
A.8	Input File Writer.	68
B.1	Temperature Conversion Constraint Graph.	77
C.1	T Joint Sample Results - Temperature Plots.	85
C.2	Girth Weld Sample Results - Temperature Plots.	86
C.3	Butt Weld Sample Results - Temperature Plots.	87
C.4	T Joint Sample Results - Microstructure Plots.	88
C.5	Girth Weld Sample Results - Microstructure Plots.	89
C.6	Butt Weld Sample Results - Microstructure Plots.	90

List of Symbols

R^*	Radial distance from center of heat source
T	Temperature
T_0	Initial workpiece temperature
k_s	Thermal conductivity
U	Welding speed
Q	Rate of heat input into the workpiece
α_s	Thermal diffusivity
δ	Radial distance from the center of the heat source
$q(r)$	Surface flux at radius r
$q(x, y, \xi)$	Power density at (x, y, ξ)
C	Concentration coefficient
ξ, ζ	Moving coordinate systems
Q	Energy input rate
c	Characteristic radius of flux distribution
τ	Lag factor to define the position of the source at time $t = 0$
f_f	Fraction of the heat deposited in the front quadrant
f_r	Fraction of the heat deposited in the rear quadrant
a, b, c	Heat source ellipsoid dimensions
ρ	Density
ρC	Volumetric specific heat
A_1	Lower critical temperature
A_3	Upper critical temperature
$R(t)$	Radius of precipitate particle at time t
R	Radius of precipitate particle at start

α	Ferrite phase
γ	Austenite phase
Fe_3C	Cementite phase
D	Coefficient of diffusion
C	Carbon concentration
t	Time
C_0	Carbon concentration in ferrite before welding
C_γ^e	Carbon content of austenite in equilibrium with cementite
C_c^γ	Carbon content in cementite in equilibrium with austenite
r_i	Initial radius of the cementite particles
r_e	Radius of the equivalent sphere
K	Supersaturation parameter
B_1, B_2	Arbitrary Constants [Eqn. 3.19]
V_c	Terminal velocity of the $\alpha - Fe_3C$ interface
X	Volume fraction of a daughter phase [Eqn. 3.34]
B	An effective rate coefficient [Eqn. 3.34]
G	ASTM grain size number for austenite [Eqn. 3.34]
m, p	Semi-empirical coefficients set to less than 1 [Eqn. 3.34]
D_c	Diffusion coefficient for carbon
D_{eff}	Effective diffusion coefficient
C_i	Concentrations of alloying elements
PS	Pearlite-start temperature
BS	Bainite-start temperature

Chapter 1

Introduction

Much of the research on the design process in mechanical engineering is motivated by the belief that better computer-aided design tools are required. To create better tools, one should clearly understand the decision making process that works within the mind of a designer. This thesis focuses on routine design models that are computer-based; the intent is to ease the analysis for the designer and to support design decisions. The word *model* is used in two different ways; one as the cognitive model and the other as a computer model. A model is said to be cognitive to the extent that it describes, simulates or emulates the mental processes used by a designer while creating a design [1]. The difference is that in contrast to a cognitive model, this computer model has been derived from observing how humans think about a given task, but such a link is not always necessary.

1.1 Creating the Framework

The realm of pipe welding provides a rich set of problems for the welding engineer - welding layout design, analyzing heat and mass transfer, microstructural changes taking place in a welding process and the non-linear transient finite element analysis of these processes. The domain of welding design can be represented as a graph,

the nodes representing various centers of activity and the edges representing the relationships between the nodes. This discretization has been inspired by the way a designer thinks and documents his ideas to an engineer. In the following subsection, these centers of activity within the welding design domain are discussed.

1.1.1 Overview

The objective of this thesis has been to design a software environment for the three dimensional finite-element heat transfer and microstructure analysis of welds. The principle behind this user interface was to enable the welding engineer to directly simulate the proposed weld design, i.e., to visualize its transient temperature history and microstructure. The simulation was to be based on the physics of welding and not on regression equations fitted with data. Here, those areas of paramount importance in welding design are identified.

The applications domain of welding can be identified to consist the following: *pipeline, petrochemical, structural, automotive and railway*, to mention a few. The fundamental categorization of structure on the basis of geometry can be identified as: *piping, structural, plate, sheet and rod*. Joint function criteria may include *type of loading, magnitude of loading, service requirements and life*. The type of loading can be either tensile, shear, compressive, bending or fatigue. The loading can be either constant or fluctuating. The service requirements greatly depend on the environment which may be corrosive, high or low temperature, marine or a combination of these. The life of a weld may be an arbitrary number based on the number of load cycles, years to failure or allowable modes of failure or even on the risk to human life. The geometry of the joint can be said to be comprised of three attributes, namely joint type (e.g. butt, lap or tee), joint position (e.g. flat, horizontal or vertical) and joint geometry (e.g. V, Y or K).

The material chemistry of the joint materials may be characterized by the following: alloy type (e.g. low alloy steel, aluminium), alloy number, microstructure

(ferrite, pearlite, bainite or martensite and the fraction of each phase present) and carbides (NbC, VC, TiC). Every phase has a grain size, a characteristic shape and distribution.

Material properties of the joint materials may be characterized as either physical, thermal or mechanical. These properties may be functions of temperatures and microstructure. Physical properties include density; thermal properties include thermal conductivity, specific heat, latent heats of transformation and mechanical properties include Young's modulus of elasticity, Poisson's ratio, the material's yield strength and plastic modulus. These are a few of the important properties that are required.

The welding procedure includes a weld process, weld parameters and consumables.

A weld design specifies the above entities as a design graph and the associated parameters; this environment provides default values since newly designed welds are in most cases modifications of existing welds.

1.2 Scope of this Thesis

The vastness of the welding design domain restricts the scope of this thesis to three basic types of weld joints, namely: (1) straight-line welds such as a butt-joint between plates, (2) multipass girth welds as in the joining of pipes, limited to four passes of weld bead deposits, and (3) saddle-shaped fillet welds as in t-joints, when connecting two pipes with their axes intersecting at right angles.

This environment provides a finite-element mesh for the weld being performed. Meshes are produced in INGRID [2], a powerful mesh-generating software. Mesh generating scripts are one of the outputs in the environment. These meshes are initially coarse, more so to capture the geometry of welds and their surroundings, imitating real life situations. These are then refined locally to finer mesh sizes along

regions of particular interest, namely the path of the heat source.

This environment integrates five different kinds of software entities:

1. The exterior is manipulated through Smalltalk-80, a class-based object-oriented environment [3].
2. Constraint satisfaction for the piping assembly layouts involving straight pipes and t-joints is achieved through declarative constraints [4].
3. Finite element meshes are produced using INGRID [2].
4. The finite element solver code for heat transfer and microstructural analysis is written in FORTRAN 77.
5. Structured mesh manipulation code is written in C.

1.3 The Advantage of Objects

Smalltalk was invented to meet the needs of the user interface. The figures that follow in later chapters clearly show that most modern concepts of the user interface design (including direct manipulation of several parameters) are provided in Smalltalk; these concepts are exploited in this environment. The most common mistake in designing a user interface is to design it after the application program is complete. This strategy invariably fails because the user is the most important person as far as the interface is concerned. The user's needs should drive the design of an applications program from the start to the end in a software development cycle. This happens naturally in Smalltalk but is very unusual in procedural languages.

Objects are much closer to cognitive models used by welding engineers than procedural programs. Also the combinatorial explosion in procedural languages quickly overwhelms the welding engineer; this can be greatly delayed in object-based systems since data-abstraction barriers are clearly defined with objects and less code is

required. Object-oriented programming has overcome many of the accidental complexities that were present in procedural programming. Behavior can be separated into two categories: internal behavior or private behavior and external behavior. A manager may not care to know how the employee accomplishes the task, only that it be done properly and on schedule. In fact, a manager would be overwhelmed if he had to know every detail of what was done. Thus, objects and message passing are a good means of managing complexity by constructing abstraction barriers.

In procedural programming, one begins by designing an algorithm or procedure to solve a specific type of problem. When it is complete, an input file is prepared and the program is tested; if the results are erroneous or if the specification of the problem changes its structure, the program will have to be rewritten. In an object-oriented design, just the opposite happens. Each object is created and tested, independent of the others. When the desired world of objects is complete, test programs consist of simply telling the objects what their expected behavior is. If new methods and messages are needed to incorporate changes in the problem specification, one can add them without having to change the representation of the objects. Such modifications, minor or major, can be brought into effect very easily through incremental improvement stages.

Welding is a means of assembling parts and structures. A weld can be considered as a constraint; in most cases welds are meant to transfer loads or stresses between parts. Every part and parts of assemblies can be seen as individual objects being pieced together through a set of relations and constraints. The welding domain was chosen because it is complex enough to be challenging and at the same time not so complex that it is unmanageable. In this thesis, an attempt is made to represent the domain of welding with objects and we were successful only to a very limited extent in this endeavor. No attempt has been made to create a complete and comprehensive computer-integrated manufacturing system. Nonetheless the advantages of having such a small scale software environment are being realized by our research group.

1.3.1 Incentives

Having such an interface saves the user anxiety in creating bits and pieces of results and connecting them through all the intermediate stages, especially between procedural programs that communicate by file reformatting. The process of changing the output of one program to suit the input requirements of another are taken care of by the interface.

Another incentive is the automatic production of finite element meshes by the environment which saves the user much frustration and time. As a matter of fact, the number of man hours required to produce a pipe mesh is very large compared to the running time for even the most typical of problems in the domain of pipe-welding; a rough estimate is a ratio of 50 : 1 as it was in the case of the author.

1.4 Layout Of the Thesis

- Chapter 1 addresses the basic issues in welding engineering such as managing the body of knowledge and the inherent software complexity and how these have influenced progress in welding research.
- Chapter 2 introduces the exterior of the environment through a series of screen-dumps; the user interface is described and explanations are given for the various representations adopted in this thesis. This chapter is specifically meant for the welding engineer who uses this environment everyday.
- Chapter 3 discusses the theory behind the simulation of welding physics. A few papers from welding metallurgy literature are reviewed.
- Appendix A discusses a representation of the welding design domain using collapsed subgraphs. This should be of more interest to the software designer than the welding engineer.

- Appendix B describes briefly the support software for this thesis, namely Declarative Constraints, Ingrid and Smalltalk-80, presenting some basic concepts and ideas.
- Appendix C is a presentation of results that were achieved by using the interface on a sample collection of problems.

Bibliography

- [1] Laird, J.E., Newell, A., Rosenbloom, P.S., *SOAR: An architecture for general intelligence*, AI Journal Vol. 33, No. 1, pages 1-64, September 1987.
- [2] Rainsberger, Robert, *INGRID, A Three-dimensional Mesh Generator for Modeling Nonlinear Systems*, Lawrence Livermore National Laboratory, August, 1988.
- [3] Goldberg, A., Robson, D., *Smalltalk-80 The Language*, Addison-Wesley Publishing Co., 1989.
- [4] Borning, A.H., *ThingLab - A Constraint-Oriented Simulation Laboratory*, Ph.D. dissertation, Department of Computer Science, Stanford, March 1979.

Chapter 2

The Welding Environment

2.1 Introduction

This chapter describes in detail the welding environment that was developed to assist the welding engineer in the design of weld joints. This is done by analyzing the physics of the welding process namely the transient temperature history and the phase fractions in the weld heat affected zone. One word of caution to the reader - this chapter is aimed at the welding engineer. Throughout this documentation, the word user means the welding engineer unless stated otherwise. The following sections discuss specific aspects covered by the environment. An attempt is made to break the user's problem into manageable pieces; these pieces are identified and represented in the environment as distinct entities. The following sections discuss the details of the user interface and present the welding design problem representation.

2.2 The Environment

The welding environment as it appears on the workstation terminal is shown in Fig. 2.1.

Figure 2.1: The Welding Environment.

Attempting to analyze a welding problem, the four essential pieces of information on which the user operates while designing weld joints are: (1) the base metal composition, (2) the geometry of the welded structure, (3) the welding process and procedure, and (4) the modeling of the physics of the welding process, i.e., the results generated by this environment.

The environment is user-friendly; it is menu-driven, very similar to the *Macintosh* interface. Menus are pulled out by the user as s/he clicks on the menu buttons. The four elements that were identified are shown as menu buttons on the left hand

side top corner of the environment. The top row of check boxes is provided for manipulating the display in the viewing area. The check box "Bounds" displays the bounding box of the mesh; the user can zoom in or zoom out by depressing the middle mouse button and moving the mouse. The bounding box can be rotated by pressing the left mouse button and moving the mouse. The check box "Plain" displays the finite element mesh in the viewing area. The check box "Surfaces" provides the option to alternately highlight the surfaces on the mesh, by clicking on the "Next Surface" button.

The aforementioned material, geometry, weld process and physics can be thought of as a set and its constituent elements; in other words, these elements can be said to span a four dimensional space. The welding engineer's domain is precisely this space and by specifying the values of the parameters, s/he specifies the weld joint design. In the following sections, these bases are discussed in detail, while describing the environment.

2.3 Base Metal Composition

Specifying the base metal composition is probably the first step in attempting to model the physics of the welding process. The composition of the base metal greatly influences the choice of welding process. If the base metal is aluminium or magnesium, the process of choice would either be gas tungsten arc or gas metal arc welding. When the base metal is carbon or low alloy steel, selection is more complex due to the fact that more welding processes become applicable.

This environment offers steel as the only base metal in its options. Clicking on the button "Base Metal Selection", the user gets a menu for material selection and review. The user is first required to select the base metal after which s/he can review the selection for any further changes. The base metal composition specification in the environment is as shown in Fig. 2.2. Clicking on the "Material Select" button

brings forward a dialog box with the basic option "Steels". The taxonomy of steels is shown in Fig. 2.3. The user can march down or up the taxonomy of steels by clicking on the "More" or "Up" buttons respectively. Once a metal is selected, the user can signal completion of base metal selection by clicking on the "OK" button; the dialog box disappears immediately. Compositions of commonly used engineering materials are stored in a materials database within the Smalltalk-80 environment and are automatically brought forward for verification by the user when s/he clicks on the "Material Review" button.

2.4 Geometry of the Welded Structure

The geometry of the structure can be described through primitive objects available in the environment. The geometries of three fundamental structures currently available are: (1) saddle-shaped fillet welds as in t-joints, (2) multipass girth welds between straight pipes, and (3) single pass butt joints between plates. These can be used directly if the user wishes to model the welding physics of the primitive individually. There is an option for the user to model the geometry of an assembly of pipes with girth welds and t-joints. This is shown in Fig. 2.4.

When the user clicks on the menu button "Geometry", a menu unfolds with three options; these are "Existing Individual Geometries", "Create Assembly" and "Welded Part Mesh". Under "Existing Individual Geometries", three primitives are shown in a sub menu. The user clicks on the geometry of his/her choice and the appropriate editor pops up on the screen.

When the user wishes to model an assembly, s/he could click on the "Create Assembly" menu button. This pops up an editor for creating assemblies involving girth welds and t-joints. This is shown as "A Piping Assembly Model" in Fig. 2.4. The girth weld and t-joint primitives have an annular interface where their individual meshes match face to face to make a valid assembly. Clicking on the "T-Joint" menu

Figure 2.2: Base Metal Composition.

Figure 2.3: Taxonomy of Steels.

button brings up a t-joint icon; likewise, a girth weld icon can be obtained by clicking on the "Girth Weld" menu button. As an example, these have been assembled in Fig. 2.4 by constraining the right annular interface of the girth weld to be in contact with the left annular interface of the t-joint bottom pipe. Any icon on the piping assembly editor can be selected by clicking on it; this places a box around the icon, as shown around the t-joint icon in Fig. 2.4. Icons can be moved around the editor area by first selecting, and moving the mouse keeping the left mouse button depressed. Any primitive can be constrained to another by constraining the appropriate ports. These ports defined for any assembly are listed in the assembly editor below the icon menu buttons. In the example of a piping assembly, only the annular interface port needs to be constrained. Any port can be selected by placing the cursor on the port and clicking the left mouse button. Two ports can be constrained to each other by selecting a port on one icon and pulling it, keeping the left mouse button pressed, onto the opposite port on another icon. When the ports match, the line between them is highlighted and the arrow cursor changes shape to a + (plus). Once the mouse button is released, an arrow appears (as shown in Fig. 2.4) indicating the

Figure 2.4: Geometry of the Welded Structure.

direction of constraint flow. Constraint flow can be in either directions; this does not confuse the environment when creating assemblies.

It should be noted that assembling these primitives can take place only along the x -axis. A girth weld primitive cannot be assembled on top of the vertical pipe of the t-joint; it can be assembled at either the left or the right annular area on the bottom pipe of the t-joint. This is due to limitations on the translation and rotation of meshes in the meshing software INGRID [8]. In this environment finite element meshes are produced for such simple assemblies; the environment does not currently produce input files for such assemblies.

The third option “Welded Part Mesh” extends the flexibility of this environment. If the user has an output file generated by other meshing software in a format acceptable to this environment, then its analysis capabilities can be utilized without having to create new geometries and their mesh script generators.

The available individual geometries are described in the following subsections and limitations in generating their meshes is discussed.

2.4.1 The T-Joint

A pipe is specified by its length and any two of its outer diameter, inner diameter or wall thickness. In t-joints, meshes are provided for one pass of weld bead deposit. The finite element mesh provided by the environment is shown in Fig. 2.5. Multipass weld deposit simulation is not yet available in t-joints.

This mesh is particularly intended for modeling welding on pipes carrying some fluid within them. The editor for modifying the dimensions of the t-joint is as shown in Fig. 2.6 (a). All dimensions are to be input in centimeters (a meter is quite a large unit of length and I have tried my best to avoid using inches).

The environment cannot create meshes when the outer diameters of the top and bottom pipes are equal; this is due to limitations in the mesh generating software INGRID [8]. T Joint analysis results are shown in Appendix C.

Figure 2.5: T - Joint Finite Element Mesh.

2.4.2 Four Pass Girth Welds

Straight pipes are defined by their lengths and any two of their internal diameter, external diameter or wall thickness. The editor for girth welds on straight pipes is shown in Fig. 2.6 (b). A weld joint is defined by an edge preparation and the number of passes. Standard edge preparations are assumed, i.e. root gap = $\frac{1}{8}th$ inch, root face = $\frac{1}{8}th$ inch, theta = $30 (-0,+7.5)^{\circ}$ as shown in Fig. 2.8 (a). All dimensions are to be input in centimeters.

For girth welds on pipes, multipass weld physics can be simulated in this environment. Finite element meshes are produced automatically for up to four passes. Topology of the finite element meshes for the four passes is shown in Fig. 2.8 (b through e). These passes are shown in differing degrees of coarseness for the sake of clarity. Uniform meshes are produced when the user runs a job on girth weld analysis. Here it is assumed that the girth welds are performed without a backup ring;

the groove is hence narrow. Since the number of passes depends on the thickness of the pipes being joined, it was decided to restrict the maximum number of passes to four. The user is responsible for ensuring that a maximum of four passes is sufficient to complete the joint. It should be noted that the user is first required to select the number of passes that s/he wishes to analyze. The environment produces a finite element mesh for the geometry with the appropriate number of passes selected by the user. The user should then choose the proper size of the heat source model so that melting and deposition of the weld bead is correctly simulated for the pass under consideration. The heat source dimensions vary from one pass to another as the requirements for each pass are different; the heat source dimensions for the fourth "cover" pass cannot be the same as that of the first "stringer bead" or second "hot" passes since the spread of weld metal deposit is more at the top in the fourth pass than at the root gap during the stringer bead [2, 3].

As an example, a three pass girth weld has been considered. For the sake of speed in demonstrating this example, only a quadrant of the pipes is considered. The analysis had taken about 3 hours on a Sparc II workstation; a full scale example would approximately take 2 to 3 days .

2.4.3 Single Pass Butt Welds between Plates

A plate can be defined by its thickness, length and breadth. The finite element mesh available in the environment for the simulation of a single pass butt joint between plates is shown in Fig. 2.7. The weld joint is defined by an edge preparation and a specific number of passes. By convention, while butt-welding plates, a pass is given on the underside also; the bottom pass has not been included in the mesh that is provided by this environment. The weld pool is modeled as part of a sphere dipping into the surface of the workpiece, as shown in Fig. 2.7. The radius of the circle on the plane of intersection of the top surface of the plate and the sphere is taken equal to the plate thickness; the center of the sphere is assumed to be at a height

(a)

TJoint

Attributes Of Bottom Pipe (in cms):

Bottom Pipe Outer Diameter: 20

Bottom Pipe Inner Diameter: 18

Bottom Pipe Length: 50

Bottom Pipe Wall Thickness: 1

Attributes Of Top Pipe (in cms):

Top Pipe Outer Diameter: 12

Top Pipe Inner Diameter: 10

Top Pipe Length: 15

Top Pipe Wall Thickness: 1

> OK

(b)

GirthWeld

Number Of Passes

One Pass ☐

Two Passes ☐

Three Passes ☐

Four Passes ☒

Pipe Dimensions (in cms):

Inner Diameter: 12

Outer Diameter: 14

Length Left Of Weld: 15

Length Right Of Weld: 15

Wall Thickness: 1

Figure 2.6: (a) T-Joint Editor (b) Girth Weld Editor.

Figure 2.7: Sectioned Finite Element Mesh for Butt Joint between Plates

equal to the plate thickness above the plate surface. Hence the radius of the sphere is computed to be equal to $\sqrt{2} \times \text{thickness}$. This proportionality was adopted for its balanced geometry after several other ratios were tried, though with less success. Once again, it is the user's responsibility to ensure that one pass is sufficient to complete the butt joint. Typically, one pass on the top would be sufficient on steel plates of thicknesses up to $\frac{1}{2}$ inch. Standard edge preparations are assumed here as shown in Fig. 2.9 (a). The editor for modifying the dimensions of the plate is shown in Fig. 2.9 (b). All dimensions are to be input in centimeters.

These butt weld meshes are mainly intended for the prescribed temperature steady state finite element analysis which is done for any one time increment at a time in Smalltalk; the analysis software is currently being upgraded. Results in Appendix C are intended to give the reader an idea of what the results would be once this step is completed; these results were obtained by placing a spherical heat source on the weld pool; the transient temperature contours are shown.

Figure 2.8: (a) Edge Preparation for Girth Weld (b to e) Sectioned Girth Weld Meshes.

Figure 2.9: (a) Edge Preparation for Butt Weld (b) Butt Weld Editor

2.5 Weld Process Selection

After considering base metal and geometry of the welded structure, the next step could be the selection of the welding process to be adopted for manufacturing the joint. The taxonomy of arc welding processes is as shown in Fig. 2.10.

For selecting the arc welding process, the user is usually required to eliminate the unsuitable processes by traversing the tree on a unique path. In this environment, the four arc welding processes used very often in industry namely, Gas Metal Arc Welding, Shielded Arc Welding, Submerged Arc Welding and Electroslag Welding are considered. The user clicks on the "Welding Process" button and a menu unfolds with a prompt for either "Process Selection" or "Process Review". Clicking on the process selection button brings up a dialog box indicating "Arc Welding". Selecting "Arc Welding" and clicking on the "More" button brings forward the four welding processes. See Fig. 2.11. The user can select the process of his/her choice and click on the "OK" button to signal completion of the welding process selection; the dialog box disappears immediately. The user can review the selected process for its parameters by clicking on the "Process Review" button. The environment brings up default values that are specified for the process variables. The user has the option to modify the defaults provided by keying in his/her choice and hitting <return>

(Adapted from Blodgett , O.W., [4])

Figure 2.10: Arc Welding Process Hierarchy.

key to register the change; this is discussed in the next section in detail.

2.6 Welding Procedure

Once the process is decided, the next step would be the specification of welding process parameters. The fundamental parameters prescribed by the user for any arc welding application are: (1) current (amperes), (2) voltage (volts), (3) arc speed (mm/sec), (4) arc efficiency (-). These parameters for the various processes can be specified through the editors that pop up when the user clicks on the "Process Review" button. These are shown in Fig. 2.12. Some points have to be mentioned before the user makes any assumptions. The class of the electrode and its size are prompted for in the editors; these are not taken into account by the environment for any of the finite element computations. Nor is the flux used in submerged arc

Figure 2.11: Arc Welding Process Selection.

GasMetalArcWelding Editor	
The Welding Process Details	
Welding Process Parameters	Pass 1
Current (A)	120
Voltage (V)	20
Arc Efficiency (-)	0.75
Arc Speed (mm/sec)	3
Angle (deg):	45
Gas (name):	HELI
Pre-Heat (deg. C):	25
Electrode Class (AWS Spec.):	AWS
Electrode Size (mm):	5
Ambient Temperature (deg. C):	25

ShieldedArcWelding Editor	
The Welding Process Details	
Welding Process Parameters	Pass 1
Current (A):	120
voltage (V)	20
Arc Efficiency (-):	0.75
Arc Speed (mm/sec):	3
Stickout (mm):	3
Angle (deg):	45
Pre-Heat (deg. C):	25
Electrode Class (AWS Spec.):	AWS
Electrode Size (mm)	5
Ambient Temperature (deg. C)	25

ElectroslagArcWelding Editor	
The Welding Process Details	
Welding Process Parameters	Pass 1
Current (A):	120
Voltage (V):	20
Arc Efficiency (-):	0.75
Arc Speed (mm/sec):	3
Pre-Heat (deg. C):	25
Electrode Class	AWS
Electrode Size	5
Ambient Temperature (deg. C):	25

SubmergedArcWelding Editor	
The Welding Process Details	
Welding Process Parameters	Pass 1
Current (A):	120
Voltage (V):	20
Arc Efficiency (-):	0.75
Arc Speed (mm/sec):	3
Stickout (mm):	nil
Flux (name):	ABC
Pre-Heat (deg. C):	25
Electrode Class (AWS Spec.):	AWS
Electrode Size (mm)	5
Ambient Temperature (deg. C):	25

Figure 2.12: Welding Process Editors.

welding processes. These are provided for the sake of completeness and record.

When a multipass girth weld is simulated, the user is provided with an editor that prompts him/her for process parameters for every pass. Each pass is treated as a separate process in this environment. Due to time limitations, a pass-by-pass analysis option for the girth weld which would take into account effects of the previous passes while simulating the physics of the current pass is not included. As a result, only the last pass parameters are taken into account for FEM analysis.

2.7 Weld Pool Representation and Internal Fluid Specification

Representation of the weld pool and the internal fluid flowing is discussed in this section. Current and voltage, the first two terms described in Sec. 2.6, represent heat input during welding. The heat source models developed so far by other research groups are discussed in the next chapter. Here the model used in this environment is described briefly. Specification of the fluid flowing inside the pipe is vital when welding "live" pipes as in a t-joint; this is often encountered in the pipeline industry.

2.7.1 Weld Pool Modeling

In this environment, the heat source is modeled as a double ellipsoid, as shown in Fig. 2.13. This model is the one that best approximates the tear drop shape of weld pools. The underlying theory is discussed in depth in later chapters. The editor for the heat source shape is shown in Fig. 2.14.

The user is required to specify the front and rear radii of the ellipsoid for representing the weld pool; weld pool dimensions can be estimated from formulae provided by Christensen *et. al.* [4]. The user should ensure that the double ellipsoid dimensions are appropriate. If the dimensions are too large the peak temperature

Figure 2.13: Double Ellipsoid Modeling [5].

Heat Source	
Heat Source Modelling	
Front Rx (cm)	<input type="text" value="0.5"/>
Front Rz (cm)	<input type="text" value="0.5"/>
Rear Rx (cm)	<input type="text" value="0.65"/>
Rear Ry (cm)	<input type="text" value="0.5"/>
Rear Rz (cm)	<input type="text" value="0.5"/>
Number of Welders Employed	
Number Of Heat Sources:	<input type="text" value="1"/>
Fraction Of Heat Dissipated	
Front Fraction:	<input type="text" value="0.3"/>
Rear Fraction:	<input type="text" value="0.7"/>

Figure 2.14: Heat Source Shape Editor.

may not be high enough to effect melting. On the other hand if the size is too small, temperatures may be unreasonably high.

When modeling girth welds, the user should specify the number of heat sources if more than one welder is operating along the girth. On pipes of 18 inches diameter or more, four welders are assumed to start from the 12, 3, 6 and 9 o'clock positions along the circumference.

The user is required to specify the fraction of heat dissipated from the front and rear halves of the double ellipsoid; the defaults provided in the menu against "Front Fraction" and "Rear Fraction" options are 0.7 and 0.3 respectively.

When the heat source is placed at the start node, the system should be permitted to form a weld pool as soon as the arc is struck between the electrode and the workpiece. This stage typically lasts for 4 to 6 seconds for steels when the heat source is kept stationary at the starting point. This information is hardwired into the environment as it is tuned only for steels as base metal.

2.7.2 Internal Fluid Selection and Specification

Welding physics simulation gets more involved if a pipe carrying a liquid has to be welded to another pipe. In such cases, care should be taken to avoid burnthrough on the live pipe at the point where the arc is started. The user is required to specify the fluid type, velocity, temperature and pressure for the FEM heat transfer calculations. Currently the user has the option to specify three kinds of fluids namely air, methane and propane. The user has to click on the appropriate check box and select the fluid. The fluid type and properties are prompted for by the environment as shown in Fig. 2.15.

Weld Analyzer

Menu ☒ Bounds ☐ Plain ☐ Surfaces > next surface

Base Metal Selection >

Geometry **heat source dimensions**

Welding Process **internal fluid options**

Fluid options **internal fluid properties**

Solver

Result Visualization

Fluid

Fluid Name

Air ☒

Methane ☐

Propane ☐

Fluid

Fluid Properties

Fluid Velocity (m/s):

Fluid Temperature (deg. C):

Fluid Pressure (atm):

heat source dimensions
internal fluid options
internal fluid properties

Figure 2.15: Internal Fluid Selection and Specification.

2.8 Simulating the Physics

The user can simulate the welding process by clicking on the "Solver" menu button. This takes quite some time; hence the user should be doubly sure at this point to have selected the process parameters and all relevant information. Even otherwise, the environment runs a check to ascertain whether the user has specified all required information.

The output of the environment consists of two binary files GNU and GNU1 in a format that is readable by the post processor TAURUS. These files essentially consist of two blocks; the first block contains the geometry and the second block contains a time increment and the corresponding temperature profiles and phase fraction contours. When the user clicks on the "Result Visualization" menu button, the environment produces a new file that serves as input to the *SolidView* software which produces colored objects on the workstation. When using this environment, the user should first experiment with a set of reasonable values for power input, heat source dimensions, fluid velocity and temperature. After a few iterations the solver should be interrupted and the results checked on the database. If the contours are found to be reasonable, then the user may run a full fledged analysis.

This completes the existing options in the environment. The next chapter is a discussion on some theoretical aspects of welding metallurgy that help model the welding physics.

2.9 Conclusions and Future Work

A variety of software was integrated in this environment. Input file preparation for the finite element analysis of welds was effected by only one or two experts in about 15 to 20 days. Now, a lay engineer can create input files in about 30 minutes without having to know the details of the analyzing software. Moreover,

finite element meshes are produced automatically by the environment which results in a huge saving in time. Possible extensions of this thesis are listed here.

1. Creating a new t-joint mesh to handle the special case when the outer diameters of the top and bottom pipes are equal.
2. Multipass weld simulation could be extended to include the physics of the previous passes.
3. Sleeve type girth welds (AGA welds) can be included in the list of available geometries.
4. Effect of liquid metal motion in the weld pool can be incorporated as and when the physics becomes analyzable.

Bibliography

- [1] Rainsberger, Robert, *INGRID, A Three-dimensional Mesh Generator for Modeling Nonlinear Systems*, Lawrence Livermore National Laboratory, August, 1988.
- [2] *The Procedure Handbook of Arc Welding*, The Lincoln Electric Company, Cleveland, Ohio, 1973.
- [3] Blodgett, O.W., *Design of Weldments*, The James F. Lincoln Arc Welding Foundation, Cleveland, Ohio, 1976.
- [4] Christensen, N., V de L Davies, Gjermundsen, K., *British Welding Journal* Vol. 7, pages 54-75, 1965.
- [5] Goldak, J.A., Chakravarti, A., Bibby, M.J., *Metallurgical Transactions B*, AIME/ASM, Vol. 15 B, page 299, 1984.

Chapter 3

Related Literature

In this chapter we review some relevant literature in welding metallurgy which the author had studied during the course of his research. We start with a brief discussion of the heat source models discussed by Rosenthal [1], Goldak *et. al.* [2] and Pavelic [3]. The model for cementite dissolution and homogenization of austenite, originally presented by Li, Easterling and Ashby [4] is also discussed. We conclude this chapter with a discussion on the austenite transformations as modeled by Watt *et. al.* [5] describing the decomposition of austenite into its daughter products ferrite, pearlite, bainite and martensite.

3.1 Heat Source Models

The nature of the heat source is an important factor governing heat flow during welding. Efficiency and energy-distribution of the heat source can directly affect heat flow during welding in general. A detailed description of the heat source is not intended here, however, a brief study of the basic models developed in this area is presented. Point heat source and line heat source models were chosen for study because they are idealizations requiring more assumptions than other more comprehensive models. These models are widely used in research and practise because of

Figure 3.1: Schematic Set-up for Welding [6].

their simplicity. We briefly discuss each one of the heat source models and touch some of the salient points.

3.1.1 The Point Heat Source Model

Rosenthal [1] was one of the first researchers to have attempted modeling the heat source. He represented the heat source as a point on the surface of an infinite workpiece. The equations he had derived were based on the assumption that an observer moving with the heat source does not see any change in the weld pool geometry and temperature distribution with respect to time. This steady state assumption was used to simplify the mathematics of heat flow during welding. His equation for 3-D heat flow is:

$$\frac{2\pi(T - T_0)k_s R^*}{Q} = \exp\left(\frac{-U(R^* - x)}{2\alpha_s}\right) \quad (3.1)$$

where R^* is the radial distance from the origin $(x^2 + y^2 + z^2)^{1/2}$, T the temperature, T_0 the initial workpiece temperature, k_s the thermal conductivity of the solid, U the welding speed, Q the rate of heat input into the workpiece, α_s the thermal diffusivity of the solid. Refer Fig. 3.1. The assumptions in this model were:

1. the temperature distribution is steady state,

2. heat is delivered by a point source on the weld centerline,
3. thermal properties remain constant with temperature,
4. heat of melting and fusion are negligible,
5. there are no heat losses through the surface, and
6. speed of the source and rate of heat input are constant.

The last two assumptions are reasonable but the first three are approximations, introducing errors; by assumption 1, peak temperatures along the weld centerline approach infinity; by assumption 2, thermal properties are not dependent of temperature and by assumption 3, latent heats do not influence the phase transformations in the fusion zone and the heat affected zone.

3.1.2 The Disc Heat Source Model

A comprehensive study of distributed heat sources is contained in Rykalin [7]. An advanced distributed heat source (disc) model suggested by Pavelic *et. al.* [3] is discussed here (see Fig. 3.2). The thermal flux has a Gaussian or normal distribution:

$$q(\delta) = q(0)e^{-C\delta^2} \quad (3.2)$$

where $q(\delta)$ is the surface flux at radius δ (W/m^2), $q(0)$ the maximum flux at the center of the heat source (W/m^2), C the concentration coefficient (m^{-2}), δ the radial distance from the center of the heat source (m).

A simple physical meaning can be associated with C . If a uniform flux of magnitude $q(0)$ is distributed in a circle of diameter $d = 2/\sqrt{C}$, the rate of energy input would be ηIV , where η is the efficiency, I the current (amp) and V the voltage (volt), i.e. the circle would receive exactly the energy from the arc. Therefore, the

Figure 3.2: Circular Disk Heat Source Model [3].

coefficient C is related to the source width; a more concentrated source would have a smaller diameter d and a larger value of C .

$$q(\delta) = q(0)e^{-\delta^2/a^2}$$

when $\delta = 0$; $q(\delta) = q(0)$ & when $\delta = a$; $q(a) = q(0)e^{-1}$ meaning at a radial distance a , q decays to $q(0)e^{-1}$. Refer Fig. 3.2.

Experiments have shown that a significant amount of heat is transferred by radiation and convection from the arc directly to the solid metal without passing through the molten pool. Based on this observation, Pavelic *et. al.* [3] developed a correlation showing the amount and distribution of this heat over the solid material. In their study, provisions were made for convective and radiative losses from the heated metal plate to the surroundings as well as variable material properties.

Friedman [8] and Krutz *et. al.* [9] suggested an alternative form of the Pavelic 'disc'. Expressed in a coordinate system ξ that moved with the heat source as shown in Fig. 3.3, eqn. 3.2 takes the form:

$$q(x, \xi) = \frac{3Q}{\pi c^2} e^{-3x^2/c^2} e^{-3\xi^2/c^2} \quad (3.3)$$

Figure 3.3: Coordinate System for the Disc Heat Source Model [2].

where (x, ξ) is the moving coordinate system characterized by $\xi = z + U(\tau - t)$, Q the energy input rate (W), c the characteristic radius of flux distribution (m) and τ being a lag factor needed to define the position of the source at time $t = 0$, U the welding speed (m/s). In the (x, y, z) coordinates the equation takes the shape:

$$q(x, z, t) = \frac{3Q}{\pi c^2} e^{-3x^2/c^2} e^{-3[z+U(\tau-t)]^2/c^2} \quad (3.4)$$

for $x^2 + \xi^2 < c^2$. For $x^2 + \xi^2 > c^2$, $q(x, \xi, t) = 0$.

To avoid the cost of full 3-D finite element analysis, some authors suggest negligible heat flow in the longitudinal direction; i.e. $\partial T / \partial z = 0$. Hence, heat flow is restricted to an $x - y$ plane, usually positioned at $z = 0$. This has been shown to cause little error except for low speed high heat input welds [10]. The disc moves along the surface of the workpiece in the z -direction and deposits heat on the reference plane as it crosses. The heat then diffuses outward ($x - y$ direction) until the weld cools.

3.1.3 The Hemispherical Heat Source Model

For welding situations, where the effective depth of penetration is small, the surface heat source model of Pavelic, Friedman and Krutz has been quite successful. How-

ever, for high power density sources such as the laser or electron beam, it ignores the digging action of the arc that transports heat well below the surface. In such cases a hemispherical Gaussian distribution of power density (W/m^3) would be a step toward a more realistic model. The power density distribution for a hemispherical volume source can be written as:

$$q(x, y, \xi) = \frac{6\sqrt{3}Q}{c^3\pi\sqrt{\pi}} e^{-3x^2/c^2} e^{-3y^2/c^2} e^{-3\xi^2/c^2} \quad (3.5)$$

where $q(x, y, \xi)$ is the power density (W/m^3). Eqn. 3.5 is a special case of the more general ellipsoidal formulation presented in the next section.

Though the hemispherical heat source is expected to model an arc better than a disc source, it, too, has limitations. The molten pool in many welds is often far from spherical. Also, a hemispherical source is not appropriate for welds that are not spherically symmetric such as strip electrode, deep penetration electron beam, or laser beam welds. In order to remove these constraints, an ellipsoidal heat source suggested by Goldak *et. al.* [2] is discussed next.

3.1.4 The Ellipsoidal Heat Source Model

The Gaussian distribution of the power density in an ellipsoid with center at (0,0,0) and semi-axes a, b, c parallel to coordinate axes x, y, ξ can be written as:

$$q(x, y, \xi) = q(0) e^{-Ax^2} e^{-By^2} e^{-C\xi^2} \quad (3.6)$$

where $q(0)$ is the maximum value of the power density at the center of the ellipsoid. Conservation of energy requires that:

$$2Q = 2\eta VI = 8 \int_0^\infty \int_0^\infty \int_0^\infty q(0) e^{-Ax^2} e^{-By^2} e^{-C\xi^2} dx dy d\xi. \quad (3.7)$$

Evaluating eqn. 3.7 produces the following:

$$2Q = \frac{q(0)\pi\sqrt{\pi}}{\sqrt{ABC}} \quad (3.8)$$

$$q(0) = \frac{2Q \sqrt{ABC}}{\pi \sqrt{\pi}} . \quad (3.9)$$

To evaluate the constants A, B, C , the semi-axes of the ellipsoid a, b, c in the directions x, y, ξ are defined such that the power density falls to $0.05 q(0)$ at the surface of the ellipsoid. In the x direction:

$$q(a, 0, 0) = q(0) e^{-Aa^2} = 0.05q(0) . \quad (3.10)$$

Hence $A = \ln 20/a^2 \approx 3/a^2$.

Similarly, $B \approx 3/b^2$ and $C \approx 3/c^2$. Substituting these values of A, B, C and $q(0)$ from eqn. 3.9 it can be shown that:

$$q(x, y, \xi) = \frac{6\sqrt{3}Q}{abc \pi \sqrt{\pi}} e^{-3x^2/a^2} e^{-3y^2/b^2} e^{-3\xi^2/c^2} . \quad (3.11)$$

The coordinate transformation ξ can be substituted into eqn. 3.11 to provide an expression for the ellipsoid in the fixed coordinate system.

$$q(x, y, z, t) = \frac{6\sqrt{3}Q}{abc \pi \sqrt{\pi}} e^{-3x^2/a^2} e^{-3y^2/b^2} e^{-3[z+U(\tau-t)]^2/c^2} . \quad (3.12)$$

If the heat flow in the z -direction is neglected, an analysis can be performed on the $x-y$ plane located at $z = 0$ which is similar to the disc source. Where the ellipsoidal source intersects this plane the power density is calculated for each time increment.

3.1.5 The Double Ellipsoid Heat Source Model

A double ellipsoid heat source model was proposed by Goldak *et. al.* [2]. Refer to Fig. 2.13 (a) for dimensions. Calculations with the ellipsoidal heat source model revealed that the temperature gradient in the front was not as steep as expected and the gentler gradient at the trailing edge of the molten pool was steeper than experimental experience [2]. To overcome this limitation, two ellipsoidal sources were placed back and back as shown in Fig. 2.13 (a). The front half of the source is the quadrant of one ellipsoidal source while the rear half is the quadrant of another

ellipsoid. The power density distribution along the ξ axis is shown in Fig. 2.13 (a). In this model, the fractions f_f and f_r of the heat deposited in the front and rear quadrants are needed such that $f_f + f_r = 2$. The power density distribution inside the front quadrant becomes:

$$q(x, y, z, t) = \frac{6\sqrt{3}f_f Q}{abc \pi \sqrt{\pi}} e^{-3x^2/a^2} e^{-3y^2/b^2} e^{-3[z+U(\tau-t)]^2/c^2} \quad (3.13)$$

and that for the rear quadrant becomes:

$$q(x, y, z, t) = \frac{6\sqrt{3}f_r Q}{abc \pi \sqrt{\pi}} e^{-3x^2/a^2} e^{-3y^2/b^2} e^{-3[z+U(\tau-t)]^2/c^2} \quad (3.14)$$

In eqns. 3.13 and 3.14, the parameters a, b, c can have different values in the front and rear quadrants since they are independent. Indeed in welding dissimilar metals it may be necessary to use four octants each with independent values for a, b and c . The shape of the double ellipsoid approximates the tear-drop profile of the molten pool. This is said to represent the combined effects of the heat source flux distribution, arc digging and weld pool stirring which would also decay in the same fashion. At the same time a high thermal conductivity ($\approx 120 \text{ W/m}^\circ \text{C}$) is applied to the weld pool to ensure rapid heat transfer in this region. In practise, the size of the double ellipsoid is selected about 10 % smaller than the weld pool dimensions. Melt-out brings the weld pool to the right size. If the double ellipsoid size is too small, unreasonably high temperatures are computed, indicating that the dimensions are unrealistic. On the other hand, if the size is too large, the weld pool temperature may not even exceed the melting point temperature which again is unrealistic. To determine the appropriate size of the ellipsoid, the weld pool dimensions can be estimated from the non-dimensional approach of Christensen *et. al.* [11].

One advantage of this model is that the assumption of temperature independent thermal properties as in [1] is removed. The volumetric specific heat ρC and thermal conductivity k , are assumed to vary as shown in the Fig. 3.4 and Fig. 3.5 respectively. The latent heats of fusion and transformation are incorporated as shown in the

Figure 3.4: Volumetric Specific Heat and Latent Heat Variation with Temperature [5].

Figure 3.5: Thermal Conductivity Variation with Temperature [5].

Fig. 3.4. These are the arrangements for the finite element analysis implementation that is discussed in detail in [2] and [12].

3.2 The Li-Easterling-Ashby Model

In this section, we look into phase transformation equations discussed by Li *et. al.* in [4] on the dissolution of cementite in spheroidized hyper-eutectoid steels and the subsequent homogenization of austenite (such steels contain a dispersion of globular cementite particles in a low-carbon, ferrite matrix). This is, in fact, an extension of what they had developed for modeling the dissolution of pearlite in hypo-eutectoid steels [13]. Here is a model that describes comprehensively the kinetics of cementite dissolution and carbon redistribution in austenite.

3.2.1 Precipitate Dissolution

During the welding of steels, when the local temperature exceeds the A_1 value, austenite begins to nucleate at the ferrite-cementite interface. This austenitic shell grows by continued dissolution of cementite. We assume that the heat treatment is isothermal; as cementite dissolution comes to a completion, the initially non-uniform carbon distribution undergoes homogenization. This process of homogenization tends to increase as higher peak temperatures are reached and slow thermal cycles. As the cooling cycle begins, the austenite now rich in carbon, transforms to martensite and other daughter products. The extent of cementite dissolution and carbon homogenization determines the amount of martensite.

To determine the distribution of solute around the shrinking particle, there are three principal approximations namely, (a) the invariant field approximation, (b) the invariant size approximation, (c) the linearized gradient approximation. Aaron *et. al.* [14] have discussed these methods at length and concluded that the invariant size approximation was most satisfactory; Li *et. al.* [4] had adopted the same for

Figure 3.6: Concentration Variation as per Invariant Size Approximation [13].

their analysis. As per the invariant size approximation, the flux of solute from a precipitate particle of instantaneous radius $R(t)$ is the same as that which would exist if the precipitate-matrix interface had been fixed at R , right from the start, ignoring the motion of the interface on the diffusion field. Refer Fig. 3.6.

There were a few assumptions to simplify the mathematics namely,

1. carbon diffusion in austenite is the rate controlling process,
2. equilibrium exists at the $\alpha - Fe_3C$ interface so that the local carbon concentrations are given by the phase diagrams (refer Fig. 3.7),
3. cementite particles are spherical, of constant size and equally spaced so that the matrix can be imagined to be divided into identical cells with an Fe_3C particle at the center (refer Fig. 3.8), and
4. there is no interaction in the diffusion fields between adjacent cells.

To obtain a function describing the carbon concentration variation with distance from the center of an Fe_3C particle, we need to solve the Fick's second law:

$$D\nabla^2 C = \frac{\partial C}{\partial t} \quad (3.15)$$

subject to the boundary conditions that $C = C_0$ at $r = r_e$ and $C = C_i^c$ at $r = R(t)$. C_0 is the carbon concentration in ferrite before welding, C_i^c the carbon content

Figure 3.7: Iron Rich Lower Section of the $Fe - C$ Equilibrium Diagram.

Figure 3.8: Cell Model for Analyzing Fe_3C Dissolution [4].

of austenite in equilibrium with cementite (refer Fig. 2.8), r_e the radius of the equivalent sphere (refer Fig. 3.8), $R(t)$ the instantaneous radius of the cementite particle and r the distance from the center of the Fe_3C particle to the point of observation.

If the particles are large or the observation point close to the $\alpha - Fe_3C$ interface, one can rewrite the above equation as:

$$D \frac{\partial^2 C}{\partial x^2} = \frac{\partial C}{\partial t} \quad (3.16)$$

subject to the boundary conditions that $C = C_0$ at $x = r_e$ and $C = C_\gamma^e$ at $x = R(t)$. To solve this, a moving coordinate system is used, such that $x = \zeta + R(t)$; the solute concentration distribution is referred to coordinates with origins at the $\alpha - Fe_3C$ interface. An important assumption is made here that a quasi steady-state carbon concentration distribution exists in the moving coordinate system - an observer sitting on the interface observes a time-invariant carbon concentration distribution. Note that this method differs from the invariant field approximation which assumes a steady-state carbon distribution in the fixed coordinate system.

Fick's second law can be rewritten for the ζ -coordinate system as follows:

$$D \frac{\partial^2 C}{\partial \zeta^2} + V \frac{\partial C}{\partial \zeta} = \frac{\partial C}{\partial t} \quad (3.17)$$

subject to the boundary conditions that $C = C_0$ at $\zeta = r_e - R(t)$ and $C = C_\gamma^e$ at $\zeta = 0$. Assuming quasi-stationary carbon distribution we can modify the above equation:

$$\frac{\partial^2 C}{\partial \zeta^2} + \frac{V}{D} \frac{\partial C}{\partial \zeta} = 0. \quad (3.18)$$

Eqn. 3.18 has the solution

$$C = B_1 \exp -\frac{V}{D} \zeta + B_2 \quad (3.19)$$

where B_1 and B_2 are constants evaluated from the boundary conditions as:

$$B_1 = \frac{C_\gamma^e - C_0}{1 - \exp\{-\frac{V}{D}[r_e - R(t)]\}}; \quad (3.20)$$

$$B_2 = C_\gamma^c - B_1 . \quad (3.21)$$

The boundary of the region in which the carbon concentration exceeds a critical concentration, C_c ($C > C_c$) is then obtained by setting $C = C_c$ in eqn. 3.19 and putting $x = \zeta + R(t)$ we have

$$x = \frac{D}{V} \ln \frac{B_1}{C_c - B_2} + R(t) \quad (3.22)$$

The instantaneous radius of the cementite particle as evaluated in [4] is

$$R(t) = r_e - \{2Dt \ln(1 + K) + (r_e - r_i)^2\}^{1/2} \quad (3.23)$$

$$V = \frac{dR(t)}{dt} = \frac{-D \ln(1 + K)}{\{2Dt \ln(1 + K) + (r_e - r_i)^2\}^{1/2}} \quad (3.24)$$

where r_i is the initial radius of the cementite particles and K , the supersaturation parameter given by

$$K = \frac{C_\gamma^c - C_0}{C_c^\gamma - C_\gamma^c} , \quad (3.25)$$

C_c^γ being the carbon content in cementite in equilibrium with austenite (see Fig. 8).

From eqns. 3.22, 3.23 and 3.24, we have for the constants

$$B_1 = C_\gamma^c - C_c^\gamma ; B_2 = C_c^\gamma . \quad (3.26)$$

Thus eqns. 3.19, 3.25 and 3.26 give a complete description of the kinetics of cementite dissolution in isothermal conditions.

3.2.2 Homogenization of Austenite

When the carbide particles have just dissolved, the carbon distribution in austenite is still very non-uniform. The concentration gradients provide a force to further redistribute the carbon in austenite, though the extent to which homogenization can occur is dependent on the thermal cycle. At the instant when cementite dissolution

is completed and carbon homogenization commences, i.e. when $R(t)$ is 0 from eqn. 3.23, we get

$$Dt = \frac{r_e^2 - (r_e - r_i)^2}{2 \ln(1 + K)}. \quad (3.27)$$

This is used as a criterion for the initiation of carbon homogenization and a function $C(x, t)$ is sought describing the subsequent variation in carbon concentration during homogenization. Setting $R(t) = 0$, in eqn. 3.19 gives the initial carbon distribution as

$$C(x) = B_1 \exp\left(-\frac{V_c}{D}x\right) + B_2, \quad x \geq 0 \quad (3.28)$$

$$C(x) = B_1 \exp\left(\frac{V_c}{D}x\right) + B_2, \quad x < 0 \quad (3.29)$$

where V_c is the terminal velocity of the α - Fe_3C interface at the moment of complete cementite dissolution. From the eqn. 3.24 we can show that

$$V_c = \frac{-D \ln(1 + K)}{r_e}. \quad (3.30)$$

The appropriate solution to the diffusional equation which satisfies the initial conditions, (eqns. 3.28 and 3.29) from [14] is:

$$C(x, t) = \frac{1}{\sqrt{4\pi Dt}} \int_0^\infty C(\zeta) \exp\left[-\frac{(x - \zeta)^2}{4Dt}\right] d\zeta \quad (3.31)$$

where $C(\zeta)$ is given by eqns. 3.28 and 3.29. This integral has been evaluated in [4] and the closed form expression is

$$C(x, t) = B_1 \exp\left[-\frac{V_c}{D}x + \left(\frac{V_c}{D}\right)^2 Dt\right] + B_2. \quad (3.32)$$

The boundary of the region within which the carbon content C exceeds some critical value C_c (which is taken to be 0.05 wt%) to transform to martensite is given by setting $C(x, t) = C_c$. This was found to be

$$x = \frac{D}{V_c} \ln\left(\frac{B_1}{C_c - B_2}\right) + V_c t. \quad (3.33)$$

We should note that the starting point of time in this equation is the instant when cementite dissolution is just completed.

3.3 The Watt-Coon Algorithm

The Watt-Coon algorithm [5] was developed for predicting the microstructure in the weld heat affected zone. This section deals with the kinetics of austenite decomposition as discussed in [5]. Austenite decomposition into its daughter products ferrite, pearlite, bainite and martensite, is determined by Kirkaldy's decomposition kinetics. Ordinary differential equations simulating these reactions are discussed in [15, 16, 17]. We present the some important points from the Watt-Coon algorithm in the following subsection.

3.3.1 Austenite Decomposition

Kirkaldy [15, 16, 17] developed a set of equations which model the austenite decomposition reactions for each one of the diffusional products of austenite. These equations assume that a single continuous function can describe both the nucleation and the subsequent growth of the daughter phases. For each reaction, the general reaction rate is characterized as

$$\frac{dX}{dt} = B(G, T)X^m(1 - X)^p \quad (3.34)$$

where X is the volume fraction of the daughter phase, B an effective rate coefficient, G the austenite grain size, and m and p being semi-empirical coefficients set to less than 1 to ensure convergence in a form that is derived from a point nucleation and impingement growth model [18]. The rate coefficient, B includes the effect of grain size on the density of eligible nucleation sites. It also includes the amount of austenite supercooling and the effect of alloying elements and temperature on diffusion.

Consider the austenite decomposition to ferrite; austenite is stable above the A_3 line. As the temperature drops below the A_3 value, ferrite begins to form by nucleating at the austenite grain boundaries, subsequently growing into the austenite

grains themselves. Thus, the volume fraction of ferrite formed is a function of the nucleation rate and the product of the area of the $\alpha - \gamma$ interface and its rate of advance. The last two factors in the general rate equation take into account the changing area of the $\alpha - \gamma$ interface. The nucleation rate is primarily a function of the amount of supercooling below the A_3 line, and of the austenite grain size. A given reaction temperature affects both the diffusion rate of carbon and the degree of supercooling. For a given temperature, the degree of supercooling varies with the alloy content because of the alloying elements' effect on the A_3 line. The alloys tend to be either austenite or ferrite stabilizers, lowering or raising the equilibrium line respectively. Alloying additions can further inhibit transformations if the reaction requires their partitioning by phase boundary diffusion. Thus molybdenum, although a ferrite stabilizer, slows reactions through its diffusional inhibition of the transformations.

Kirkaldy *et. al.* [15] prescribe the austenite to ferrite reaction rate to be

$$\frac{dX}{dt} = \frac{2^{(G-1)/2} \Delta T^3 X^{2(1-X)/3} (1-X)^{2X/3}}{\text{COM} \exp(-23,000/RT)} \quad (3.35)$$

where $\text{COM} = 59.6 \% \text{ Mn} + 1.45 \% \text{ Ni} + 67.7 \% \text{ Cr} + 2.44 \% \text{ Mo}$; $\frac{dX}{dt}$ in units of volume fraction per second; G the ASTM grain size number for austenite; ΔT being $A_3 - T$, the amount of supercooling and X the existing daughter product volume fraction, as defined below.

The denominator takes into account the diffusional resistance of the reaction and is developed from the general form

$$\frac{1}{D_{eff}} = \frac{1}{D_c} + \sum_{i=1}^n \frac{k_i c_i}{D_i} \quad (3.36)$$

where D_{eff} is the effective diffusion coefficient calculated as the diffusion coefficient for carbon D_c modified by the presence of concentrations C_i of other alloying elements.

To get a sense of the significance of each of these terms, we consider the changing of the grain size from 8 to 2, which would decrease the reaction rate by a factor of 8. Combining the interactive effects of composition and temperature, the value of the rate coefficient B from eqn. 3.34 reaches a maximum of about 0.65 /sec for 4140 steel at a temperature of about 865° K. For a highly weldable, high Mn, low C pipeline steel, the maximum B is about 1.45 /sec at 900° K. As the temperature decreases, the supercooling increases the rate as ΔT^3 but the diffusion rate slows as $\exp(-23,000/RT)$. The former term demonstrates that it is important to calculate A_3 accurately, particularly for slow cooling rates.

It is important to use the precise definition of X as given by Kirkaldy for eqn. 3.35. If one simply defines X as the volume fraction of ferrite then it is possible, using the kinetic equation 3.35 alone, to form more than the equilibrium amount of ferrite in the intercritical temperature region. This and the relative change in the thermodynamic driving force for the reaction are corrected by proposing a phantom reaction which can go to completion. The phantom phase product is $X_F = X X_{FE}$ where X_{FE} is the equilibrium fraction of ferrite given by the lever law at that temperature. Thus in eqn. 3.35 we set $X = X_F/X_{FE}$. For temperatures below the eutectoid and at times before the substantial production of a second daughter product, the eutectoid composition is used to find X_{FE} .

For pearlite formation, Kirkaldy prescribes a similar equation

$$\frac{dX}{dt} = \frac{2^{(G-1)/2} \Delta T D_{eff} X^{2(1-X)/3} (1-X)^{2X/3}}{1.79 + 5.42 (\%Cr + \%Mo + 4\%Mo\%Ni)} \quad (3.37)$$

where D_{eff} is evaluated as

$$D_{eff} = \frac{1}{\exp(-27,500/RT)} + \frac{0.01\%C + 0.52\%Mo}{\exp(-37,000/RT)} \quad (3.38)$$

and ΔT , the pearlite undercooling being $(PS - T)$, where PS is the pearlite-start temperature.

If some of the austenite had previously transformed to ferrite at the onset of pearlite formation, then the amount of unreacted austenite will be $(1 - X_F)$. As the pearlite forms, then the appropriate value for eqn. 3.38 is $X = X_P/(1 - X_{FE})$.

When the bainite-start temperature is reached, it is assumed that pearlite growth degenerates to bainite formation. For the bainitic reaction, we use

$$\frac{dX}{dt} = \frac{2^{(G-1)/2} (\Delta T)^2 \exp(-27,000/RT) X^{2(1-X)/3} (1-X)^{2X/3}}{(2.34 + 10.1\%C + 3.8\%Cr + 19\%Mo)10^{-4}f(X, C_i)} \quad (3.39)$$

where

$$f(X, C_i) = \exp[X^2(1.9\%C + 2.5\%Mn + 0.9\%N + 1.7\%Cr + 4\%Mo - 2.6)] \quad (3.40)$$

and ΔT , the metastable bainite undercooling being $BS - T$, where BS is the bainite-start temperature. It must be noted that $f(X, C_i)$ is set to 1 for all negative values of the exponent.

If pearlite exists when the bainite-start temperature is crossed, then it is assumed that the existing pearlite-austenite interface continues to transform, but now yielding bainite. Then X in eqn. 3.40 is originally X_P . If no pearlite exists, then because the bainite reaction is sluggish, X is simply the fraction of bainite formed and it continues to transform until the remaining austenite is consumed or the martensite-start temperature is reached.

It is assumed that the austenite which has not yet transformed at martensite-start temperature will become martensite on further cooling. To simulate the transformation of austenite to martensite a kinetic reaction discussed by Koistinen and Marburger [19] is applied.

It would be a simple matter to develop an equation for the retained austenite as a function of carbon content. However, this constituent becomes a significant volume fraction only at medium to high carbon contents where the martensite formed is brittle. For the purposes of weld heat affected zone, the objective is to find the

conditions that will avoid the development of brittle regions. Thus, the presence of retained austenite is usually not regarded as critical.

Bibliography

- [1] Rosenthal, D., *Transactions of the ASME* Vol. 68, page 849, 1946.
- [2] Goldak, J.A., Chakravarti, A., Bibby, M.J., *Metallurgical Transactions B* . AIME/ASM, Vol. 15 B, page 299, 1984.
- [3] Pavelic, V., Tanbakuchi, R., Uyehara, O.A., Myers, P.S., *Welding Journal Research Supplement* Vol. 48, pages 295s - 305s, 1969.
- [4] Li, W-B., Easterling, K.E., Ashby, M.F., *Acta Metallurgica* Vol. 34, No. 8, pages 1533-1543, 1986.
- [5] Watt., D.F., Coon, L., Bibby, M., Goldak, J., Henwood, C., *Acta Metallurgica* Vol. 36, No. 11, pages 3029-3035, 1988.
- [6] Kou, Sindo , *Welding Metallurgy*, John Wiley & Sons Inc., 1989.
- [7] Rykalin, N.N., *Energy Resources for Welding*, Houdremont Lecture, IIW, London, pages 1 - 23, 1974.
- [8] Friedman, E., *Journal Pressure Vessel Technology, Transactions of the ASME* Vol. 97, pages 206 - 13, 1975.
- [9] Krutz, G.W., Segerlind, L.J., *Welding Journal Research Supplement* Vol. 57, pages 211s - 16s, 1965.
- [10] Andersson, B.A.B., *Journal of Engineering Materials and Technology, Transactions of the ASME*, Vol. 100, pages 356 - 62, 1978.

- [11] Christensen, N., V de L Davies, Gjermundsen, K., *British Welding Journal* Vol. 7, pages 54-75, 1965.
- [12] Goldak, J.A., Patel, B., Bibby, M.J., Moore, J., Computational Weld Mechanics, *AGARD presentation - Structures and Materials* 61st Panel Meeting, Oberammergau, Germany, September 8 -13, 1985.
- [13] Ashby, M.F., Easterling, K.E., *Acta Metallurgica* Vol. 32, page 1935, 1984.
- [14] Aaron, H.B., Fainstein, D., Kotler, G.R., *Journal of Applied Physics* Vol. 41, page 4404, 1970.
- [15] Kirkaldy, J.S., Venugopalan, D., *American Institute of Mining Engineers*, Phase Transformations in Ferrous Alloys, pages 125 - 148, Philadelphia, 1984.
- [16] Kirkaldy, J.S., *Metallurgical Transactions* Vol. 4, page 2327, 1973.
- [17] Kirkaldy, J.S., Sharma, R.C., *Scripta Metallurgica* Vol. 16 page 1193, 1982.
- [18] Cahn, J.W., *Acta metallurgica* Vol. 4, page 572, 1956.
- [19] Koistinen, D.P., Marburger, R.E., *Acta Metallurgica* Vol. 7, page 59, 1959.

Appendix A

A.1 On Routine Design

Mechanical design to a large extent is routine. Routine design is limited in its creativity. An engineer produces an artifact with what is available to him. Those parts of the design that are routine and methodological can be formalized. The engineer is not permitted to change routine design procedures but can vary certain parameters and observe the effect of these parameters on the magnitudes of others.

The elements that make up a routine design process can be defined as a set of tokens, rules and procedures. Hence, routine design can be reduced to a problem in which these elements communicate with one another. We illustrate these elements - *tokens*, *rules* and *procedures* - with an example from real life. Considering the game of chess as a routine design problem, it can be said to be composed of a finite set of tokens namely the pieces, the board and a set of rules to play the game. Central to this idea is the belief that despite differences in procedures, technology and context, a common basis can be provided for routine design.

Object-oriented environments offer several methodologies and mechanisms such as polymorphism and encapsulation [1] which appeal to the representation of routine design as a preferable choice. Prototype-based systems and class-based systems are two primary paradigms currently available. Prototypical systems support multiple inheritance based on functional and structural metrics while class-based systems represent the philosophical counter idea - *primitives* - through set theory. Specification and description are two distinct and important stages in routine design. Specification is the construction of a formal environment or a "design culture" as Navinchandra [2] describes it. It is composed of libraries for parts, rules, procedures and heuristics. The designer works with them and cannot make changes to the

environment. Description is a configuration stage where the designer describes a problem within the boundaries of a specification.

A.2 Specification and Description

The view of design as a set of communicating agents leads naturally to a representation of the design process as a graph. The nodes represent partitions or *chunks* in space and time and the edges representing shared variables or data between them. Before describing a chunk, we clarify the notion of a token [3].

In automata and formal language theory, standard tokens are sometimes referred to as the alphabet, a set of standard parts, things or objects. Aristotle was the first to formalize this idea in his "primitives" based view of the universe [4]. In attempting to describe the universe as a hierarchical combination of a few primitives, he tried categorizing it into a finite set of basic primitives which cannot be decomposed further. A basic primitive, which cannot be expressed in terms of other tokens, is called a primitive token or simply a token.

Chunking is a concept that was borrowed from psychology. Simon [5] discusses chunks and the limits of human information processing, including short and long term memory in his chapters about remembering and learning. Chunks are collections of tokens based on maximal familiarities and semantic bonds. A chunk encapsulates some form of knowledge represented through a set of tokens and their relations. In other words, a chunk is a *collapsed subgraph* consisting of tokens and declarative constraints. This can be recursively structured; a chunk can encapsulate other chunks when partitioning complex domains. Chunks appear in different environments in many forms. Higher level programming languages support forms of chunks via record structures and accompanying procedures, objects, etc. Class hierarchies, inheritance mechanisms and many other features of modern programming practises allow chunking.

Specification in this formalism is: (a) specifying templates for nodes and (b) specifying an environment in which these parts can be assembled according to specific rules. In the case of a hierarchic part, this template describes a set of macroscopic parameters and a mapping between these parameters and a lower level graph which describes the behavior at a microscopic level. A template is a parameterization of a node along with mappings between parameters. Parameters can be classified as internal and external. This provides a natural information hiding mechanism. Similarly, relations can be divided as internal and external. An environment is a workbench where one describes specific problems. It is composed of a library of node templates and a set of rules to create topologically valid assemblies. Rules may be in the form of allowable constraints among parts.

Description takes place in one or several workbenches, as in the case of hierarchic designs. One creates virtual objects from a library of templates for nodes and defines relations between them according to some rules of assembly which results in a particular statement of a design case.

A.3 The Representation

We have attempted to model the domain of welding design as a graph, the nodes representing various centers of activity, i.e., the tokens and/or chunks, and the edges representing the various relationships between them. This representation has been inspired by the way a designer organizes his/her thoughts. We present here the welding domain representation that was adopted in this thesis.

The domain of welding has been identified as consisting of the following:

- a representation for the geometry of the welded structure,

- a representation for the base metal,
- a representation for the welding process, and
- a representation for information handling and file reformatting to automatically produce an input file for the finite element solver.

These representations form a block diagram; these blocks can be seen as distinct bodies of knowledge. At the moment, we assume that we do not know the inner details of these black boxes. These are concepts being represented as chunks. In trying to explore the interaction between these concepts, one can see that the geometry of the welded structure depends on the base metal composition since its shape and attributes are prescribed by the designer after considering service conditions. Base metal selection criteria could alter the geometry of the structure because of the obviously differing mechanical properties of different metals. One point is to be noted that changing the base metal would not change the geometry from a t-joint to a k-joint, but might perhaps change the diameters of the pipes in a t-joint. This interaction between the chunks *Welding Geometry* and *Base Metal* can be called as the **composition-geometry type** relation, since base metal composition drives geometry (and vice versa).

In exploring the relation between the chunks *Base Metal* and *Welding Process*, one can see that base metal selection also influences welding process selection in either restricting the choice as with non-ferrous metals or giving a wide range of applicable processes as with steels. Since we have restricted ourselves to steel as base metal and the available processes to the four processes mentioned in Sec. 2.5, this interaction between the chunks *Base Metal* and *Welding Process* can be called the **steel-arc welding type** relation; in a broader scope it might have been called the **base metal-arc welding type** relation.

For the interaction between the chunks *Welding Geometry* and *Welding Process*, one can see that for the geometries currently available in the environment, arc welding can be performed using any of the four processes mentioned in Sec. 2.5. The selection of a process is purely application specific. This relation can be called the **geometry-arc welding type** relation. These chunks whose interactions have been established so far contribute to the physics of the welding process. *Base metal* contributes to the welding physics simulation through microstructural transformations that take place during welding. *Welding Process* provides the heat input to effect metal joining. *Welding Geometry* is the result of the applying the welding process. Here the contribution is from *Welding Physics Simulation* in the form of transient temperature history and microstructure contours.

These interactions take place at a high level; the designer should bear in mind that the inner details of these chunks are unknown. This level of abstraction is called as the *Weld World Abstraction Level*. This is a graph whose nodes are chunks embodying knowledge and whose edges provide an explanation for structuring knowledge this way. See Fig. A.1.

A.3.1 Welding Geometry

The chunk *Welding Geometry* encapsulates knowledge about object geometries and their finite element meshes. No detail about geometry is represented here. The object geometry can either be an individual part or an assembly. This is represented as a chunk *Part-Piping Assembly*, as we wish to be able to model not only individual part geometries but also their assemblies. The other representation is *Finite Element Mesh*, a token that stores as its state the finite element mesh of the geometry. The interaction between *Part-Piping Assembly* and *Finite Element*

Figure A.1: Weld World Abstraction Level.

Mesh is a mesh that represents the geometry in 3-space. This is a relation of the part/assembly-mesh type. These two entities help represent the geometry form a graph that lies encapsulated in the chunk *Welding Geometry*. This is an abstraction one step inside the chunk *Welding Geometry* in the *Weld World Abstraction Level*. We call this as the *Part/Piping Assembly Abstraction Level*. From one level higher, the chunk *Welding Geometry* can be seen as a collapsed subgraph consisting of an abstract geometry and its finite element mesh. We say abstract since we assume we do not know the geometry of the primitives available in the environment. See Fig. A.2.

The next step is exploring the representation of the chunk *Part-Piping Assembly*. Here the designer adds the primitives of his/her choice. We have in this environment represented three object geometries of the t-joint, girth weld and butt weld; these are tokens placed in this abstraction called the *Primitives Abstraction Level*. See Fig. A.3. The interaction between these tokens can now be explored. The interaction

Figure A.2: Part/Piping Assembly Abstraction Level.

Figure A.3: Primitives Abstraction Level.

between *T-Joint* and *Girth Weld* is obvious in that they are composed of sections of pipes and hence can be assembled; this relation is called the **pipe sections type**. The token *Butt Weld* is for modeling plate-welding. Hence there is nothing common in the geometry with the other two. Thus, the *Primitives Abstraction Level* is a graph with two connected components.

One can observe that moving down the hierarchy of chunks inside chunks as collapsed subgraphs, the leaves are standard tokens such as *Butt Weld*, *T-Joint*, *Girth Weld* and *Finite Element Mesh*.

A.3.2 Base Metal

The chunk *Base Metal* in the *Weld World Abstraction Level* should encapsulate knowledge about (i) base metals available for selection and (ii) their compositions. The representation for base metal selection should support a hierarchic structure and the result of selection should be represented against composition. These are at an abstraction called the *Steel Selection Abstraction Level*. See Fig. A.4. The interaction between the chunk *Taxonomy Of Steels* and the token *Composition* is called the **percentage composition type**.

The taxonomy of steels is divided into groups of chunks and/or tokens placed at abstraction levels as shown in Fig. A.5. Once the user marches down the hierarchy and makes his selection, the composition of that selection is stored in the token *Composition*. It should be noted that the tokens and chunks in the base metal selection hierarchy are disjoint. The user can select only one base metal for an assembly. Currently, there is no means to simulate the welding of an assembly of parts of differing compositions.

Figure A.4: Steel Selection Abstraction Level.

A.3.3 Welding Process

The chunk *Welding Process* in the *Weld World Abstraction Level* encapsulates knowledge (see Fig. A.1) about arc welding process selection as a hierarchy and process parameter specification. The result of welding process selection fixes the parameters to be specified. Hence the relation between the chunk representing arc welding process selection *Process Selection* and the token *Process Parameters* can be said to be **process choice type**. These entities exist in an abstraction called the *Process Specification Abstraction Level*. See Fig. A.6. The chunk *Process Selection* consists of tokens representing the four processes discussed in Sec. 2.5 are placed in an abstraction called the *Arc Welding Process Abstraction Level* as shown in Fig. A.7. It is remarked here that this subgraph inside the chunk *Process Selection* has no edges. This means that only one process can be selected by the user to model an assembly consisting of several parts; this is one of the current limitations of this environment.

A.3.4 Input File Writer

The representation for information handling and file reformatting activities is a chunk called the *Input File Writer*; it handles the maze of information coming from the structured mesh code at various stages and organizes the output in the right

Figure A.5: Hierarchy Of Steel Selection.

Figure A.6: Process Specification Abstraction Level.

Figure A.7: Arc Welding Process Abstraction Level.

sequence for the finite element solver. Fig. A.8 illustrates the data driven data flow that takes place in this chunk.

Once the job specification is completed, a sequence of operations are triggered as the user clicks on the "Solve:" menu button in the Weld Analyzer interface. As mentioned earlier in the introduction, this environment integrates the following software components namely

1. the user interface manipulated through Smalltalk-80 [1],
2. constraint satisfaction for piping assemblies involving girth welds and t-joints implemented through declarative constraints [6],
3. finite element meshes produced in INGRID [7],
4. finite element solver for heat transfer and microstructural analysis in FORTRAN 77, and
5. structured mesh manipulation code written in C.

Constraint satisfaction is used when creating assemblies with piping primitives; this ensures uniformity in geometry. Once geometry satisfaction is achieved, constraints play no role in the input file preparation. We briefly describe the stages in manipulating meshes; these steps seem simple but are not as simple as they appear.

Figure A.8: Input File Writer.

1. The user signals completion of job specification by clicking on the "Solver" menu button. The next step is the creation of the finite element mesh script. The output at this stage is the "weld_mesh" file that contains the source code in INGRID to produce a mesh for the single part or assembly. This corresponds to data-point D1 in Fig. A.8.
2. The coarse mesh created to capture the geometry is not shown to the user on the screen. The output at this stage is a file containing the list of elements, the topology and a listing of the coordinates of all the nodes, followed by a list of nodes lying on the weld path. Element topology for an eight node brick is reported as a list of eight nodes, the first four nodes that make up one face and the next four nodes that make up the other parallel face. This corresponds to data-points D5 and D7 in Fig. A.8.
3. Nodes lying along the weld path (picked up in step 2) are now used to extract the elements lying along the weld path in this stage. Element numbers are stored in a file called "elements". This corresponds to data-point D6 in Fig. A.8.
4. The next step is to refine the coarse mesh produced by the environment. For doing this, we need the list of elements along which refining will have to be performed (the output of step 3). In one pass of refinement, an element is divided into 8 smaller elements. Refinement can be done twice or thrice if the mesh size is too large. A new file called "weld_remeshed" is written as output, containing the new elements and nodes introduced by remeshing, exactly in the same format as the input file. Refinement becomes necessary since coarse meshes produce too approximate results; this should be done till an acceptably fine mesh is achieved. Ideally, it would be appropriate to have a minimum of 6 elements along the major axis of the weld pool ellipsoid and 6 along the minor axis. This corresponds to data-point D8 in Fig. A.8. Note that till

the previous stage there were no graded elements but after remeshing, graded elements appear in the mesh.

5. Surfaces are extracted using the "weld_remeshed" file as input. Elements lying on the surface of the mesh are identified. These elements with their six faces are stored in a table. Faces that appear twice in the table are ignored; faces appearing only once are picked up and reported as external. The result of this step is a file containing the surface elements as four node quadrilaterals. The output is the "external_surf" file containing information to generate a refined mesh with graded elements. This corresponds to data-point D9 in Fig. A.8.
6. The "external_surf" file becomes the input to the GRPSURFACE software that reads the "external_surf" file and gives as output a binary file "TORR" that becomes the input to the TAURUS database; this piece of software GRPSURFACE is essentially a format translator. This corresponds to data-point D10 in Fig. A.8.
7. In the next step, the user identifies the surfaces one by one as being insulated, being the outside or the inside with the help of the TAURUS database. Every surface picked up in step number (5) is highlighted one after another and the user is asked to enter 1 for inside, 2 for outside or 0 for insulated; this is performed to specify the boundary conditions of convection to atmosphere on the outside and convection to the fluid flowing inside. For pipe welding problems like the t-joints, girth welds and butt welds, the default values for the surfaces are provided and this step can be conveniently skipped. If new configurations are assembled by the user, this step of picking the surfaces has to be performed on a workstation with a color monitor so that the yellow highlighting for the surfaces stands out against the background. This corresponds to data-point D11 in Fig. A.8.

8. Next, we create a file with face elements. The refined mesh is taken as input and elements lying on the surface are extracted, hereafter referred to as the face elements. This face elements list is appended to the "weld_remeshed" file. This corresponds to data-point D12 in Fig. A.8.

9. In the next stage, we search for new nodes introduced by refining the mesh along the path of the heat source. To do this, we loop over the nodes extracted in step 2 (Data Point D7) and get the neighbors of any two successive nodes. We put them into a set and pick the node that is a member of either set. This common node and its nodal coordinates are searched out from the "weld_remeshed" file and stored in a file called "weld_path_nodes". It should be noted that this should be updated every time a refinement is done. If the mesh is refined twice, then the list of nodes along the weld path should be updated to include the new nodes introduced after the first refinement. Currently remeshing is limited to one round only; this is one of the limitations of the environment. With this new list of nodes on the weld path as input, further refinement, if necessary, may be performed and the list of nodes have to be updated after every subsequent refinement. This is data point D13 in Fig. A.8.

10. The next stage is the calculation of timesteps for the finite element solver. We start off with the heat source placed at the start node for the first few seconds. This permits the system to come to equilibrium after which the heat source is moved away. This stage lasts typically for 4 to 6 seconds. Thereafter the heat source is moved so that between one time step and the next, it does not move further than one-half of the major-axis distance of the ellipsoidal heat source.

$$(Time\ required\ to\ complete\ welding.t) * (Velocity\ of\ the\ arc,\ U) = (Total\ distance\ from\ start\ node\ to\ finish\ node)$$

$$t\ is\ the\ time\ required\ to\ move\ over\ n\ nodes\ along\ the\ weld\ path.\ Time\ required$$

to move over a distance equal to the major axis of the ellipsoid is calculated: this is the time step for the remaining $(n-4)$ or $(n-6)$ nodes. At the last node on the weld path the heat source is switched off. The temperature gradient is sharp. To capture the temperature in very short intervals at the beginning of cooling, we take small timesteps and slowly increase to larger values as the exponential decay approaches its asymptote. This is data point D14 in Fig. A.8.

11. The next stage is calculation of weld path normals to be appended to the "ellip" file that contains information about the welding process, the internal fluid and heat source dimensions. These normals are required by the solver to ensure proper orientation of the heat source ellipsoid along the weld path: this calculates the local coordinate systems of the nodes on the surface of the weld. This is data point D15 in Fig. A.8.

Bibliography

- [1] Goldberg, A., Robson, D., *Smalltalk-80 The Language*, Addison-Wesley Publishing Co., 1989.
- [2] Navinchandra, D., *Exploration and Innovation in Design*, Springer-Verlag Publishing Co., 1991.
- [3] Dai, Naci M., *Ph.D. thesis proposal in preparation*, Department of Mechanical Engineering, Carleton University, Ottawa.
- [4] Sowa, John F., *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley Publishing Co., 1983.
- [5] Simon, Herbert A., *The Sciences of the Artificial*, The M.I.T. Press, Cambridge, Massachusetts, 1981.
- [6] Borning, A.H., *ThingLab - A Constraint-Oriented Simulation Laboratory*, Ph.D. dissertation, Department of Computer Science, Stanford, March 1979.
- [7] Stillman, D.W., Hallquist, J.O., *INGRID, A Three-dimensional Mesh Generator for Modeling Nonlinear Systems*, Lawrence Livermore National Laboratory, July, 1985.

Appendix B

Here we briefly describe the support software that was used in designing the interface. Constraints are used to assemble primitive objects in the interface when designing welding assemblies. These constraint satisfaction systems are built into Smalltalk-80 and exist within this environment. The potential constraint-satisfaction techniques are briefly explained. Then, a brief introduction to INGRID, the meshing software, is presented. Object-oriented languages and related definitions are introduced before concluding this chapter.

B.1 Declarative Constraint Programming

We introduce some basic definitions in connection with constraints. These definitions are taken from Wm Leler [1], an excellent reference on constraint languages, and a paper by Freeman-Benson *et. al.* [2].

B.1.1 Definitions

Constraints and Constraint Satisfaction

A *constraint* is defined as a desired relationship among one or more objects. A *constraint language* is the language used to describe the objects and the constraints. A *constraint language program* is a program written in a constraint language: this program defines a set of objects and set of constraints on the objects. A *constraint satisfaction system* finds solutions to constraint language programs. The constraint satisfaction system uses problem-solving methods called *constraint-satisfaction techniques*, to find the values of the objects that will make the relationships true.

Constraint Strength can be explained better with ThingLab [3] as an example. In ThingLab, each constraint in a set of constraints is labeled with a *strength*. Of these, the label *required* is special, in the sense that the constraints it labels *must be satisfied*. The remaining strengths all indicate preferences of varying degrees.

In many applications, it is useful to state both the required and preferential constraints. The required constraints must hold; the system should try satisfying the preferential constraints if possible, but no error condition would arise if they are not satisfied. The set of all constraints labeled with their relative strengths is called a *constraint hierarchy*.

A constraint system consists of a set of constraints C and a set of variables V . A constraint can, in other words, be defined as an n -ary relation among a subset of V . Constraints can be *multi-directional*, i.e., a constraint $c = a + b$ might be used to find a value of one of a , b or c . The various *methods* are $a = c - b$, $b = c - a$ and $c = a + b$. Each method may be executed when all its inputs and none of its outputs have been determined by other constraints.

A *solution* to a given hierarchy is a mapping from input variables to output variables. A solution that satisfies all the required constraints in the hierarchy is said to be *admissible*. There may be many admissible solutions; *good* solutions to the hierarchy are those that satisfy the required constraints and also satisfy the preferred constraints such that no other better solution exists. It should be noted that there may be several different good solutions.

These definitions are broad and can be interpreted to include a wide variety of systems, from languages that allow some constraintlike statements, to special-purpose systems that satisfy relations between objects. By constraints, we primarily mean only numeric constraints not logic constraints unless stated to the contrary. Constraint Theory has been extensively discussed in three papers by Friedman and Leonides [4, 5, 6]. The various constraint-satisfaction techniques are explained briefly in the next subsection.

Figure B.1: Temperature Conversion Constraint Graph [1].

B.1.2 Local Propagation

This is the simplest form of satisfying a constraint. To illustrate this, we cite the classical example of temperature conversion from Fahrenheit to Centigrade scales or vice-versa [1]. In this, known values of variables are propagated along the edges of the constraint graph (see Fig. B.1) that pictorially captures the essence of the problem. The formula $F = 9/5 C + 32$ is represented as a constraint graph in which nodes containing unknown objects are shown as square boxes and known variables and operators are shown as circles. Nodes with known values always are ready to “fire” off, i.e. send some data or information to the rest of the constraint graph. When an unknown node receives information enough to simplify the problem to a lesser number of variables, it in turn “fires” off. This triggers off a chain of such events culminating into a final result being calculated at some point in the constraint graph, eventually terminating the chain. For example, let the node C in Fig. B.1 be assigned a value, say 10. The constant node fires 1.8 which combines with 10 fired off from node C . Once the input requirement to the $\times$ node is satisfied, it fires 18. At the $+$ node, there is an adjacent constant node which fires 32. This satisfies the input requirements of the $+$ node which fires off 50. The unknown variable F thus gets assigned 50 which terminates the chain. We should note that a constraint program can work backwards also, as to compute the argument for a required result.

An incremental constraint satisfier has been implemented by Freeman-Benson *et. al.* [2] called the DeltaBlue algorithm, which they say is a variation of local propagation in the sense that it is a fast local propagation algorithm that uses a *locally-predicate-better* comparator. This is specifically useful for user-interactive applications in which a user adds or removes constraints at will. The constraint hierarchy often evolves gradually. The system maintains a "current solution" to the constraints. As constraints are added to or removed from the hierarchy, the system modifies the "current solution" to find a solution to satisfy the new constraint hierarchy. The high-level details of the incremental constraint satisfier can be found in [2]; a precise pseudo-code description can be found in [7].

B.2 Meshing Software - INGRID

Mesheres are generated in INGRID, a three-dimensional mesh generating program [8]. INGRID uses an index space and a progression of control points along x, y and z axes to describe the topology of a model. Surface definitions can then be imposed on the faces of the model and the control points can be pulled or pushed as we wish. The software is at the outset intimidating but is a powerful means of generating some very complex meshes. Interactive graphics in INGRID are patterned after TAURUS, a three-dimensional post processor for FEM analysis [9].

B.2.1 Definitions

Some basic definitions relevant to INGRID, taken from [8] are given below.

An *index space* is a three-dimensional discrete coordinate system with integer values greater than or equal to 1 in each of the three directions. The three discrete coordinates are labeled I, J and K axes respectively. Each point in the index space, (i, j, k) represents a potential nodal point. Elements are defined as groups of adjacent nodes in the index space.

A *region* is any rectangular or cubic block of nodes. A region is usually defined by a block of nodes in an index space.

A *part* is a collection of regions which can be grouped and generated conveniently in an index space. We typically use one region per part while more experienced users group numerous regions together into complex parts.

A complex body can be decomposed into several smaller parts each of which can be created individually and then assembled. A *model* is a collection of parts. Each part has its own index space and is independent of other parts. The user is responsible for ensuring that the nodes and faces at the interfaces between adjacent objects match properly. Once properly assembled, a tolerance can be specified, say 0.001, which means that any number of nodes within a proximity of 0.001 or lesser should collapse into one.

More information on INGRID may be obtained from [8].

B.3 The object-oriented environment

Before discussing the Smalltalk-80 environment, some basic definitions related to object-oriented languages are presented from [10].

B.3.1 Definitions

Object

An object has a set of "operations" and a "state" that remembers the effect of operations. Objects may be contrasted with functions which have no memory. Functional values are completely determined from their arguments, being precisely the same for each invocation.

Class

A class is a template (the cookie-cutter) from which objects (cookies) may be created by "create" or "new" (manufacture) operations. Objects of the same class have the same operations and hence uniform behavior.

Object-based language

A language is object-based if it supports objects as a language feature.

Object-oriented language

An object-based language is object-oriented if its objects belong to classes and class hierarchies may be incrementally defined by an incremental mechanism. That is:

$$\textit{object-oriented} = \textit{objects} + \textit{classes} + \textit{inheritance}.$$

Inheritance

A class may inherit its operations from "superclasses" and may have its operations inherited by "subclasses". An object of the class C created by the operation "C new" has C as its "base class" and may use operations defined in its base class as well as the operations defined in superclasses.

Data Abstraction

A data abstraction is an object whose state is accessible only through messages. The state is generally represented by instance variables. Instance variables are hidden from its clients and are accessible only through the object's messages.

Strong Typing

A language is strongly typed if type compatibility of all expressions representing

values can be determined from the static program representation at compile time.

B.3.2 Smalltalk-80

Smalltalk-80 is an object-oriented environment [11] which has a rich variety of objects in it. Objects support modularity - the behavior of any object does not depend on the internal details of any other object. Classes are the chief mechanism by which Smalltalk-80 reduces complexity; this involves grouping similar objects and organizing them in a hierarchy. User defined classes become a part of the environment. Appropriate choices of objects depend on the purpose for which the object is put and hence one needs to understand the application fully to choose the right object definitions. Many data-structures like dictionaries, sets, bags and ordered collections are readily available; extensive graphics support is also available in Smalltalk-80. Ease of graphics software utility is one distinct advantage by virtue of which this has become the indispensable interface tool for this thesis. The keywords in Smalltalk-80 are: *object*, *message*, *class*, *instance* and *method*. Here are some explanations from [11] for these keywords:

- an object represents a component in the Smalltalk-80 software system; the nature of the object depends on the type of the component it refers.
- a message is a request for an object to carry out one of its operations; there is no specification of how it should be performed.
- a class describes the implementation of a set of objects that all represent the same kind of system component.
- individual objects described by the classes are called instances.
- a method is a message interface specifying how to carry out certain operations.

For details, one may refer to [11, 12, 13].

Bibliography

- [1] Leler, Wm. *Constraint Programming Languages, Their Specification and Generation*, Addison-Wesley Publishing Co., 1988.
- [2] Freeman-Benson, B.N., Maloney, J., Borning, A.H., An Incremental Constraint Solver, *Communications of the ACM* Vol. 33, No. 1, January 1990.
- [3] Borning, A.H., *ThingLab - A Constraint-Oriented Simulation Laboratory*, Ph.D. dissertation, Department of Computer Science, Stanford, March 1979.
- [4] Friedman, G.J., Leonides, C.T., Constraint Theory, Part I: Fundamentals, *IEEE Transactions on Systems Science and Cybernetics* Vol. SSC-5, No. 1, pages 48 - 56, January 1969.
- [5] Friedman, G.J., Leonides, C.T., Constraint Theory, Part II: Model Graphs and Regular Relations, *IEEE Transactions on Systems Science and Cybernetics* Vol. SSC-5, No. 2, pages 132 - 140, April 1969.
- [6] Friedman, G.J., Leonides, C.T., Constraint Theory, Part III: Inequalities and Discrete Relations, *IEEE Transactions on Systems Science and Cybernetics* Vol. SSC-5, No. 3, pages 191 - 199, July 1969.
- [7] Freeman-Benson, B.N., Maloney, J., Borning, A.H., *The Delta Blue Algorithm: An Incremental Constraint Hierarchy Solver* Technical Report 89-08-06, University of Washington, August 1989.

- [8] Stillman, D.W., Hallquist, J.O., *INGRID, a three-dimensional mesh generator for modeling nonlinear systems*, Lawrence Livermore National Laboratory, July, 1985.
- [9] Brown, B.E., Hallquist, J.O., *TAURUS, An Interactive Post-Processor for the Analysis Codes NIKE3D, DYNA3D, TACO3D and GEMINI*, University of California, Lawrence Livermore National Laboratory, Rept. UCID-19392 Rev. 1, 1984.
- [10] Wegner, Peter, Dimensions of object-based language design, *Communications of the ACM. OOPSLA '87 Proceedings*, pages 168 - 182, 1987.
- [11] Goldberg, A., Robson, D., *Smalltalk-80 The Language*, Addison-Wesley Publishing Co., 1989.
- [12] Kaehler, T., Patterson, D., Dave, *A Taste of Smalltalk*, W. W. Norton & Co., N.Y. 10110, 1986.
- [13] Pugh, J.R., Lalonde, W.R., *Inside Smalltalk*, Addison-Wesley Publishing Co., 1990.

2

PM-1 3 1/2" x 4" PHOTOGRAPHIC MICROCOPY TARGET

of/de

2

PM-1 3 1/2" x 4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

Appendix C

```

time = 0.11388E+02
CONTOURS OF TEMPERATURE
min=-0.400E+03 in element 238
max= 0.180E+04 in element 7009

contour values
A= 2.00E+02
B= 3.75E+02
C= 5.50E+02
D= 7.25E+02
E= 9.00E+02
F= 1.08E+03
G= 1.25E+03
H= 1.42E+03
I= 1.60E+03

```

Figure C.1: T Joint Sample Results - Temperature Plots.

```

time = 0.10182E+02
CONTOURS OF TEMPERATURE
min= 0.439E+01 in element 4174
max= 0.180E+04 in element 5069

```

```

contour values
A= 1.55E+02
B= 3.42E+02
C= 5.29E+02
D= 7.15E+02
E= 9.02E+02
F= 1.09E+03
G= 1.28E+03
H= 1.46E+03
I= 1.64E+03

```

Figure C.2: Girth Weld Sample Results - Temperature Plots.

```

time = 0.17407E+02
CONTOURS OF TEMPERATURE
min=-0.354E+02 in element 154
max= 0.178E+04 in element 1556

```

	contour values
A=	1.17E+02
B=	3.05E+02
C=	4.94E+02
D=	6.82E+02
E=	8.71E+02
F=	1.06E+03
G=	1.25E+03
H=	1.44E+03
I=	1.62E+03

Figure C.3: Butt Weld Sample Results - Temperature Plots.

time = 0.13066E+02
 CONTOURS OF FERRITE FRACTION
 min= 0.100E-09 in element 7026
 max= 0.286E+00 in element 2275

contour values
 A= 2.00E-01
 B= 2.13E-01
 C= 2.25E-01
 D= 2.38E-01
 E= 2.50E-01
 F= 2.62E-01
 G= 2.75E-01
 H= 2.87E-01
 I= 3.00E-01

Figure C.4: T Joint Sample Results - Microstructure Plots


```

time = 0.25000E+01
CONTOURS OF MARTENSITE (BASE)
min= 0.000E+00 in element 5302
max= 0.162E+03 in element 5134

```

Figure C.5: Girth Weld Sample Results - Microstructure Plots.

Figure C.6: Butt Weld Sample Results - Microstructure Plots

END

2/3/1 2/92

FIN