Effect of Linear Direction Oscillation on Grain Refinement

THESIS

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By
Agni Arumugam Selvi
Graduate Program in Mechanical Engineering

The Ohio State University
2014

Master's Examination Committee:
Prof Dave F Farson, Advisor
Prof Wei Zhang
Abstract

The objective of this study was to determine the effect of linear oscillation on grain structure refinement in gas tungsten arc welds. Experimental welds were made with various oscillation frequencies and the resulting microstructures were imaged by optical microscopy. Then, the observed grain size was assessed with solidification temperature gradient and boundary velocity predicted by a numerical simulation based on a commercial transport code (Flow 3D). The results show that as the oscillation frequency increased, the dendritic arm spacing decreased.
Dedication

This work is dedicated to my family.
Acknowledgments

I would like to acknowledge the support given by my advisor Prof. Dave Farson and Mr. Steve McCracken from the Electrical Power Research Institute (EPRI) for giving me the opportunity to work in this project. I would also like to thank my lab coworkers Yousub Lee, David Pan, Changkyoo Park for their continuous assistance in helping me familiarize with both experimental and analysis work in the lab.
Vita

March 2007 .................................................. DAV Matric HSS Chennai India.

May 2011 .................................................. B.E. Mechanical Engineering, SSNCE

Anna University, Chennai, India

2011 to present ........................................ Graduate Research Assistant, Department of

MSE, The Ohio State University

Fields of Study

Major Field: Mechanical Engineering
# Table of Contents

Abstract ................................................................................................................................. ii

Dedication .............................................................................................................................. iii

Acknowledgments .................................................................................................................. iv

Vita ........................................................................................................................................ v

Fields of Study ...................................................................................................................... v

Table of Contents ................................................................................................................... vi

List of Tables .......................................................................................................................... viii

List of Figures ......................................................................................................................... ix

Nomenclature ........................................................................................................................ x

Introduction ........................................................................................................................... 1

Objectives ............................................................................................................................. 3

Simulation Modeling ............................................................................................................. 4

Boundary Conditions and Heat Transfer ............................................................................. 5

Experimental Procedure ....................................................................................................... 8

Results and Discussion ......................................................................................................... 11

Experimental Results .......................................................................................................... 11

vi
Simulation Results ........................................................................................................ 14

Conclusion .................................................................................................................. 18

References ................................................................................................................... 19

Appendix A - Material Properties of Inconel 690 alloy ............................................. 22

Appendix B - Optical Micrographs ............................................................................. 25

Appendix C - Custom Subroutine for Heat Input in Flow 3D ..................................... 29

Appendix D - Custom Subroutine for Pressure input and Lorentz force .................. 35
List of Tables

Table 1. Welding trial parameters........................................................................................................8
Table 2. The Composition of Inconel alloy 690, wt-%.........................................................................9
List of Figures

Figure 1 Pattern of Linear direction oscillation............................................................... 9

Figure 2. Top surface of the Experimental Welds .......................................................... 10

Figure 3 a) Images obtained from Optical Microscope b, c & d) Further analysis using ImageJ software. e) Graph of average grain size vs frequency of LBD ......................... 12

Figure 4 Image captured from simulation model............................................................ 14

Figure 5 G-R plots for trial 3 and trial 4. ......................................................................... 15

Figure 6 G-R plots for trial 5 and trial 6 ......................................................................... 16

Figure 7 Comparison between ‘Calculated CET boundary for different nucleation site density overlaid with G and R’ for Magnetic stirring and Linear direction oscillation... 17

Figure 8 Density of Inconel 690 alloy ........................................................................... 22

Figure 9 Specific Heat of Inconel 690 alloy .................................................................... 22

Figure 10 Liquid viscosity of Inconel 690 alloy ............................................................. 23

Figure 11 Thermal conductivity of Inconel 690 alloy .................................................... 23

Figure 12 Surface Tension of Inconel 690 alloy ............................................................ 24

Figure 13 Trial 3 – Optical Microscope graphs and ImageJ analyzed graphs.................. 25

Figure 14 Trial 4 – Optical Microscope graphs and ImageJ analyzed graphs.................. 26

Figure 15 Trial 5 – Optical Microscope graphs and ImageJ analyzed graphs.................. 27

Figure 16 Trial 6 – Optical Microscope graphs and ImageJ analyzed graphs.................. 28
# Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>Temperature Gradient</td>
</tr>
<tr>
<td>R</td>
<td>Solidification Rate</td>
</tr>
<tr>
<td>( \vec{v} )</td>
<td>Velocity Vector</td>
</tr>
<tr>
<td>F</td>
<td>Fraction of fluid occupied by a single simulation cell</td>
</tr>
<tr>
<td>h</td>
<td>Enthalpy</td>
</tr>
<tr>
<td>( \rho_s )</td>
<td>Density of solid</td>
</tr>
<tr>
<td>( C_s )</td>
<td>Specific heat of solid</td>
</tr>
<tr>
<td>( T_s )</td>
<td>Solidus temperature</td>
</tr>
<tr>
<td>( h_{sl} )</td>
<td>Latent heat of fusion (solid-liquid phase change)</td>
</tr>
<tr>
<td>( \rho_l )</td>
<td>Density of liquid phase</td>
</tr>
<tr>
<td>( C_l )</td>
<td>Specific heat of liquid</td>
</tr>
<tr>
<td>( T_l )</td>
<td>Liquidus temperature</td>
</tr>
<tr>
<td>( \sigma_a )</td>
<td>Gaussian heat distribution parameter</td>
</tr>
<tr>
<td>Q</td>
<td>Total heat Input</td>
</tr>
<tr>
<td>( \eta )</td>
<td>Efficiency</td>
</tr>
<tr>
<td>( T_0 )</td>
<td>Ambient temperature</td>
</tr>
<tr>
<td>( h_c )</td>
<td>Coefficient of convection</td>
</tr>
<tr>
<td>I</td>
<td>Current</td>
</tr>
<tr>
<td>( P_{arc} )</td>
<td>Arc pressure</td>
</tr>
<tr>
<td>( \sigma_p )</td>
<td>Gaussian density distribution parameter</td>
</tr>
<tr>
<td>( J_z ) and ( J_r )</td>
<td>z and r components of Lorentz force J</td>
</tr>
<tr>
<td>( J_0 ) and ( J_1 )</td>
<td>Bessel functions of the zero and first order</td>
</tr>
<tr>
<td>( B_0 )</td>
<td>( \theta )-component of B, in the cylindrical co-ordinate system (r,z,( \theta ))</td>
</tr>
<tr>
<td>( N_0 )</td>
<td>Number of nucleation sites</td>
</tr>
<tr>
<td>( \Delta T_N )</td>
<td>Undercooling in temperature needed for nucleation</td>
</tr>
</tbody>
</table>
Introduction

Weld metals with high percentage of chromium and nickel have resistance to cracking caused by stress corrosion. These metals are largely employed in the joining of dissimilar metals and also for the repair of coolant component piping in reactors used in the nuclear power plants (McMinn, 1986). For the past decades, extensive research has been conducted worldwide to further understand and utilize these metals.

But certain characteristics observed make it difficult to apply these materials in certain industries. Some of these characteristics are the presence of high volume of oxides in the weld pool, harmful dilution effects, high viscosity of molten metal causing poor bead shape, ductility dip cracking and solidification cracking. One of the main cause of weld metal cracking is associated with the formation of large columnar grains during the solidification process. These coarse large columnar grains cause low signal/noise ratio during ultrasonic inspection. This adversely affects the size of defects that may be detected by ultrasonic non-destructive examination (Chen, Shi, & Shi, 1999) essential in a nuclear industry.

Thus, the following research aimed at modifying the gas tungsten arc weld (GTAW) process parameters to produce solidification characteristics to make NDE possible.

The main parameters affecting the solidification of grain structures in GTAW are the temperature gradient G, solidification rate, R (liquid/solid interface velocity) as well as
other fluid flow conditions. Prior publications (David & Vitek, 1989 and Kou S., Welding metallurgy, 1987) described various GTAW techniques by which G and R were varied to alter solidification properties.

Processes such as pulsed current gas tungsten arc welding (Sundaresan, Janaki Ram, & Madhusudhan Reddy, 1999) and mechanical arc oscillation (Tewari, 1993) have been employed. Magnetic arc stirring, a method which does not require any addition or changes to the weld pool composition (unlike inoculants) was adapted by several researchers (Arakawa, Hirano, Ishida, Kokawa, & Sato, 1998), (Mousavi, Hermans, Richardson, & den Ouden, 2003), (Pearce & Kerr, 1981), (Vollersten & Thomy, 2006), and (Koteswara Rao, Madhusudhana Reddy, Kamaraj, & Prasad Rao, 2005). They established that the weld pool stirring action can lead to considerable grain structure refinement in alloys of nickel, aluminum and titanium.

Research by (Lim, et al., 2010) showed significant refinement of grain size occurring at a magnetic stirring frequency of ~7 Hz on Inconel 690 substrates. To replicate these effects of magnetic stirring (which requires the installation of additional sophisticated equipment and considerable capital investment) using simple linear direction oscillation is the motivation behind the current research. So, the effect of back and forth oscillation (or linear direction oscillation, LDO) was studied for various oscillating frequencies. It was predicted that LDO leads to refinement of solidification grain structure as it periodically reverses the rear weld pool boundary similar to the action caused by magnetic stirring.

By incorporating previous numerical modeling techniques (Cho, Lim, & Farson, 2006) into the simulation, the analytical model was implemented and run for various
combination of parameters. These simulation results were then used with current theories of columnar to equiaxed transition (CET) to analyze the grain structure refinement observed in the experiments.

**Objectives**

- To create a weld-pool model in Flow 3D to simulate linear direction oscillation of GTAW in order to analyze the effect on weld pool fluctuations.
- To compare the model predictions of grain boundary temperature gradient and solidification velocity to experimental results of solidification microstructure to understand the effects of weld pool fluctuations on fusion zone grain structure.
- To compare the solidification conditions predicted by simulations of cross seam oscillation with the simulation of magnetic stirring.
Simulation Modeling

Flow3D, a commercial CFD software, is used to simulate the arc welding process by solving the mathematical model with its governing equations in 3D Cartesian coordinates of the GTAW process. This includes the numerical solution of mass, momentum, and energy conservation relationships.

The volume of fluid method employed in the simulation, states that the three dimensional conservation of volume for the flow of an incompressible, Newtonian fluid with laminar flow as

\[ \nabla \cdot \mathbf{V} = 0 \]  

(1)

A scalar variable \( F \) is used to monitor the volume fraction of fluid, with values equal to 0 when a cell lies in a void region and equal to 1 if it is in fluid region. Mesh cells containing free surfaces have fractional values of \( F \). The conservation of volume based on \( F \) is as follows

\[ \frac{DF}{Dt} = \frac{\partial F}{\partial t} + \nabla \cdot (\mathbf{V} F) = 0 \]  

(2)

The phase change from solid to liquid is modeled by the enthalpy-temperature relationship given by:

\[
h = \begin{cases} 
\rho_s C_s T_s, & T \leq T_s, \\
h(T_s^l) + h^l, & T_s \leq T \leq T_i, \\
h(T_i^l) + \rho_i C_i (T - T_i), & T_i < T 
\end{cases}
\]  

(3)
Boundary Conditions and Heat Transfer

The heat input from the weld arc was modeled as a Gaussian density function,

\[ q(r) = \frac{Q}{2\pi\sigma_a^2} \exp\left(\frac{-r^2}{2\sigma_a^2}\right) \]  \hspace{1cm} (4)

Where \( \sigma_a \) is the Gaussian heat distribution parameter, which is mainly used to adjust the heat input distribution on the free surface of the weld pool and \( Q \) is the actual heat input directly to the substrate from the arc. Also, convection and radiation effects are applied on the free surface which then leads to the equation,

\[ K \frac{\partial T}{\partial n} = q(r) = \eta \cdot \frac{Q}{2\pi\sigma_a^2} \exp\left(\frac{-r^2}{2\sigma_a^2}\right) - h_c (T - T_0) - \sigma \varepsilon (T^4 - T_0^4) \]  \hspace{1cm} (5)

The Marangoni flow is modeled based on the equations explained in the work of (Cho, Lim, & Farson, 2006). The Gaussian heat distribution parameter was estimated based on the empirical equation obtained from the literature by (Tsai & Eager, 1985). For a 4-mm arc length, the equation, which is a function of current \( I \), is expressed as

\[ \sigma_a = 0.533I^{0.2941} \]  \hspace{1cm} (6)

Lorentz force generates pressure gradients in the negative \( z \) direction of the arc plasma due to flow of the ionized gas. Gaussian density distribution \( \sigma_p \) is used to characterize this
stagnation pressure. The equation is then approximated with the magnitude and radius derived from experimental results.

\[ P_{arc}(r) = \frac{P}{2\pi \sigma^2_p} \exp\left(-\frac{r^2}{2\sigma^2_p}\right) \]  

(7)

The empirical equations for the Gaussian density distribution \( \sigma_p \) and the total force \( P \) and Gaussian are expressed according to (Lin & Eager, 1986) and (Zhang, Kim, & DebRoy, 2004) as

\[ P = -0.04017 + 0.0002553I(N) \]

\[ \sigma_p = 1.4875 + 0.00123I(mm) \]  

(8)

Where \( I \) is the current in A and 60deg electrode tip angle.

The Lorentz force effects were custom built in flow3d using the following equations by (Kou & Sun, 1985).

\[ J_z = \frac{I}{2\pi} \int_0^\infty \lambda J_0(\lambda r) \exp\left(-\lambda z - \frac{\lambda^2 b^2}{12}\right) d\lambda \]  

(9)

\[ J_r = \frac{I}{2\pi} \int_0^\infty \lambda J_1(\lambda r) \exp\left(-\lambda z - \frac{\lambda^2 b^2}{12}\right) d\lambda \]  

(10)

\[ B_\theta = \frac{\mu_0 I}{2\pi} \int_0^\infty J_1(\lambda r) \exp\left(-\lambda z - \frac{\lambda^2 b^2}{12}\right) d\lambda \]  

(11)

\[ J \times B = \frac{\mu_0 I^2}{4\pi^2 \left(r^2 + z^2\right)^{3/2}} \left(1 - \frac{z}{\left(r^2 + z^2\right)^{1/2}}\right) \left(z - \frac{z}{r} r\right) \]  

(12)

Where \( J_z \) and \( J_r \) are z and r components of Lorentz force, \( J \) and, \( J_0 \) and \( J_1 \) are Bessel functions of the zero and first order, \( B_\theta \) is \( \theta \)-component of B, in the cylindrical co-ordinate system \((r,z,\theta)\). \( J \times B \) gives the electromagnetic force for the current source.
The remaining boundary conditions are set to continuative boundary conditions. This ensures the normal derivatives are zero for all quantities including velocity, pressure, and force at the boundary. The full domain has dimensions of 0.030m in the x-direction, 0.012m in the y-direction, and 0.005m in the z-direction. The total number of cells was 225000 with cell size of 200 microns. The expression for the position of the center of the arc on the top surface of the substrate in simulation coordinates are

\[ X = a \sin(2\pi f) t + vt \]
\[ Y = 0, Z = 0 \]  \hspace{1cm} (13)

Where a, f, t and v represent amplitude of linear oscillation, frequency of linear oscillation, time from the beginning of the weld and linear travel speed respectively. The properties of the material Inconel690 alloy obtained from JMat Pro used in the simulation are given in Appendix A.
Experimental Procedure

Linear direction oscillating welds were made on a 26.5 mm thick plate using GTAW process in Jetline Series 9500 with no filler metal. The top and side surfaces were exposed to an ambient temperature of 303 K (Tamb). Welding Current (I) and Voltage (V) were measured to be (170 A, 12 V) using a digital voltmeter and ammeter, provided in the Thermal Arc 400GTSW. The time taken for each welding pass was measured with a digital stop clock. The arc plasma and shielding gas used was high-purity argon (99.99% pure) with linear travel speed of 3.6 ipm (0.00152 m/s). The details of the welding parameter trials and the chemical composition of the weld metal employed are given in Table 1 and Table 2.

<table>
<thead>
<tr>
<th>Trial No.</th>
<th>Oscillation speed in 9200A Oscillation control</th>
<th>Amplitude</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>No oscillation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.11 in = 2.032 mm</td>
<td>2.5 Hz</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.11 in = 2.032 mm</td>
<td>2.38 Hz</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>0.11 in = 2.032 mm</td>
<td>1.56 Hz</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0.11 in = 2.032 mm</td>
<td>0.76 Hz</td>
</tr>
</tbody>
</table>

Table 1. Welding trial parameters
Figure 1 Pattern of Linear direction oscillation

<table>
<thead>
<tr>
<th>Element</th>
<th>Inconel alloy 690</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ni</td>
<td>58.0</td>
</tr>
<tr>
<td>Cr</td>
<td>27.0-31.0</td>
</tr>
<tr>
<td>Fe</td>
<td>7.0-11.0</td>
</tr>
<tr>
<td>C</td>
<td>0.05 max</td>
</tr>
<tr>
<td>Mn</td>
<td>0.5 max</td>
</tr>
<tr>
<td>S</td>
<td>0.015 max</td>
</tr>
<tr>
<td>Cu</td>
<td>0.5 max</td>
</tr>
</tbody>
</table>

Table 2. Composition of Inconel alloy 690, wt.\%
Figure 2. Top surface of the Experimental Welds
Results and Discussion

Experimental Results

To analyze the effectiveness of stirring process in the weld microstructure the top surface of the weld were polished, etched and imaged by optical microscope. Then the images obtained from the optical microscope were further processed by ImageJ software to analyze the grain structure and grain size. Columnar grain structures are evident in the micrograph with 200 µm scale of case ‘zero’ back and forth oscillation and 0.76 Hz oscillation. So, very low frequency stirring is not favorable for grain refinement.

As we increase the oscillation rate, the tendency for columnar grain structure gradually decreases and equiaxed grain structures begin appearing. At 2.5 Hz frequency, the tendency for equiaxed grain structure is more evident in the micrograph shown in Figure 3. Further analysis with ImageJ software reveals the structure and formation of solidification grain boundaries in each of the trials (refer Fig 3 b). (Refer Appendix B for Optical microscope graphs of various magnifications).
Figure 3 a) Images obtained from Optical Microscope b, c & d) further analysis using ImageJ software. e) Graph of average grain size vs. frequency of LBD

The grain orientation maps for the no oscillation welds clearly show columnar grain structure with grain orientation at an angle of 60 deg to the weld travel direction. As the frequency increases the grain boundaries become clearer and well defined even though some misorientation still exists. The figure 3 c) and d) reveal the total number of effective
closed boundaries and their shape if it were to be represented as ellipses. It gives a better understanding of how the grains get divided as the frequency increases.

Also, to increase the efficiency of ultrasonic NDE, it is necessary to analyze the average area of the grains. The ImageJ software helps in determining the size of the grains by calculating the area of the individual ellipses. The average is then formulated for each trial. From the figure it is evident that the average grain size decrease as the frequency of linear oscillation increases. So for a 2.5 Hz oscillation, the average grain size is about 50 µm$^2$ and with no oscillation, it was found to be 10 times larger than 2.5 Hz oscillation.

The theory behind the formation of equiaxed grain structure and the collapse of columnar pattern in welds when stirring methods introduced is effectively put forth in literature by (Mousavi, Hermans, Richardson, & den Ouden, 2003), (Koteswara Rao, Madhusudhana Reddy, Kamaraj, & Prasad Rao, 2005), (Kou & Le, 1985) and (Kou & Le, Nucleation mechanism and grain refining of weld metal, 1986). As the frequency increases the solidification boundary gets continuously affected due to the nature of the back and forth oscillation which helps in breaking down the grain structure with each interruption. This leads to a fine grain growth in the direction of the weld. So to understand the mechanism of grain refinement, a detailed study of how the temperature gradient, solidification rate and fluid flow interact with each other needs to be analyzed.
Simulation Results

The simulations were performed using Flow3D by FlowScience with custom subroutines build to model Gaussian heat input, pressure inputs and Lorentz forces to the simulation. These custom subroutines are presented in Appendices C and D. The simulations done for the trials are analyzed to understand the solidification effects brought down by the linear direction oscillation.

![Image captured from simulation model](image)

**Figure 4** Image captured from simulation model

The simulation patterns are observed to be complicated, characterized by colliding eddies with molten fluid flowing to the colder region from hotter regions. A sample of the simulation is presented in Figure 4. The simulation results can be quantified by extracting the temperature gradient (G) and solidification rate (R) from the rear edge along the
center of the weld pool boundary. The values of G and R for trials 3 to 6 are plotted in the graphs shown in fig. The temperature gradient G oscillates within ~5 x 10^5 range periodically keeping up with the back and forth oscillation. The gradual decline in frequency and range is observed as the linear direction oscillation frequency decreases. In the case of solidification rate, positive and negative peaks are observed in concordance with the oscillation frequency. As the frequency gets lower the frequency and the range is less pronounced. The negative solidification rate is indicative that retrograde oscillation occurs at the rear weld pool boundary causing dendrite fragmentation due to continuous impingement leading to remelting of the region.

Figure 5 G-R plots for trial 3 and trial 4.
The assessment of equiaxed grains formed from dendrite fragmentation under possibly one of the weld conditions is required. From established theory by (Gaumann, Trivedi, & Kurtz, Nucleation ahead of the advancing interface in directional solidification, 1997), (Gaumann, Bezenon, Canalis, & Kurz, 2001) and (Vitek, 2005) for a given probability of equiaxed grains, the boundary between CET can be calculated by

\[
G = \frac{1}{n+1} \left[ \frac{-4\pi N_o}{3 \ln(1 - \phi)} \right]^{\frac{1}{3n}} \left[ 1 - \frac{\Delta T_N^{n+1}}{(aR)^n} \right] (aR)^{\frac{1}{n}}
\]  

(14)

Where \( G \) is the temperature gradient, \( R \) is the solidification rate, \( N_o \) is the number of nucleation sites and \( \Delta T_N \) is the undercooling in temperature needed for nucleation. Also,
a and n are material parameters for a given alloy, which can be calculated using interface response function models. Comparing with the graph for Magnetic Stirring for unstirred and stirred welds (7Hz) (Lim, et al., 2010) as the same alloy is used in the simulation, it can be observed that as the frequency of linear direction oscillation increases the simulation results occupy the area predominantly in the equiaxed dendritic zone whereas the no oscillation simulation rests completely in the columnar region. This is in congruent with the experimental results where equiaxed dendrites were predominant as the frequency increases. The optimum frequency for these weld conditions were not determined due to the limitations of the machinery present in the lab. But the general trend is established wherein the increase of the frequency of linear direction oscillation as the potential to enhance equiaxed dendrite formation.

Figure 7 Comparison between ‘Calculated CET boundary for different nucleation site density overlaid with G and R’ for Magnetic stirring and Linear direction oscillation.
Conclusions

The effects of cross seam oscillation on the motion of the rear weld pool boundary and weld microstructure were studied for frequencies ranging from 0 to 2.5 Hz. Experimental results were quantified using optical microscopy and ImageJ software to determine the columnar and equiaxed dendrite growth regions. Numerical simulations performed with Flow3d software predicted the weld pool solidification temperature gradient and solidification rate for each of the weld trials. It was found that as the frequency of the linear direction oscillation increased, the probability of equiaxed dendritic growth in the weld pool also increased. Thus, linear cross seam arc oscillation could provide comparable grain refinement to circular arc oscillation and is expected to increase the performance of ultrasonic NDE.
References


Appendix A - Material Properties of Inconel 690 alloy

Material properties of Inconel 690 from JMat Pro used in simulation

Figure 8 Density of Inconel 690 alloy

Figure 9 Specific Heat of Inconel 690 alloy
Figure 10 Liquid viscosity of Inconel 690 alloy

Figure 11 Thermal conductivity of Inconel 690 alloy
Figure 12 Surface Tension of Inconel 690 alloy
Appendix B - Optical Micrographs

Figure 13 Trial 3 – Optical Microscope graphs and ImageJ analyzed graphs
Figure 14 Trial 4 – Optical Microscope graphs and ImageJ analyzed graphs
Figure 15 Trial 5 – Optical Microscope graphs and ImageJ analyzed graphs
Figure 16 Trial 6 – Optical Microscope graphs and ImageJ analyzed graphs
Appendix C - Custom Subroutine for Heat Input in Flow 3D

subroutine qsadd

c     this subroutine is called when nsc>0. the call is near the
cend of the cycle, after the pressure/velocity update and fluid
advecton and diffusion,
but before new cells are initialized, nf's are set,
and the chemistry routine is called.

**********************************************************************
** notice **
** this subprogram contains flow science, inc. proprietary **
** trade secret and confidential information. **
** unauthorized use prohibited **
** copyright 1985-2005 flow science, inc. **
**********************************************************************

use mblock_module
use arrays_module
use arrayp_module
use meshcb_module
use voids_module
use obsijk_module

#ifdef SINGLE
    include '../comdeck/precis4.f'
#else
    include '../comdeck/precis.f'
#endif
 include '../comdeck/params.f'
 include '../comdeck/dparam.f'
 include '../comdeck/bafdef.f'
 include '../comdeck/obsd.f'
 include '../comdeck/cntrl.f'
 include '../comdeck/const.f'
 include '../comdeck/dumn.f'
 include '../comdeck/phiou.f'
 include '../comdeck/scala.f'
include '../comdeck/state.f'
include '../comdeck/pardat.f'

c scalar species sources and sinks

c (not currently implemented)

c variable description
-------- --------------------------------
ijk          current cell index
ipjk         cell to right
imjk         cell to left
ijpk         cell to back
ijmk         cell to front
ijkp         cell to top
ijkm         cell to bottom
i            current x index
j            current y index
k            current z index
t            time
delt         time step size
nbl          current mesh block number
x(i)         mesh coordinate at right of cell ijk
xi(i)        cell ijk center
y(j)         mesh coordinate at back of cell ijk
yj(j)        cell ijk center
z(k)         mesh coordinate at top of cell ijk
zk(k)        cell ijk center
delx(i)      cell size in x direction
dely(j)      cell size in y direction
delz(k)      cell size in z direction
rri(i)       correction factor for cylindrical coordinates
              i.e., delta y at x(i) is dely(j)/rri(i)
vf(ijk)      open volume fraction in cell
afr(ijk)     open area fraction at right face
afb(ijk)     open area fraction at back face
aft(ijk)     open area fraction at top face
u(ijk)       x velocity at right face
v(ijk)       y velocity at back face
w(ijk)       z velocity at top face
fn(ijk)      fluid fraction in cell at beginning of cycle
p(ijk)       pressure in cell
tn(ijk)      temperature in cell
rhoe(ijk)    density\times specific energy in cell
arint(ijk)   free surface area in cell
rho(ijk)     density in cell (only for variable density)
nf(ijk)      free surface indicator in cell
              =0 interior fluid cell
c  =1  surface cell - fluid at left
c  =2  surface cell - fluid at right
c  =3  surface cell - fluid at front
c  =4  surface cell - fluid at back
c  =5  surface cell - fluid at bottom
c  =6  surface cell - fluid at top
c  =7  surface cell - cavitating cell
>=8  void cell -- void id number

c  nsc  number of scalars
sclr(ijk,ns) concentration of scalar ns at cell ijk
   after advection and diffusion
   (update this variable to change scalar concentration)
sclrn(ijk,ns) concentration of scalar ns at cell ijk
   at beginning of time step

c  skip over if no scalars exist and this subroutine is used for scalar sources
if(nsc.eq.0) return

c  --- loop over real cells (set boundary cells in subroutine bc)
do 100 k=kprb,kprt
do 100 j=jprf,jprbk
do 100 i=iprl,iprr
----- calculate current cell index
include '../comdeck/ijk.f'
------ skip calculation for completely blocked cells
   if(vf(ijk).lt.em6) goto 100
-------- calculate "neighbor indices"
   include '../comdeck/mijk.f'
   include '../comdeck/pijk.f'
------ skip empty (void) cells
   if(fn(ijk).lt.emf .and. nmat.eq.1) go to 100

================================ SPECIAL CUSTOMIZATION =========================
c  Add thermal energy beam source to fluid surfaces. A phantom obstacle is used to define the beam location. This routine computes an energy deposition where a fluid surface overlaps the phantom obstacle. The obstacle is defined as a type ifob=2 with ospin<>0, oadrg=0, and obdrg=0. (Note: ospin<>0 is needed.) If a surface cell is less than half filled (f<0.5) energy is distributed over surface cell plus its neighbor

c  skip if energy transport is not turned on
   if(ifenrg.eq.0) goto 200

c  --- loop over real cells (set boundary cells in subroutine bc)
c
do 100 k=kprb,kprt
    do 100 j=jprf,jprbk
    do 100 i=iprl,iprr

    c ------ calculate current cell index
    c
    include '../comdeck/ijk.f'
    c
    c ------ skip calculation for completely blocked cells
    c
    if(vf(ijk).lt.em6) cycle
    c
    c ------ skip passive cells
    c
    if(icstat(ijk).le.0) cycle
    c
    c ------ skip non-surface cells
    c
    if(f(ijk).lt.emf .or. fn(ijk).lt.emf) cycle
    c
    c ------ look for empty neighbor which indicates a surface cell
    c
    nff=0
    if(f(ijk-1).lt.emf) nff=2
    if(f(ijk+1).lt.emf) nff=1
    if(f(ijk-ii1).lt.emf) nff=4
    if(f(ijk+ii1).lt.emf) nff=3
    if(f(ijk-ii2).lt.emf) nff=6
    if(f(ijk+ii2).lt.emf) nff=5
    c
    if(nff.eq.0) cycle
    c
    c ------ look for phantom obstacle of type ifob=2 in cell
    c
    c ------ velocity=dum3
    c
    c ------ mag of oscillation, a=dum4
    c
    c ------ ang velocity of oscillation=dum5

    xit=xi(i)-dum3*t-dum4*sin(dum5*t)
    yjt=yj(j)
    zkt=zk(k)
    ijmb=2
    do while(xit.lt.x(ijmb-1).or.xit.gt.x(ijmb))
        ijmb=ijmb+1
        if(ijmb.gt.im1) goto 100
    enddo
    jjmb=2
    do while(yjt.lt.y(jjmb-1).or.yjt.gt.y(jjmb))
        jjmb=jjmb+1
        if(jjmb.gt.jm1) goto 100
    enddo
    kjmb=2
    do while(zkt.lt.z(kjmb-1).or.zkt.gt.z(kjmb))
kjmb=kjmb+1
   if(kjmb.gt.kml) goto 100
enddo
ijkjm=ii2*(kjmb-1)+imax*(jjmb-1)+ijmb
m=ijkfob(ijkjm)
ob=indxob(m)
if(nob.eq.0) cycle
if(ifob(nob).ne.2) cycle
if(abs(oadrg(nob)).gt.ztest .or.
   abs(obdrg(nob)).gt.ztest) cycle
   1   abs(oadrg(nob)).gt.ztest) cycle

------ calculate "neighbor indices"
include '../comdeck/mijk.f'
include '../comdeck/pijk.f'

------ let delteng be thermal energy flux (per unit area and time)
assume energy directed in negative z direction of phantom
obstacle before it may be copied into a new orientation with
normal components (amox,amoy,amoz).
call srfnorm to get surface normal (snx,sny,snz) for angle of
incidence.
c
cmoxt=cmox(nob)+dum3*t+dum4*sin(dum5*t)
cmoyt=cmoy(nob)
call srfnorm(fn,ijk,snx,sny,snz,sar,leneax,0)
dotnorm=snx*amox(nob)+sny*amoy(nob)+snz*amoz(nob)
if(dotnorm.le.zero) goto 100

c ------ normal distance from center of phantom obstacle is disnorm.
this may be used to impose a shape factor on the beam energy,
i.e., here we use a Gaussian decay with radius from the
center:
c
rsq=(xi(i)-cmoxt)**2+
   1   (yi(j)-cmoyt)**2+
   2   (zk(k)-cmoz(nob))**2
rdotz=(xi(i)-cmoxt)*amox(nob)+
   1   (yi(j)-cmoyt)*amoy(nob)+
   2   (zk(k)-cmoz(nob))*amoz(nob)
disnorm=sqrt(rsq-rdotz**2)
sigma_arc=0.533*0.001*dum6**0.2941
argexp=(disnorm/sigma_arc)**2

c--------calculate energy

c--------current=dum6

c--------voltage=dum7
power=dum6*dum7*0.7
bamp=(power*exp(-0.5*argexp))/(2*3.14*(sigma_arc)**2))
delteng=delt*arint(ijk)*vjkobs(m)*bamp*dotnorm
inbr=ijk
vfac=one
if(fn(ijk).ge.0.5) goto 80
goto(10,20,30,40,50,60) nff
10  continue
   if(afr(ijk-1).gt.em6) inbr=ijk-1
      vfac=delx(i-1)*rdx(i)*rri(i)/rri(i-1)
      goto 80
20  continue
   if(afr(ijk).gt.em6) inbr=ijk+1
      vfac=delx(i+1)*rdx(i)*rri(i)/rri(i+1)
      goto 80
30  continue
   if(afb(ijk-ii1).gt.em6) inbr=ijk-ii1
      vfac=dely(j-1)*rdy(j)
      goto 80
40  continue
   if(afb(ijk).gt.em6) inbr=ijk+ii1
      vfac=dely(j+1)*rdy(j)
      goto 80
50  continue
   if(aft(ijk-ii2).gt.em6) inbr=ijk-ii2
      vfac=delz(k-1)*rdz(k)
      goto 80
60  continue
   if(aft(ijk).gt.em6) inbr=ijk+ii2
      vfac=delz(k+1)*rdz(k)
80  continue
   cvol=vf(ijk)*delx(i)*dely(j)*delz(k)/rri(i)
   if(inbr.ne.ijk) then
      cvoln=cvol*vfac*vf(inbr)/vf(ijk)
   else
      cvoln=cvol
   endif

c   denom=cvol*fn(ijk)+cvoln*fn(inbr)
   rhoe(ijk)=rhoe(ijk)+delteng*fn(ijk)/denom
   rhoe(inbr)=rhoe(inbr)+delteng*fn(inbr)/denom

c100  continue
200  continue
return
end
Appendix D - Custom subroutine for Pressure input and Lorentz forces

subroutine forcal
  
evaluate additional forces on fluid -- user customizable

  ****************************************************************************************************
  ** notice **
  ** this subprogram contains flow science, inc. proprietary **
  ** trade secret and confidential information. **
  ** unauthorized use prohibited **
  ** copyright 1985-2002 flow science, inc. **
  ****************************************************************************************************

  use arrays_module
  use meshcb_module
  use avgco_module
  use obsijk_module
  
  #ifdef SINGLE
  include '../comdeck/precis4.f'
  #else
  include '../comdeck/precis.f'
  #endif
  include '../comdeck/params.f'
  include '../comdeck/const.f'
  include '../comdeck/cntrl.f'
  include '../comdeck/edit.f'
  include '../comdeck/scala.f'
  include '../comdeck/dumn.f'
  include '../comdeck/obsd.f'
  include '../comdeck/cbusr.f'

  integer ii, nxmax
  real meshgap
  parameter(nxmax=100, meshgap=0.0001)
  real A_A,B_B_B,C_C,J0,J1,mu_o, mu_m, mu_r, r_r, z_z
  & ,sumAA,sumBB,sumCC, lamd, Coefl, ar_r, az_z
  real dummaa(5000), sig_rr(nxmax,nxmax), sig zz(nxmax,nxmax)
save sig_rr, sig_zz, timepulse

c fsigx(ijk)= x component of added total force on fluid in cell

ijk
c fsigy(ijk)= y component of added total force on fluid in cell

ijk
c fsigz(ijk)= z component of added total force on fluid in cell

ijk
c
c these forces are either zero or contain non-zero contributions

from thermo-capillary surface forces if ifsten<>0

c
c the forces are located at grid cell centers

c
c ------- definition of quantities for customizing -------

variable description

-----------------------------

ijk current cell index

ipjk cell to right

imjk cell to left

ijpk cell to back

ijmk cell to front

ijkp cell to top

ijkm cell to bottom

i current x index

j current y index

k current z index

c
t time

delt time step size

x(i) mesh coordinate at right of cell ijk

xi(i) cell ijk center

y(j) mesh coordinate at back of cell ijk

yj(j) cell ijk center

z(k) mesh coordinate at top of cell ijk

zk(k) cell ijk center

delx(i) cell size in x direction

dely(j) cell size in y direction

delz(k) cell size in z direction

rri(i) correction factor for cylindrical coordinates

i.e., delta y at x(i) is dely(j)/rri(i)

c vf(ijk) open volume fraction in cell

c afr(ijk) open area fraction at right face

c afb(ijk) open area fraction at back face

c aft(ijk) open area fraction at top face

c
c u(ijk) x velocity at right face

c v(ijk) y velocity at back face

c w(ijk) z velocity at top face

c fn(ijk) fluid fraction in cell at beginning of cycle

c p(ijk) pressure in cell

c tn(ijk) temperature in cell

c rhoe(ijk) density*specific energy in cell

c arint(ijk) free surface area in cell
c       rho(ijk)       density in cell (only for variable density)
c
nf(ijk)        free surface indicator in cell
   =0           interior fluid cell
   =1           surface cell - fluid at left
   =2           surface cell - fluid at right
   =3           surface cell - fluid at front
   =4           surface cell - fluid at back
   =5           surface cell - fluid at bottom
   =6           surface cell - fluid at top
   =7           surface cell - cavitating cell
   >=8          void cell -- void id number

c       nsc            number of scalars
sclr(ijk,ns)   concentration of scalar ns at cell ijk
   after advection and diffusion
   (update this variable to change scalar concentration)
sclrn(ijk,ns)  concentration of scalar ns at cell ijk
   at beginning of time step

****** initialize force arrays if not done in thermocapillary model ******

if(ifsten.eq.0) then
   do k=1,km1
      do j=1,jm1
         do i=1,im1
            include '../comdeck/ijk.f'
            fsigx(ijk)=zero
            fsigy(ijk)=zero
            fsigz(ijk)=zero
         enddo
      enddo
   enddo
endif

... enter changes here ...

====== Calculate the Lorentz force at the beginning of simulation ======
corfac=0.0
if(t.gt.em6) goto 2500

   do 2000 rindex=1,nxmax
      do 2000 zindex=1,nxmax

2000 continue

====== User Input for MHD ==============
PI=3.14159265359
mu_r=1.0
mu_o=4*PI*0.0000001
\[ \mu_m = \mu_r \mu_o \]
\[ \text{coef1} = 5.0 \]

---

**End User Input for MHD**

---

**Arc pressure calculation**

```fortran
  do 1000 k=kprb,kprt
  do 1000 j=jprf,jprbk
  do 1000 i=iprl,iprr

  c ---- calculate current cell index
      include '../comdeck/ijk.f'

  c ---- skip calculation for completely blocked cells
  if(vf(ijk).lt.em6) goto 1000

  c ---- skip non-surface cells
  if(nf(ijk).eq.0 .or. nf(ijk).ge.8) goto 1000

  c ---- look for empty neighbor which indicates a surface cell
  nff=0
  if(f(ijk-1).lt.emf) nff=2
  if(f(ijk+1).lt.emf) nff=1
  if(f(ijk-ii).lt.emf) nff=4
  if(f(ijk+ii).lt.emf) nff=3
  if(f(ijk-ii2).lt.emf) nff=6
  if(f(ijk+ii2).lt.emf) nff=5

  if(nff.eq.0) cycle

  current=dum7
  power=dum7*dum8*0.7

  c ---- look for phantom obstacle of type ifob=2 in cell
  c ---- velocity=dum3
  c ---- mag of oscillation, a=dum4
  c ---- ang velocity of oscillation=dum5

  xit=xi(i)-dum3*t-dum4*sin(dum5*t)
  yjt=yj(j)
  zkt=zk(k)
  ijmb=2
  do while(xit.lt.x(ijmb-1).or.xit.gt.x(ijmb))
    ijmb=ijmb+1
    if(ijmb.gt.im1) goto 1000
  enddo
  jjmb=2
  do while(yjt.lt.y(jjmb-1).or.yjt.gt.y(jjmb))
    jjmb=jjmb+1
    if(jjmb.gt.jm1) goto 1000
  enddo
```

---

38
kjmb=2
do while(zkt.lt.z(kjmb-1).or.zkt.gt(z(kjmb))
   kjmb=kjmb+1
   if(kjmb.gt.km1) goto 1000
enddo
ijkjmb=ii2*(kjmb-1)+imax*(jjmb-1)+ijmb+i5
nob=indxob(m)
if(nob.eq.0) goto 1000
if(nob.eq.2) goto 1000
if(ifob(nob).ne.2) goto 1000
if(abs(oadrg(nob)).gt.ztest .or.  
   abs(obdrg(nob)).gt.ztest) goto 1000
  c
  apxt=cmox(nob)+dum3*t+ dum4*sin(dum5*t)
  apyt=cmoy(nob)
  apzt=cmoz(nob)
  c
  ------ Rotation of Arc -------------------
  rot_angle=0.0
  amoxt=0.0
  amoxt=sin(rot_angle*PI/180)
  amoxt=cos(rot_angle*PI/180)
  c
  call sr fmapnorm(fn,ijk,snx,sny,snz,sar,leneax)
  dotnorm=snx*amoxt+sny*amoxt+snz*amoxt
  if(dotnorm.le.ztest) goto 1000
  c
  ------ normal distance from center of distribution is disnorm.
  c this distance is used to impose a shape factor on the
  pressure,
  i.e., a Gaussian with radius from the center:
  rsq=(xi(i)-apxt)*(xi(i)-apxt)+
  (zk(k)-apzt)*(zk(k)-apzt)
  c
  ------ add y distance if not cylindrical coordinates
  if(cyl.lt.half) rsq=rsq+(yj(j)-apyt)*(yj(j)-apyt)
  rdotz=(xi(i)-apxt)*amoxt+
  (yj(j)-apyt)*amoxt+
  (zk(k)-apzt)*amozt
  disnorm=sqrt(rsq-rdotz*rdotz)
  c
  ------ evaluate Guassian distribution
  sigma_arc=0.533*0.001*current**0.2941
  argexp=(disnorm/sigma_arc)**2
  papp=(mu_o*current**2)*exp(-0.5*argexp)/(4*(PI*sigma_arc)**2)
  c
  ------ compute surface force magnitude as pressure*area
  c (factor of two needed because of force averaging)
  pforce=-2.0*arint(ijk)*papp*dotnorm
  c
  ------ update cell centered forces in surface cells
  fsigx(ijk)=fsigx(ijk)+pforce*snx
  fsigy(ijk)=fsigy(ijk)+pforce*sny
  fsigz(ijk)=fsigz(ijk)+pforce*snz
There is a possible instability here because of forces applied in directions normal to the axis of pressure jet. This happens because the applied pressure is constant and does not adjust to surface elevation changes. The instability will not occur if surface tension or gravity forces are large enough to smooth any horizontal variations in elevation caused by the applied pressure.

---

store the 'k' index for MHD evaluation

---

\[
i_{jpp} = \text{imax} \times (j-1) + i \\
dummya(i_{jpp}) = k
\]

1000 continue

---

Evaluate the distance \((r, z)\) from the center of the arc

---

\[
\begin{align*}
  r_r &= \text{meshgap} \times (r_{\text{index}} - 0.5) \\
  z_z &= \text{meshgap} \times (z_{\text{index}} - 0.5)
\end{align*}
\]

---

End of Evaluate the distance \((r, z)\)

---

\[
\begin{align*}
  \text{sumAA} &= 0 \\
  \text{sumBB} &= 0 \\
  \text{sumCC} &= 0
\end{align*}
\]

\[
\begin{align*}
  \sigma_{\text{arc c}} &= 0.001 \times 0.5342 \times \text{current}^{0.2684} \\
  \text{gcc} &= 0.5
\end{align*}
\]

do 100 ii=1,2001

\[
\begin{align*}
  \lambda &= \text{coef1} \times (ii-1) \\
  \lambda &= r_r \times \lambda \\
  J0 &= \text{bessj0}(\lambda) \\
  J1 &= \text{bessj1}(\lambda) \\
  A_A &= \lambda \times \text{J0} \times \exp(-\lambda \times z_z \times \lambda^2 \times \sigma_{\text{arc c}}^2 / (\text{gcc} \times 4.0)) \\
  B_B_B &= \lambda \times \text{J1} \times \exp(-\lambda \times z_z \times \lambda^2 \times \sigma_{\text{arc c}}^2 / (\text{gcc} \times 4.0)) \\
  C_C &= \text{J1} \times \exp(-\lambda \times z_z \times \lambda^2 \times \sigma_{\text{arc c}}^2 / (\text{gcc} \times 4.0)) \\
  \text{sumAA} &= \text{sumAA} + A_A \\
  \text{sumBB} &= \text{sumBB} + B_B_B \\
  \text{sumCC} &= \text{sumCC} + C_C
\end{align*}
\]

100 continue

\[
\begin{align*}
  J_z &= \text{current} / (2.0 \times \pi) \times \text{sumAA} \times \text{coef1} \\
  J_r &= \text{current} / (2.0 \times \pi) \times \text{sumBB} \times \text{coef1} \\
  B_0 &= \mu_m \times \text{current} / (2.0 \times \pi) \times \text{sumCC} \times \text{coef1}
\end{align*}
\]

\[
\begin{align*}
  \text{sig}_{rr}(r_{\text{index}}, z_{\text{index}}) &= \text{abs}(J_z \times B_0) \\
  \text{sig}_{zz}(r_{\text{index}}, z_{\text{index}}) &= \text{abs}(J_r \times B_0)
\end{align*}
\]

2500 continue
c ====== Calculate the Lorentz force at the begining of simulation ======

**SPECIAL CUSTOMIZATION FOR MHD**

```fortran
if(idum2.ne.4) goto 1200

open(unit=8, file='data.txt', status='replace')

**calculate the depth from free surface**

```fortran
k=2,km1
j=2,jm1
i=2,im1

include '../comdeck/ijk.f'

ijp

---

Do not store completely blocked cells

if(vf(ijk).lt.em6) goto 1150

---

Do not store non-surface cells

if(f(ijk).lt.emf .or. fn(ijk).lt.emf) goto 1150

dumma(ijpp)=k
write(8,*') dumma
1150 continue

do 1110 k=2,km1
j=2,jm1
i=2,im1

---

calculate current cell index

include '../comdeck/ijk.f'

---

skip calculation for empty cells

if(fn(ijk).lt.em6) goto 1110

---

Edited by Min 10-14-04

if(tn(ijk).lt.1768) goto 1110

---

Edited by Min 10-14-04

---

position parameters

---

look for phantom obstacle of type ifob=2 in cell

xit=x(i)-dum3*t-dum4*sin(dum5*t)
yjt=y(j)
zkt=z(k)

ijmb=2

do while(xit.lt.x(ijmb-1).or.xit.gt.x(ijmb))
   ijmb=ijmb+1
   if(ijmb.gt.im1) goto 1110
enddo

jjmb=2

do while(yjt.lt.y(jjmb-1).or.yjt.gt.y(jjmb))
   jjmb=jjmb+1
   if(jjmb.gt.jm1) goto 1110
enddo

kjmb=2

do while(zkt.lt.z(kjmb-1).or.zkt.gt.z(kjmb))
   kjmb=kjmb+1
   if(kjmb.gt.km1) goto 1110
enddo

ijkjm=ii2*(kjmb-1)+imax*(jjmb-1)+ijmb+ii5
```
m=ijkfob(ijkjmb)
nob=indxob(m)
if(nob.eq.0) goto 1110
if(nob.eq.2) goto 1110
if(ifob(nob).ne.2) goto 1110
if(abs(oaodrg(nob)).gt.ztest .or.
1 abs(obdrd(nob)).gt.ztest) goto 1110

c
open(unit=8, file='data.txt', status='replace')
c
open(unit=9, file='min_z1.txt', status='replace')
apxt=cmox(nob)+dum3*t+dum4*sin(dum5*t)
apyt=cmoy(nob)
apzt=cmoz(nob)

c ------ Rotation of Arc -----------------
rot_angle=0.0
amoxt=0.0
amoyt=sin(rot_angle*PI/180.0)
amozt=cos(rot_angle*PI/180.0)

rsq=(xi(i)-apxt)*(xi(i)-apxt)+
1 (yj(j)-apyt)*(yj(j)-apyt)+
2 (zk(k)-apzt)*(zk(k)-apzt)

rdotz=(xi(i)-apxt)*amoxt+
1 (yj(j)-apyt)*amoyt+
2 (zk(k)-apzt)*amozt
disnorm=sqrt(rsq-rdotz*rdotz)
ar_r=disnorm

c ------ Find the index of z at the surface ----------
ijp=imax*(j-1)+i
kkp=floor(dummaa(ijp))
az_z=abs(zk(kkp)+0.5*delz(kkp)-zk(k))
c
open(unit=11, file='data.txt', status='replace')
c
write(11,*) sig_rr(1,40:50),sig_rr(40:50,1)
c
write(11,*) kkp,zk(kkp), zk(k), z_z
c
z_z=zk(k)
c
-----------------------------------------------------
c
------ Evaluate the r and z indexes -----------------
do 10 rindex=1,nxmax
   r_rp=meshgap*(rindex-0.5)
   if(ar_r.ge.r_rp.and.ar_r.lt.r_rp+0.5*meshgap) goto 11
10 continue
   r_rp=0.5*meshgap
11 continue
do 12 zindex=1,nxmax
   z_zp=meshgap*(zindex-0.5)
   if(az_z.le.z_zp.and.z_z.gt.z_zp+0.5*meshgap) goto 13
12 continue
   z_zp=0.5*meshgap
13 continue
c ------ End of Evaluate the r and z indexes ----------
c ------ First order Linear interpolation for r and z direction ------

rindexp=rindex+1
zindexp=zindex+1

c

r_slope=(sig_rr(rindexp,zindex)-sig_rr(rindex,zindex))/meshgap
z_slope=(sig_zz(rindex,zindexp)-sig_zz(rindex,zindex))/meshgap

c

if(r_r.le.0.5*meshgap) then
  sig_r=sig_rr(rindex,zindex)-r_slope*(0.5*meshgap-ar_r)
else
  sig_r=sig_rr(rindex,zindex)+r_slope*(ar_r-r rp)
endif

if(z_z.le.0.5*meshgap) then
  sig_z=sig_zz(rindex,zindex)-z_slope*(0.5*meshgap-az_z)
else
  sig_z=sig_zz(rindex,zindex)+z_slope*(az_z-z_zp)
endif
c

if(t.ge.delt) goto 2200
c

Do 21 i=1,11

write(9,*) (sig_z(i,j),j=1,11)
21 continue
c

c ------- Convert from axisymmetry to 3D cartesian -------
c

yyc=yj(j)-apyt
xxc=x1(i)-apxt
theata=atan(abs(yyc/xxc))
if(yyc.ge.0.and.xxc.ge.0) theata=theata
if(yyc.ge.0.and.xxc.le.0) theata=PI-theata
if(yyc.le.0.and.xxc.le.0) theata=PI+theata
if(yyc.le.0.and.xxc.ge.0) theata=-theata

fmhdx=-sig_r*cos(theata)*delx(i)*dely(j)*delz(k)
fmhdy=-sig_r*sin(theata)*delx(i)*dely(j)*delz(k)
fmhdz=-sig_z*delx(i)*dely(j)*delz(k)
c

write(8,*) fmhdx,fmhdy,fmhdz,theata
c

t-timepulse, period,
fsigx(ijk)=fsigx(ijk)+fmhdx*f(ijk)
fsigy(ijk)=fsigy(ijk)+fmhdy*f(ijk)
fsigz(ijk)=fsigz(ijk)+fmhdz*f(ijk)
c

1110 continue
1100 continue

return
FUNCTION bessj0(x)
REAL bessj0, x
REAL ax, xx, z
DOUBLE PRECISION p1, p2, p3, p4, p5, q1, q2, q3, q4, q5, r1, r2, r3, r4, r5, r6,
* s1, s2, s3, s4, s5, s6, y
SAVE p1, p2, p3, p4, p5, q1, q2, q3, q4, q5, r1, r2, r3, r4, r5, r6, s1, s2, s3, s4,
* s5, s6
DATA p1, p2, p3, p4, p5/1.d0, -0.1098628627d-2, -2.2734510407d-4,
* -.2073370639d-5, .2093887211d-6, q1, q2, q3, q4, q5/-1.5624999995d-1,
* .1430488765d-3, -.6911147651d-5, 7.6210951616d-6, -9.34945152d-7/
DATA r1, r2, r3, r4, r5, r6/57568490574.d0, -13362590354.d0,
* 651619640.7d0, -11214424.18d0, 77392.33017d0, -184.9052456d0/, s1, s2,
* s3, s4, s5, s6/57568490411.d0, 1029332985.d0, 949680.718d0,
* 59272.64853d0, 267.8532712d0, 1.d0/
if(abs(x).lt.8.)then
  y=x**2
  bessj0= (r1+y*(r2+y*(r3+y*(r4+y*(r5+y*r6)))))/(s1+y*(s2+y*(s3+y*
* (s4+y*(s5+y*s6)))))
else
  ax=abs(x)
  z=8./ax
  y=z**2
  xx=ax-.785398164
  bessj0=sqrt(.636619772/ax)*(cos(xx)*(p1+y*(p2+y*(p3+y*(p4+y*
* p5))))-z*sin(xx)*(q1+y*(q2+y*(q3+y*(q4+y*q5)))))
endif
RETURN
END

FUNCTION bessj1(x)
REAL bessj1, x
REAL ax, xx, z
DOUBLE PRECISION p1, p2, p3, p4, p5, q1, q2, q3, q4, q5, r1, r2, r3, r4, r5, r6,
* s1, s2, s3, s4, s5, s6, y
SAVE p1, p2, p3, p4, p5, q1, q2, q3, q4, q5, r1, r2, r3, r4, r5, r6, s1, s2, s3, s4,
* s5, s6
DATA r1, r2, r3, r4, r5, r6/72362614232.d0, 7895059235.d0,
* 242396853.1d0, -2972611.439d0, 15704.48260d0, -30.16036606d0/, s1, s2,
* s3, s4, s5, s6/144725228442.d0, 2300535178.d0, 18583304.74d0,
* 99447.43394d0, 376.9991397d0, 1.d0/
DATA p1, p2, p3, p4, p5/1.d0, .183105d-2, -.3516396496d-4,
* .2457520174d-5, -.240337019d-6, q1, q2, q3, q4, q5/-.04687499995d0,
* .2002690873d-3, .8449199096d-5, -88228987d-6, .105787412d-6/
if(abs(x).lt.8.)then
  y=x**2
  bessj1=x*(r1+y*(r2+y*(r3+y*(r4+y*(r5+y*r6)))))/(s1+y*(s2+y*(s3+
* y*(s4+y*(s5+y*s6)))))
else
  ax=abs(x)

44
z = 8 / ax
y = z ** 2
xx = ax - 2.356194491

bessj1 = sqrt(.636619772 / ax) * (cos(xx) * (p1 + y * (p2 + y * (p3 + y * (p4 + y * p5)))) - z * sin(xx) * (q1 + y * (q2 + y * (q3 + y * (q4 + y * q5)))))) * sign(1., x)
endif
return
END