University of California, San Diego UCSD-ENG-112

Thermal and Mechanical Analysis of IFE Direct Drive Targets

B. R. Christensen

July 8, 2004

Fusion Division
Center for Energy Research

University of California, San Diego
La Jolla, CA 92093-0417

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Thermal and Mechanical Analysis of IFE Direct Drive Targets

A thesis submitted in partial satisfaction of the requirements for the degree of Master of Science in Engineering Sciences (Mechanical Engineering)

by

Brian Russell Christensen

Committee in charge:

Mark S. Tillack, Chair
Farhat Beg
A. René Raffray
George R. Tynan

2004

The thesis of Brian Russell Christensen is approved:

_____

_____

_____

_____
Chair

University of California, San Diego

2004

TABLE OF CONTENTS

LIST OF SYMBOLS

a	characteristic molecular velocity of a gas (m/s).
c_p	specific heat capacity (J/kg-K).
c_p*	apparent specific heat (J/kg-K).
d_{DT}	diameter of a DT molecule (m).
f	number flux of atoms at a surface (atoms/m^2s).
h	heat transfer coefficient (W/m^2-K).
Δh	change in specific enthalpy (J/kg).
j	mass flux (kg/m^2-s).
k	thermal conductivity (W/m-K).
k_b	Boltzmann's constant (J/K-molecule).
n	number density (m^{-3}).
p	uniform pressure (Pa).
p_{sat}	saturation pressure (Pa).
q''	heat flux at the surface of the target (W/m^2).
r_a	outer radius of DT solid (m).
r_b	inner radius of DT solid (m).
r_{pol}	radius of the polymer target shell (m).
t	time (s).
Δt	time step in the finite difference scheme (s).
t_f	thickness of the foam insulator (m).
t_o	thickness of the outer polymer shell of an insulated target (m).
t_p	thickness of the inner polymer shell of an insulated target (m).
t_{pol}	thickness of the polymer shell for a basic target (m).
u	velocity (m/s).
A	area (m^2).
E_{DT}	elastic modulus of DT (Pa).
E_{pol}	elastic modulus for the polymer (Pa).
E_T	average translational energy of a gas (J/mole).
Kn	Knudsen number.
L_f	latent heat of fusion (J/kg).
L_v	the latent heat of vaporization (J/kg).
M	molecular weight (kg/mole).
N_A	Avogadro's number (1/mole).
P_f	pressure at the surface of a liquid or solid (Pa).
P_g	ambient gas pressure (Pa).
$P_{cr,DT}$	critical pressure of DT (Pa).
R	universal gas constant (J/K-mol).
Su_{Poly}	ultimate strength of polystyrene (Pa).
T	temperature (K).
$T_{cr,DT}$	critical temperature of DT (K).
T_f	temperature at the surface of a liquid or solid (Pa).
T_g	ambient gas temperature (K).
T_{sat}	saturation temperature (K).

$T_{TP,\,DT}$	triple point temperature of DT (K).
T_w	reactor wall temperature (K).
ΔT_{pc}	assumed temperature range for phase change (K).
V	volume (m^3).
α	accommodation coefficient.
η	specific enthalpy (J/kg).
φ	foam porosity.
κ	thermal diffusivity (m^2/s-K).
λ	vapor layer thickness (m).
μ	reflectivity of the outer surface of the target.
ν_l	liquid specific volume (m^3/kg).
θ	angle measured from the positive x-axis (rad).
ρ	density (kg/m^3).
σ_c	condensation coefficient.
σ_e	evaporation coefficient.
$\sigma_{S\text{-}B}$	Stefan-Boltzmann constant (W/m^2 K^4).
σ_I	initial trapping probability.
υ_{pol}	Poisson's ratio for the polymer.
υ_{DT}	Poisson's ratio for DT.
ω	surface tension (N/m).
ψ_{To}	translational energy of a molecule before interacting with a surface.
ψ_{Tc}	translational energy of a molecule after interacting with a surface.
ψ_{Ts}	translational energy of a molecule at the surface temperature.
ζ	mean free path (m).
Γ	weighting function.

LIST OF FIGURES

LIST OF TABLES

ABSTRACT OF THE THESIS

Thermal and Mechanical Analysis

of IFE Direct Drive Targets

by

Brian Russell Christensen

Master of Science in Engineering Sciences (Mechanical Engineering)

University of California, San Diego, 2004

Mark S. Tillack, Chair

During injection, inertial fusion energy (IFE) direct-drive targets are subjected to heating from energy exchange with the background gas and radiation from the reaction chamber wall. This thermal loading could cause unacceptable deformation of the target due to phase change (vaporization and/or melting) of the deuterium-tritium (DT) and/or thermal expansion. The objectives of this thesis are to quantify and characterize the thermal loading of the target under various chamber conditions, model the behavior of direct-drive targets subjected to an imposed heat flux, demonstrate the potential various target designs, and propose methods for resolving outstanding issues.

The high Knudsen number (Kn ~ 1-100) flow around the target, including heat transfer, is modeled using DS2V (a commercially available DSMC program). The coupled thermal and mechanical response of the target to an imposed heat flux is modeled using a one-dimensional finite difference numerical computer model.

The results of this study illustrate the potential of decreasing the initial target temperature, insulating the target, and allowing phase change. A particularly exciting result is the apparent elimination or minimization of vapor defects under certain conditions. It is concluded that a more sophisticated 2-dimensional model is needed to gain further insight into the effects of phase change and thermal expansion.

CHAPTER 1

Introduction

The direct drive fusion concept utilizes multiple laser beams (drivers) to compress and heat small spherical pellets (targets) loaded with fusion fuel. The compression and heating process occurs when the rapid deposition of energy at the surface of the target causes ablation (net outward mass flux) and hence a reaction force directed inward (implosion). Subsequent energy pulses from the driver, on the already compressed fusion fuel, results in a sufficient fusion fuel density and temperature to initiate a fusion fuel burn [1].

Due to the nature of the implosion process (accelerating the light, low density plasma into heavy, high density target material) the Rayleigh-Taylor instability, where small initial perturbations are amplified in time, will always be manifest. The amplification of initial perturbations impedes the compression of a target (Fig. 1.1) [2,31-34]. Thus, perturbations caused by surface roughness, vapor bubbles, or other inconsistencies must be minimized in order to maximize the implosion efficiency.

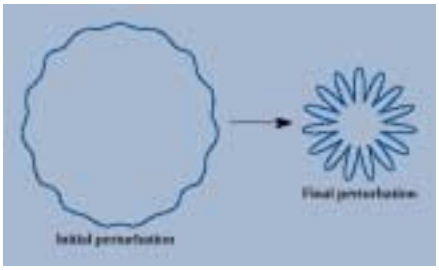

Figure 1.1. During implosion, small amplitude initial perturbations will grow due to the Rayleigh-Taylor instability, resulting in less fuel compression [2].

To reduce the amount of driver beam steering, and to ensure that the energy from the various driver beams is deposited symmetrically on the surface of the target at essentially the same time, the target must be accurately and repeatably placed at a specified point in the reaction chamber. The displacement of the target from its intended final location will be highly dependent on the background gas density in the reaction chamber. The background gas

density also largely determines the heat load on the target; therefore, the background gas density couples the design of the reactor and targets.

It is expected that the process of presenting a direct-drive target at chamber center and imploding it must occur at a rate of ~ 5-10 Hz [1] in an inertial fusion energy (IFE) power plant. For this thesis it is assumed that the direct-drive target is presented at chamber center by pneumatically injecting the target at ~ 400 m/s.

This thesis focuses on modeling the thermal loading and response of direct-drive targets under loading. A schematic of a typical IFE direct-drive target considered in this paper is shown in Fig. 1.2. The target is mainly composed of solid deuterium-tritium (DT).

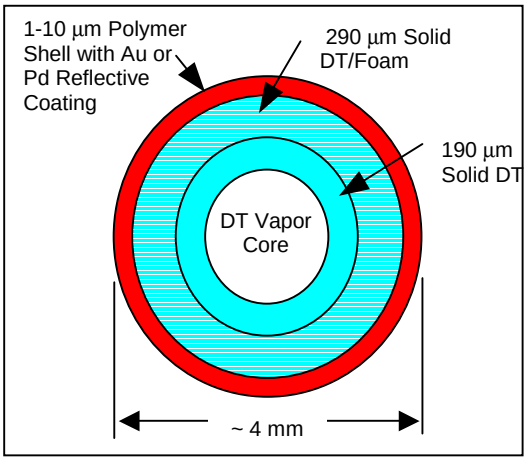

Figure 1.2. A typical direct-drive IFE target considered in this study (not to scale).

An understanding of the thermal response of the target is important; since, as the temperature of the target is increased thermal expansion and phase change could threaten the integrity of the target. While the amount of allowable target deformation is not well established, the previous assumption was that the maximum DT temperature must remain below the triple point of the DT ($T_{TP, DT}$ = 19.79 K) [3].

This thesis presents the results of a detailed numerical study of the thermal and mechanical response of a direct-drive target to a simulated IFE chamber environment. The study is based on a numerical model that has been developed in the Advanced Energy Technology Group at the University of California, San Diego. First, the sources of heating are investigated and discussed. The need for a novel numerical model, and a description and validation of the model, are then presented. The results of several parametric studies

are then presented and the potential of several design solutions are discussed. Finally, conclusions and recommendations are offered.

CHAPTER 2

Thermal Loading

To model the thermal and mechanical behavior of an IFE target in a reaction chamber environment, the thermal loading resulting from the reaction chamber environment must be quantified. Radiation and convection (with condensation) are the thermal loading mechanisms considered in this thesis. The radiation loading is simply calculated based on expected reaction chamber wall temperatures and the target surface reflectivity. Due to the high Knudsen number (~ 1-10) flow around a target in typical chamber environments, the convective loading is calculated using a DSMC program.

2.1 Radiation Heat Flux

An estimate of the radiation heat flux at the surface of the target is given by:

$$q''_{rad} = (1-\mu)\sigma_{S-B}T_w^4 \tag{2.1}$$

where μ is the reflectivity of the outer surface of the target, $\sigma_{S\text{-}B}$ is the Stefan-Boltzmann constant, and T_w is the reactor wall temperature. It has been proposed that the surface of the target be coated with Au or Pd. The results from a multi-layer wave model show that μ is ~ 0.96 (96% reflective) for 400 Å of Au [3, 28, 29]. Table 2.1 shows the results of Eq. 2.1 for two reactor wall temperatures.

Table 2.1. Radiation heat flux based on expected reactor wall temperatures assuming the target coating is 96% reflective.

Reactor Wall Temperature (K)	Radiation Heat Flux (W/cm^2)
1000	0.2
1500	1.2

2.2 Condensation and Convection Heat Flux

For each fusion micro-explosion (~10 Hz), ions and thermomechanical stresses from heat loads threaten to damage the reactor wall and driver optics. The loading on the wall and optics must remain sufficiently low to ensure that economic and safety constraints are met. One proposed method for decreasing the intensity of the wall loading is to fill the reactor chamber with a gas, such as Xe, at low density [35]. The gas will absorb much of the radiation and ion energy from the fusion event, and then slowly release it to the chamber wall. Unfortunately the protective gas introduces major heat loads on the target due to convection and condensation.

Previous works have investigated convection heat transfer on a direct-drive target [4,30]. The condensation of the background gas on the target is completely neglected in the work by Siegel [4]. Raffray et al [30] account for the release of latent heat with condensation, but each particle that interacts with the surface is reflected back into the flow. In this thesis the effect of condensation is fully accounted for by considering the release of latent heat upon condensation, as well as the removal of the condensed particles from the flow.

2.2.1 Estimating the Heat Flux and Number Flux

It is useful to have a simple method of calculating the heat and number flux on the target for the verification of DSMC results. An estimate of the heat flux due to condensation and convection can be obtained by applying the kinetic theory of gases. For a stream of gas traveling toward a transparent plane at the overall velocity *u*, the mass flux (kg/m^2-s) is given by [5]:

$$j = \left(\frac{M}{2\pi R} \right)^{1/2} \left[\Gamma \sigma_c \frac{P_g}{T_g^{1/2}} - \sigma_e \frac{P_f}{T_f^{1/2}} \right] \tag{2.2}$$

where M (kg/mol) is the molecular weight of the gas, R (J/K-mol) is the universal gas constant, σ_c and σ_e are the condensation and evaporation coefficients, P_g and T_g are the gas pressure (Pa) and temperature (K), and P_f and T_f are the condensed fluid/solid pressure and temperature. Γ is a weighting function based on u and the characteristic molecular velocity of the gas, a, and is given by:

$$\Gamma(\pm a) = \exp\left(-a^2\right) \pm a\pi^{\frac{1}{2}}\left[1 \pm erf\,(a)\right] \tag{2.3}$$

where

$$a = \frac{u}{\left(2RT_g/M\right)^{1/2}}. \tag{2.4}$$

This model neglects interactions between particles that have collided with the surface and those that have not. For T_f much less than the saturation temperature (T_{sat}) of the background gas, the evaporation term in Eq. 2.2 can be neglected.

The heat flux to the surface, from the stream, is given by:

$$q'' = j\left[\Delta h + \sigma_c\left(L_f + L_v\right)\right] \tag{2.5}$$

where Δh (J/kg) is the change in enthalpy of the gas, L_f (J/kg) is the latent heat of fusion, and L_v(J/kg) is the latent heat of vaporization. The number flux, j, should be calculated with σ_c = 1 in Eq. 2.2 since each particle that interacts with the surface transfers energy.

Due to the lack of data for the enthalpy of Xe from 4000 K to 20 K, cases were executed using O_2 as the working gas. The results from DSMC were in good agreement with Eq. 2.5 for O_2. The O_2 cases were also used to determine that the DSMC model does not include latent heat in the calculation of the heat flux.

While the heat flux resulting from DSMC cannot be explicitly checked with the kinetic theory, the number flux is easily checked. The number flux of particles at the surface, f, (atoms/m^{2}s) is given by:

$$f = \frac{j}{M} N_A . \tag{2.6}$$

Fig. 2.1 shows the number flux based on Eq. 2.6 for Xe at 4000 K, as a function of Xe number density, and injection velocity. The number flux increases one order of magnitude for each order of magnitude increase in Xe density.

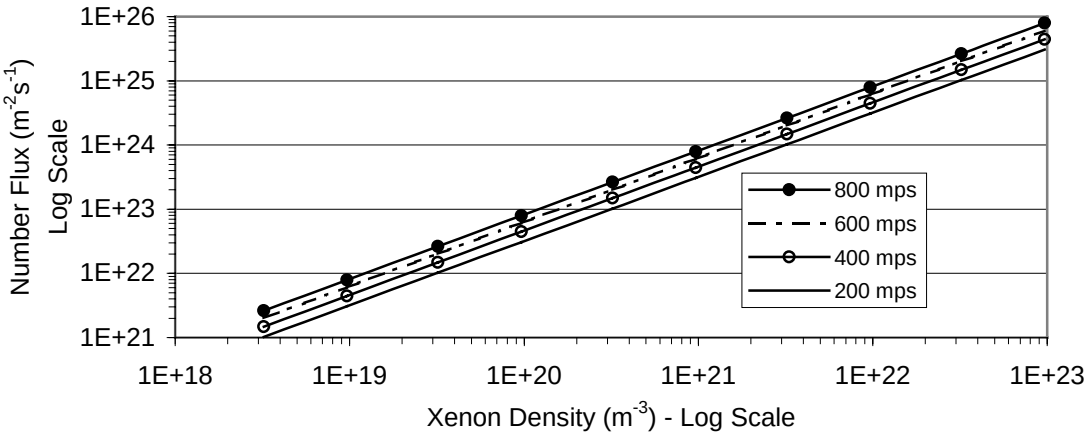

Figure 2.1. The number flux as calculated using Eq. 2.6 increases one order of magnitude for each order of magnitude increase in the gas density.

2.2.2 Condensation and Accommodation Coefficients

One of the major drawbacks of the simple kinetic theory presented in 2.2.1 is its inability to account for the interaction of molecules that have collided with a surface and are reflected into incoming stream of molecules. For flows where each of the molecules that interact with the surface are condensed, or absorbed, the results from kinetic theory should be very accurate. However, when only a portion of molecules is absorbed at the surface the reflected particles will interact with the incoming stream, changing the local density and temperature.

The fraction of molecules that condense on a surface is given by the condensation or sticking coefficient, σ_c. Note that σ_c is included in Eq. 2.5 only to account for the fact that

only the molecules that condense release latent heat. Since reflected molecules could significantly effect the flow and temperature field around the target, σ_c should be known.

The appropriate σ_c for Xe at 4000 K interacting with a cryogenic target surface is uncertain. Several studies have been aimed at determining σ_c for gases at temperatures < 2000 K, interacting with a cryogenic surface [6,7]. Since the gas temperature in an IFE reaction chamber is expected to be as high as 4000 K, data for σ_c at higher gas temperatures is needed.

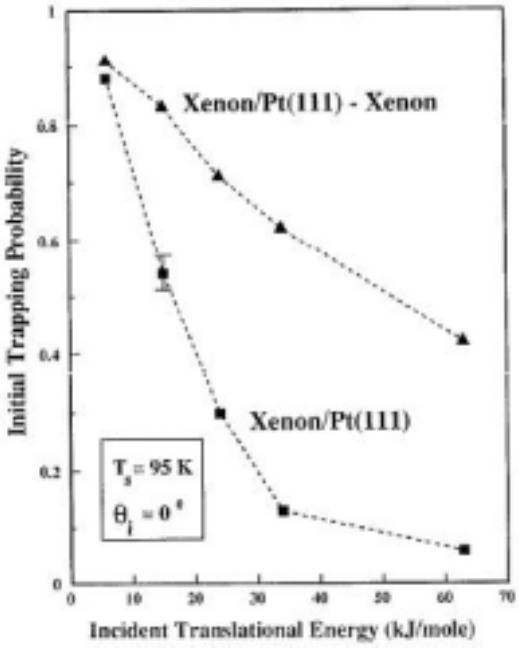

Figure 2.2. The trapping probability for Xe interacting with a Pt surface [8].

Arumainayagam et al [8] studied the condensation of Xe on a Pt surface held at 95 K. They found that the probability of a Xe molecule being trapped on the Pt surface, during its initial interaction with the surface, decreased dramatically with increasing translational energy of the Xe (see Fig. 2.2).

The average translational energy of a gas, E_T (J/mole) is given by [9]:

$$E_T = \frac{3}{2} N_A k_b T_g \tag{2.7}$$

where N_A is Avogadro's number, k_b (J/K-molecule) is Boltzmann's constant. Table 2.2 shows E_T, and the initial trapping probability (σ_I), based on Fig. 2.2, for the upper and lower limits of expected chamber gas temperature.

Table 2.2. The initial trapping probability of Xe on a clean Pt surface, and on a Xe coated Pt surface, for expected xenon translational energies [8].

Gas Temperature (K)	Translational Energy (kJ/mole)	σ_I, Clean Surface	σ_I, Xenon Coated
2000	25	0.3	0.7
4000	50	0.1	0.5

These results suggest that the *initial* condensation coefficient could be significantly less than unity for a direct-drive target injected into a chamber filled with high temperature Xe.

It is presumed that the "steady state" condensation coefficient (σ_c) would be larger than σ_I, as reflected gas would effectively decrease the energy of the incoming gas. Perhaps σ_c could also be increased for surface temperatures lower than 95 K.

For each particle that does not condense it is important to know how much of the particle's incident energy is transferred to the surface of the target. The accommodation coefficient (α) determines the amount of energy that is transferred during an interaction between a gas and a surface. For a monatomic gas α can be calculated as:

$$\alpha = \frac{\psi_{To} - \psi_{Tc}}{\psi_{To} - \psi_{Ts}} \tag{2.8}$$

where ψ_{To} is the translational energy of a molecule before interacting with a surface, ψ_{Tc} is the translational energy of a molecule after interacting with a surface, and ψ_{Ts} is the translational energy of a molecule at the surface temperature. The data for the accommodation coefficient is limited to gases with temperatures of ~ 1400 K, interacting with cryogenic surfaces [10], where α is found to be very near to unity. See Appendix K for the results of the literature search involving the sticking and accommodation coefficient.

α is not completely independent of the condensation coefficient, since $\alpha = 1$ for each particle that condenses. If σ_c is different from unity, α becomes exceedingly important since each particle that does not condense is only partially accommodated. Accounting for partial accommodation, the heat flux is calculated as:

$$q'' = j\left[\sigma_c\left(\Delta h + L_f + L_v\right) + \alpha\left(1-\sigma_c\right)\Delta h\right] \tag{2.9}$$

Assuming that there is no interaction between incoming molecules and reflected molecules the effect of σ_c and α can be seen by dividing Eq. 2.9 by Eq 2.5:

$$\frac{q''_{partial}}{q''_{full}} = \sigma_c + \alpha(1-\sigma_c)\frac{\Delta h}{\Delta h + L_f + L_v}. \qquad (2.10)$$

When $\Delta h >> L_f + L_v$ (i.e., a high gas temperature) the quotient on the right side of Eq. 2.10 is ~ 1. In this case σ_c and α must *each* be less than ~ 0.9 to decrease the heat flux by 10% or more from the fully condensing value.

When Δh is of the same order as, or less than, $L_f + L_v$ (i.e., a low gas temperature) the second term in Eq. 2.10 is significantly less than one. In this case the heat flux of the partially condensing gas ($\sigma_c < 1$) will decrease significantly with decreasing σ_c even if α is near unity.

2.2.3 Numerically Modeling the Heat and Number Flux

To determine the heat flux on a target, and to investigate the influence of condensation (σ_c), a commercial numerical code DS2V [11] (a DSMC program) was employed. The assumptions used in DS2V for modeling target injection are:

1. Axisymmetric flow around a 4mm diameter sphere (target).
2. Target surface temperature = 18 K = constant.
3. σ_c = 0 or 1.
4. α = 1 = constant.
5. Xe is the protective gas (assumed molecular diameter = 216 pm, M_{xe} = 131.29 g/mol).
6. Xe density = 3.22×10^{19} or 3.22×10^{21} atoms/m^3.
7. Xe temperature = 4000 K or 1300 K.

8. Target injection velocity = 400 m/s.

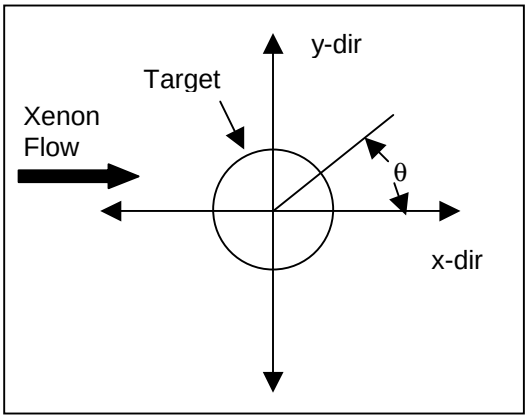

Figure 2.3. The coordinate system, flow direction, and target placement used in DS2V.

The coordinate system and placement of the target used in DS2V are shown in Fig. 2.3. Due to symmetry all of the results from DS2V are plotted for one half of the target, as a function of the angle from the trailing edge, i.e. $\theta = 0$ is the trailing edge of the target and $\theta = \pi$ is the leading edge.

Fig. 2.4 shows the number flux distribution as calculated by DS2V for the low-density cases. The results from Eq. 2.6 are plotted in Fig. 2.4 and show good agreement with DS2V at $\theta = 0, \pi$. Notice the large increase in number flux as θ is increased from 0 to π. The number flux is virtually unaffected by σ_c for the low-density cases.

Fig. 2.5 shows the number flux for the high-density cases. The number flux in Fig. 2.5 is two orders of magnitude larger than the number flux given in Fig. 2.4, when $\sigma_c = 1$, just as predicted by the simple model (Fig. 2.1). For the high-density cases Eq. 2.5 and DS2V are in good agreement at $\theta = \pi$ when $\sigma_c = 1$. At $\theta = 0$ and $\sigma_c = 1$, Eq. 2.5 predicts a number flux of approximately one-half of the value given by DS2V. For each gas temperature the number flux is nearly doubled by changing σ_c from 1 to 0. The harmony of the results from DS2V and Eq. 2.5 serve to verify the correctness of DS2V in modeling this flow.

Because DS2V does not account for latent heat when calculating heat flux, a simple calculation was done utilizing the heat and number flux calculated by DS2V in the following equation:

$$q''_{TOT} = q''_{DS2V} + n_{DS2V}(L_f + L_v) \tag{2.11}$$

where q''_{DS2V} is the heat flux and n_{DS2V} is the number flux as calculated by DS2V.

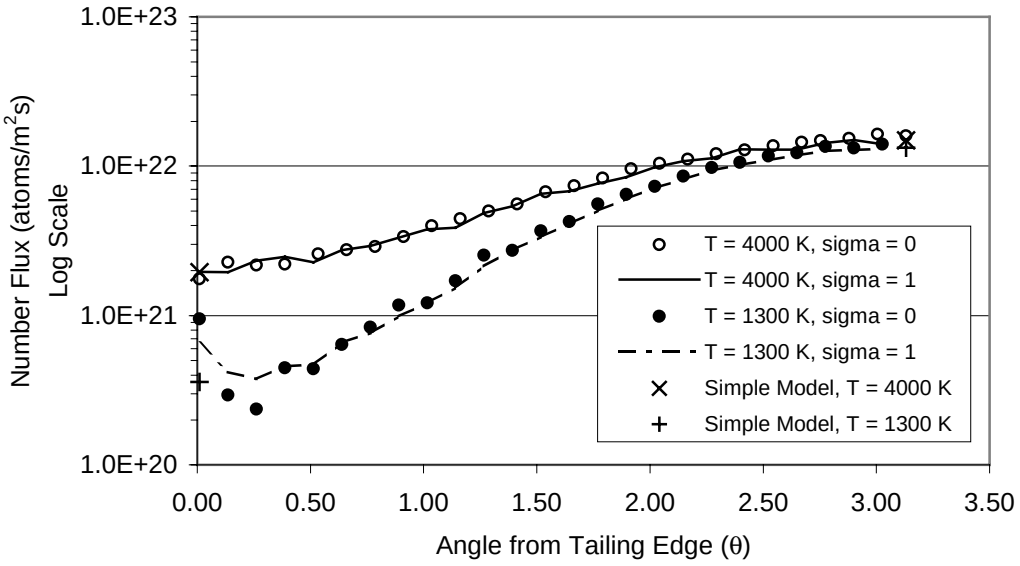

Figure 2.4. The number flux at the target surface for Xe at 3.22e19 m^{-3} is not a strong function of σ_c. Note the agreement between the simple model and DS2V.

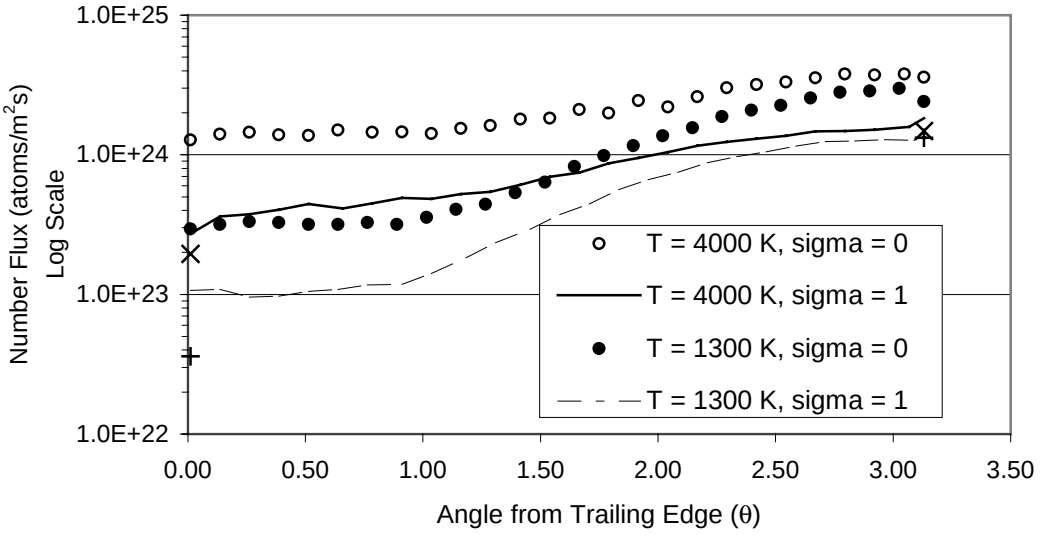

Figure 2.5. The number flux at the target surface for Xe at 3.22e21 m^{-3} is a strong function of σ_c.

Fig. 2.6 shows the heat flux for the low-density cases as given by DS2V and Eq. 2.11. Regardless σ_c, the heat flux is increased nearly 2.5 times when the temperature is increased from 1300 K to 4000 K. Increasing σ_c from 0 to 1 increases the heat flux by more than 30% when the latent heat is included using Eq. 2.11.

Fig. 2.6 shows that when the Xe temperature is low, the heat flux is unaffected by σ_c for the low-density cases unless the latent heat is included. This indicates that the interaction of reflected molecules with incoming molecules, “shielding”, is not important in this environment. For the high temperature, low-density, cases the heat flux is larger for σ_c =1 than for σ_c = 0, even when the latent heat is disregarded. Coupling this data with the knowledge that the number flux is virtually unchanged (Fig. 2.4) by σ_c, the heat flux must actually be reduced, when σ_c = 0, as result of the low temperature reflected particles decreasing the temperature of the incoming gas.

Comparison of Fig. 2.6 (low-density) and Fig. 2.7 (high-density) shows an increase of approximately two orders of magnitude in the heat flux with an increase of two orders of magnitude in density. This is in accordance with the number flux results shown in Fig. 2.1.

Fig. 2.7 (high-density) shows that increasing σ_c from 0 to 1 increases the heat flux by ~2.5 times for the 1300 K case, and ~ 2 times for the 4000 K case. Since the number flux is decreased by setting σ_c =1 (Fig. 2.5), while the heat flux is increased, the uncondensed particles are “shielding” the target by decreasing the average temperature of the gas stream, thus reducing the heat flux.

The rapid increase in heat flux with θ suggests that if the target were rotated about the y- or z-axis (Fig. 2.3), the time average maximum heat flux would be reduced. A summary of the maximum heat flux values from Fig. 2.6 and Fig. 2.7 is given in Table 2.3. When σ_c = 1 the latent heat is included.

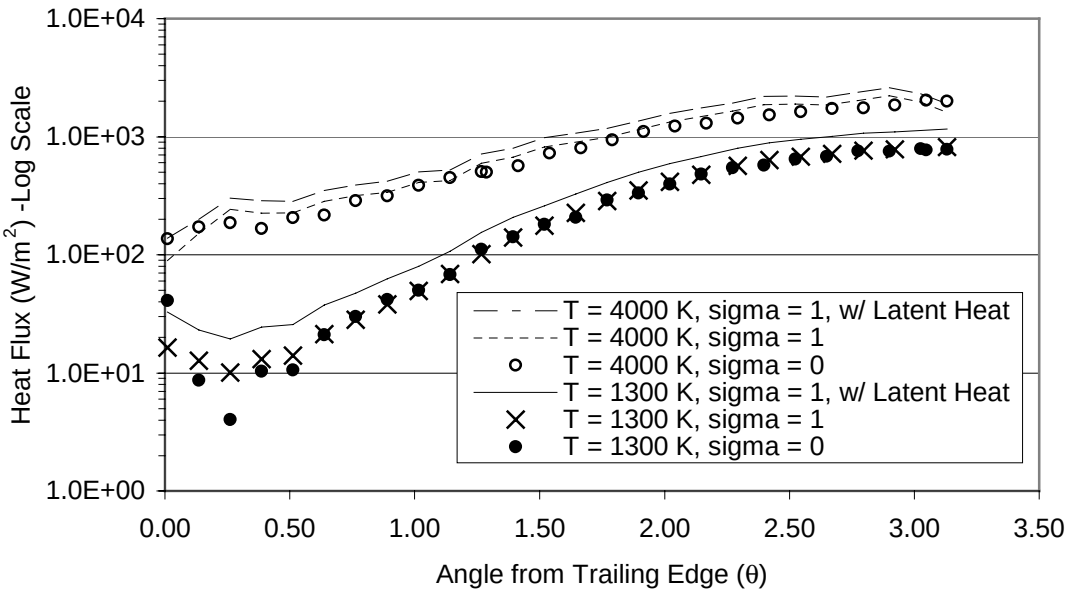

Figure 2.6. The heat flux at the surface of the target for Xe at 3.22e19 m^{-3} is clearly a function of σ_c when the latent heat is included.

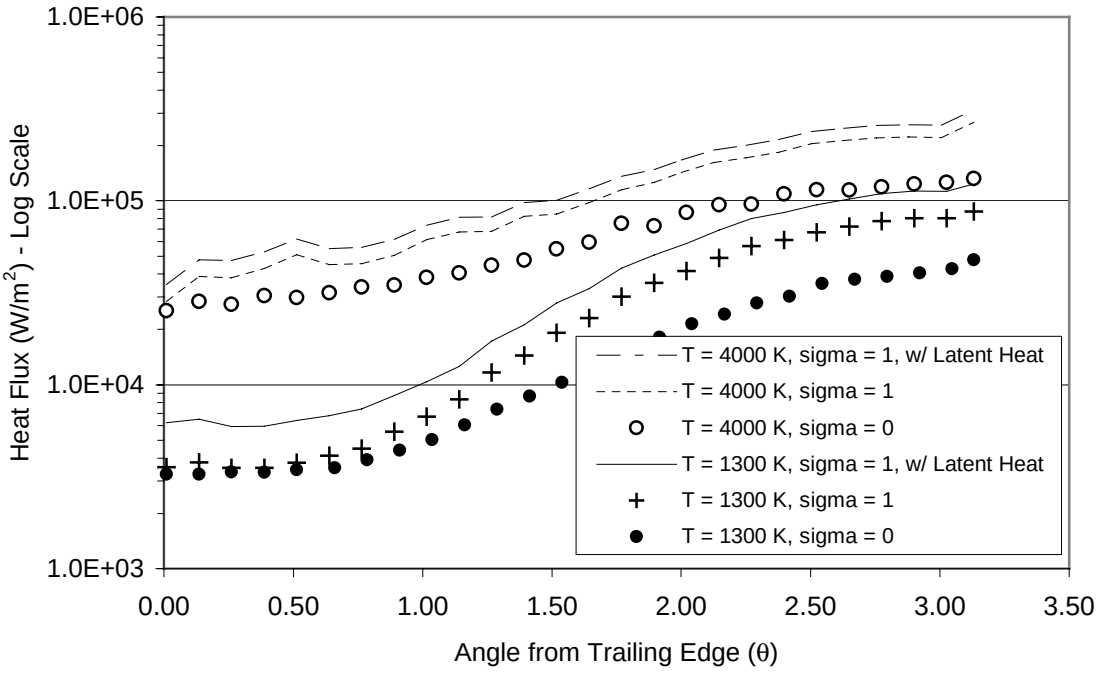

Figure 2.7. The heat flux at the surface of the target for Xe at 3.22e21 m^{-3} is strong function of σ_c.

Table 2.3. A Summary of expected maximum heat flux (W/cm^2) due to condensation and convection. The total heat flux will be increased by the presence of radiation heat transfer from the chamber walls.

	***n* = 3.22e19 m^{-3}**		***n* = 3.22e21 m^{-3}**	
σ_c	Tgas = 1300 K	Tgas = 4000 K	Tgas = 1300 K	Tgas = 4000 K
0	0.1	0.2	4.2	13.9
1	0.12	0.27	11.3	27.1

With σ_c = 1, and n = 3.22e20 m^{-3} the heat flux is estimated to be ~ 1.2 and ~ 2.7 W/cm^2 for 1300 and 4000 K gas temperatures respectively.

2.2.4 Other Considerations

The above discussion assumed that the temperature of the condensed Xe (T_f) remains near the triple point temperature of DT. However, if T_f equals or exceeds the saturation temperature T_{sat} of the protective gas, evaporation becomes significant, and the evaporating particles would interact with the gas stream.

The sublimation temperature-pressure relationship for several rare gas solids, including Xe, is given in Appendix A. Based on the expected Xe pressures, the sublimation temperature will be in the range of 79 K and 104 K. Evaporation should not be significant for a basic target since the surface temperature is not expected to reach ~ 79 K. Evaporation may be significant for an insulated target since the surface temperature could reach and exceed 100 K.

2.3 Total Heat Flux

Table 2.4 gives a summary of the maximum expected heat flux under various conditions. The radiation heat flux is taken to be a minimum of 0.2 W/cm^2 and a maximum of 1.2 W/cm^2.

Table 2.4. A Summary of total expected heat flux reported in W/cm^2.

	n = 3.22e19 m^{-3}		**n = 3.22e20 m^{-3}**		**n = 3.22e21 m^{-3}**	
σ_c	Tgas = 1300 K	Tgas = 4000 K	Tgas = 1300 K	Tgas = 4000 K	Tgas = 1300 K	Tgas = 4000 K
0	0.3 - 1.3	0.4 - 1.4			4.4 - 5.6	14.1 - 15.1
1	0.32 -1.32	0.47 - 1.47	1.40* - 2.40 *	2.9* - 3.9 *	11.5 - 15.5	27.3 - 28.3

*** Indicates interpolated values**

The values for n = 3.22e20 m^{-3} are estimated based on the trends in Fig. 2.6, Fig. 2.7, and Fig. 2.1. Table 2.4 serves as a basis for determining the potential of several target design options and the trade-off between protecting the chamber and heating the target.

Note: An additional heat load not considered in this thesis would exist if plasma conditions were present in the chamber at the time of injection.

CHAPTER 3

The Integrated Thermomechanical Model

3.1 The Need for a New Thermomechanical Model

For IFE to be successful an acceptable target must be presented at chamber center approximately 10 times per second. Previously it was assumed that the maximum DT temperature must remain below $T_{TP,DT}$ for a target to remain viable [3]. This criterion assumes that DT phase change would violate the stringent smoothness, symmetry, and/or continuity requirements placed on the target.

Many commercial software packages are suitable for modeling the temperature distribution in a direct drive target subjected to a heat flux; however, the ability to couple the mechanical response (thermal expansion, deflection due melting and vapor formation) with the thermal (heat conduction, phase change) is not readily available. Therefore, a numerical model was created that incorporated each of the important processes so that the consequences of phase change could be studied.

3.2 Simplifying Assumptions

This being the initial attempt to model the thermal and mechanical response of a direct drive target, several simplifying assumptions were made. Each assumption will be discussed in the subsections below.

3.2.1 One-dimensional Heat Transfer

The imposed heat flux calculated by DS2V shown in Fig's. 2.6 & 2.7 changes rapidly with θ. For this model it is assumed that the heat flux is uniformly distributed over the target surface, at the maximum value (Table 2.4) according to chamber conditions. This assumption allows for the determination of the minimum time to reach $T_{DT,TP}$ and/or the maximum amount of phase change.

A 1-d model restricts the ability to model asymmetries caused by non-uniform thermal loading and multidimensional heat transfer due to small vapor bubbles. See Appendix L for a discussion of the effect of vapor bubbles on heat transfer.

3.2.2 Constant DT Mechanical Properties

The deflection of the solid DT is calculated using a crude model that assumes a uniform elastic modulus. An estimate for the elastic modulus of DT (E_{DT}), as a function of temperature is given in Appendix B. E_{DT} is assumed to be constant at the initial value corresponding to the initial target temperature. The changing thickness of the DT shell due to phase change is included in the model.

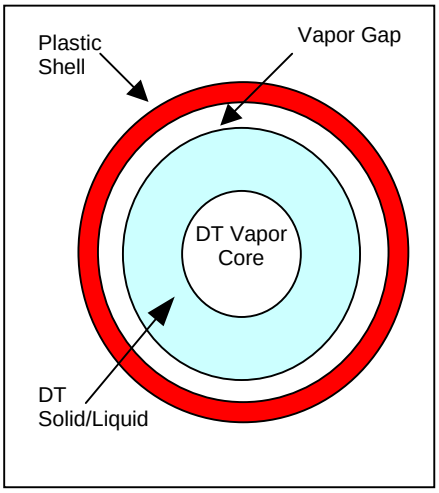

Figure 3.1. A direct drive target with a uniform vapor layer.

3.2.3 Continuous Vapor Layer

When the effect of DT vapor is studied, it is assumed that a vapor layer exists over the entire DT-polymer interface (see Fig. 3.1). This allows for a 1-d model to be constructed. The initial thickness of the vapor layer is completely determined by the

deflection of the DT solid and the polymer shell under the saturated vapor pressure at the initial target temperature.

The deflection of the thin polymer shell, subjected to a uniform internal pressure, is calculated using membrane theory as [20]:

$$\delta_{polymer} = \frac{p r_{pol}^{\,2}(1-\upsilon_{pol})}{2E_{pol}t_{pol}} \tag{3.1}$$

where p (Pa) is the uniform internal pressure, r_{pol} (m) is the radius of the polymer shell, υ_{pol} is Poisson's ratio for the polymer, E_{pol} (Pa) is the elastic modulus for the polymer, and t_{pol} (m) is the thickness of the polymer shell.

The deflection of the outer surface of a uniformly loaded thick spherical shell is given by [21]:

$$\Delta r_a = \frac{-p r_a}{E_{DT}}\left[\frac{(1-\upsilon_{DT})\left(r_b^{\,3}+2r_a^{3}\right)}{2\left(r^3-r_b^{\,3}\right)}-\upsilon_{DT}\right] \tag{3.2}$$

where r_a (m) and r_b (m) are the radii of the outer and inner surface respectively, E_{DT} is the elastic modulus for DT (Pa), and υ_{DT} is Poisson's ratio for DT.

3.2.4 Thermal Resistance of DT Vapor

When DT vapor is present it is assumed to behave as a linear thermal resistor, where heat transfer takes place only by continuum conduction through the DT vapor. Due to the low thermal conductivity of the vapor, as compared to the conductivity of the polymer and DT solid/liquid, the vapor will act as an insulator between the polymer shell and the DT solid/liquid.

In Appendix C it is shown that for vapor layers with thickness < 1 μm, the DT vapor operates in the transition or slip regime. In these regimes the thermal conductivity of the vapor

will be significantly lower than the continuum value [36]. Thus, for small gaps the model will under predict the insulating value of the vapor. For a vapor layer with a thickness of ~ 10 μm the vapor transitions from the slip regime to the continuum regime as temperature increases. Therefore, the thermal conductivity for large gaps is closely approximated by the continuum value.

3.2.5 Evaporation and Sublimation

The latent heat required for evaporation/sublimation of DT creates an apparent heat flux leaving the surface of the DT solid/liquid in the outward normal direction. This occurs since a portion of the heat transferred to the DT solid is used to evaporate/sublimate the DT. The amount of 'heat flux' depends on the mass flux and the latent heat of evaporation/sublimation for the DT. The mass flux is a non-linear function of pressure and temperature; therefore, the model must allow non-linear boundary/interface conditions.

3.2.6 DT Vapor as an Ideal Gas and the Presence of other Gases

The validity of the ideal gas assumption fades as the critical point or saturation line (see Appendix D) is approached [22]. The critical pressure of DT is, $P_{cr,DT} = 1.77$ MPa, and the critical temperature is, $T_{cr,DT} = 39.42$ K. Since the vapor pressure and temperature in a target could be at or near these critical values, the compressibility factor should be included in future models.

Helium-3 gas will be present in the target since tritium decays to helium-3. The half-life of tritium is 12.3232 years [18]. The effect of helium-3 is neglected, since the typical amount of helium-3 in a target is unknown. The presence of helium-3 will change the pressure in a vapor bubble or layer, and increase the thermal conductivity of the vapor [18].

3.3 Modeling Heat Conduction and Phase Change

To understand the response of a direct drive target to an imposed heat load, the numerical model must account for heat conduction including phase change. Many methods, of varying complexity, exist for modeling phase change [23]. A simple method for modeling phase change, called the apparent c_p method, is used in this numerical model.

3.3.1 The Heat Conduction Equation

To account for the rapid change in thermal properties at temperatures in the cryogenic region (see Appendix B), and to model solid-liquid phase change, the heat conduction equation must include variable properties. The 1-d heat conduction equation, in spherical coordinates, with variable properties, without volumetric heat generation is given (in expanded form) as:

$$\frac{\partial T}{\partial t} = \frac{1}{\rho c_p(T)}\left[\frac{\partial T}{\partial r}\left(\frac{2k}{r}+\frac{\partial k}{\partial r}\right)+k\frac{\partial^2 T}{\partial r^2}\right] \tag{3.3}$$

where T is the temperature (K), ρ is the density (kg/m^3), c_p is the heat capacity (J/kg-K), k is the thermal conductivity (W/m-K), and t is time (s).

Using the forward time central space (FTCS) finite difference method (or simple implicit) the conduction equation for a hollow sphere is discretized for node i as [23]:

$$\frac{T_i^{n+1}-T_i^n}{\Delta t} = \frac{1}{\rho c_p(T)}\left[\left(\frac{T_{i+1}^{n+1}-T_{i-1}^{n+1}}{2\Delta r}\right)\left(\frac{2k_i^{n+1}}{r_o+\Delta r(i-1)}+\frac{k_{i+1}^{n+1}-k_{i-1}^{n+1}}{2\Delta r}\right)+k_i^{n+1}\left(\frac{T_{i+1}^{n+1}-2T_i^{n+1}+T_{i-1}^{n+1}}{\Delta r^2}\right)\right] \tag{3.4}$$

where Δt is the time step, r_o is the inner radius of the hollow sphere, and Δr is the node spacing. Subscripts denote node position relative to node *i* and superscripts denote the time step. This method is second order accurate in space and first order accurate in time. It is stable for any choice of Δt and Δr.

k^{n+1} at each node is needed in Eq. 3.4; yet, T^{n+1} is unknown. To circumvent this obstacle, without resorting to iteration, k^{n+1} is extrapolated using the equation [23]:

$$k^{n+1} = k^n + \left(\frac{\partial k}{\partial T}\right)^n (T^n - T^{n-1}). \tag{3.5}$$

Similar equations are used for evaluating ρ^{n+1} and $c_p{}^{n+1}$. Note: extrapolation is certainly a source of error in the method, but it eliminates the need for iteration to obtain the correct properties, and thereby significantly reduces the computation time (see [23] for other algorithms for accounting for variable properties).

3.3.2 Boundary and Interface Conditions

The boundary condition at the outer surface of the target is assumed to be a constant heat flux. The boundary between the DT solid and the DT vapor core (Fig. 1.2) is assumed to be adiabatic.

The interface condition applied at the interface of different materials is of the form:

$$-k_a \frac{\partial T}{\partial r} = h\left(T_a^{n+1} - T_b^{n+1}\right) \tag{3.6}$$

where *h* is the heat transfer coefficient across the boundary. Eq. 3.6 is written for the boundary node of material a, where the position $r_a < r_b$. To apply this condition to the boundary node of material b, k_b must be substituted for k_a. This interface condition allows for the inclusion of a contact resistance between materials. Since the contact resistance between the various interfaces in a direct drive target has not been established, *h* is assumed to be 10,000 W/m^2-K

for each interface. One notable exception to the above interface condition exists when vapor is present between the polymer shell and the DT. This particular interface condition is discussed in subsection 3.3.4.

3.3.3 Modeling Solid to Liquid Phase Change – The Apparent c_p Method

A simple, approximate method is used to account for melting in the DT. This method is implemented by defining an apparent specific heat (c_p) for the DT. In general c_p is defined as:

$$c_p = \frac{d\eta}{dT} \tag{3.7}$$

where η is the specific enthalpy (J/kg), and T is the temperature (K). As with all pure substances, the enthalpy of DT jumps at $T_{TP,DT}$ (see Fig. 3.2a), causing Eq. 3.7 to be infinite at $T_{TP,DT}$. Bonacina [24] reported that a good engineering approximation of phase change is made by assuming that phase change takes place over a small temperature range ΔT_{pc} near $T_{TP,DT}$. Over the phase change interval ΔT_{pc}, the apparent c_p is taken as [24]:

$$c_p{}^* = \frac{L_f}{\Delta T_{pc}}\,. \tag{3.8}$$

The apparent c_p method stems from the analysis of alloys, where phase change actually occurs over a small temperature range. When applying this method to a finite difference model Bonacina [24] noted that the best results are obtained when at least 2-3 nodes are in the "melting" region (that is the node temperatures are in the range ΔT_{pc}) at each time step.

Fig. 3.2b shows that the thermal conductivity of DT (k_{DT}) also experiences a jump during solid to liquid phase change. In order for the numerical model to function properly the

thermal conductivity must also change slowly over ΔT_{pc}. The suggested function for determining k_{DT} in the phase change interval is [24]:

$$k_{DT}^{*} = k_1 + \frac{k_2 - k_1}{\Delta T_{pc}}(T - T_1) \tag{3.9}$$

where k_1 is the thermal conductivity of the solid at the lower bound of ΔT_{pc}, k_2 is the thermal conductivity of the solid at the upper bound of ΔT_{pc}, and T_1 is the lower bound of ΔT_{pc}.

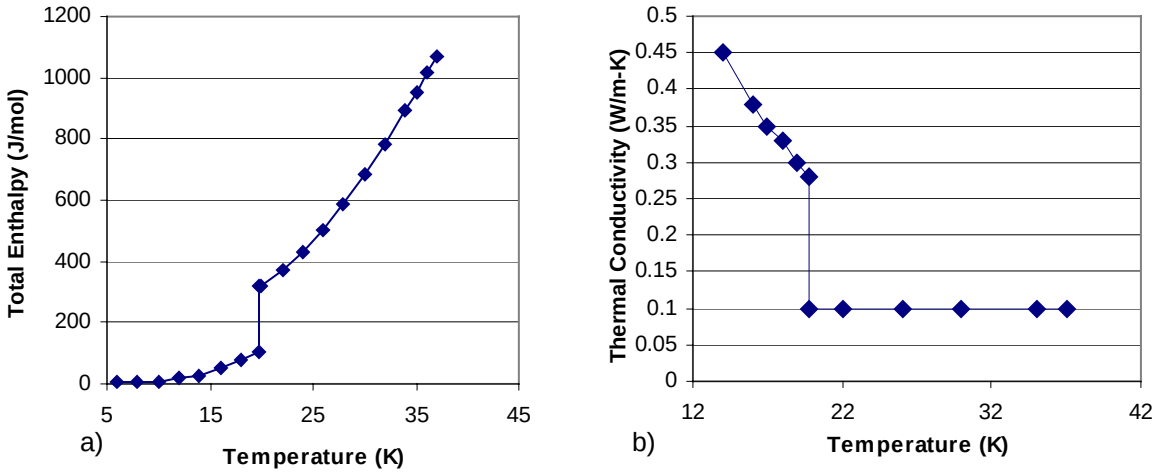

Figure 3.2a,b. The enthalpy (a) and the thermal conductivity (b) of DT as a function of temperature are discontinuous at $T_{TP,\,DT}$ [18].

3.3.4 Modeling Vaporization at the DT-Shell Interface

Recalling the assumptions of section 3.2, the vaporization of DT can be simply modeled. The equation for the net mass flux leaving a surface due to condensation and evaporation is given by:

$$j = \left(\frac{M}{2\pi R}\right)^{1/2}\left[\frac{p_{sat}}{T_{surface}^{1/2}} - \frac{p_{vap}}{T_{vap}^{1/2}}\right] \tag{3.10}$$

where p_{sat} is the saturation pressure of the DT (Pa), p_{vap} is the pressure of the DT vapor in the vapor layer, T_{vap} is the vapor temperature (K), and T_s is the temperature (K) of the DT surface where vaporization/condensation occurs.

In Appendix E it is shown that the vapor layer will be saturated (zero net mass flux) by the end of a time step, when the time step is larger than ~ 0.1 μs. The saturated vapor condition simplifies the calculation of the mass in the vapor layer at the n+1 time step to:

$$m^{n+1} = \frac{p_{sat} V}{R T_{vap}^{1/2} T_s^{1/2}} \tag{3.11}$$

where V is the volume of the vapor layer (m^3).

Eq. 3.11 allows for the simple calculation of the average mass flux over the time step n to n+1:

$$\bar{j} = \frac{m^{n+1} - m^n}{A \cdot \Delta t} = \frac{V}{A \cdot \Delta t \cdot R}\left[\left(\frac{p_{sat}}{T_s^{1/2} T_{vap}^{1/2}}\right)^{n+1} - \left(\frac{p_{sat}}{T_s^{1/2} T_{vap}^{1/2}}\right)^{n}\right] \tag{3.12}$$

where A is the area of the surface (m^2) where evaporation/condensation occurs.

The average heat flux over the time step n to n+1 due to evaporation and condensation is then given by:

$$q''_{evap} = \bar{j} \cdot L \tag{3.13}$$

where L is the latent heat (J/kg) of sublimation or evaporation of the DT.

Since the vapor is saturated at time n+1, the mass flux (Eq. 3.10) must equal zero; therefore, the pressure in the vapor layer at n+1 is given by:

$$p_{vap} = p_{sat}\left(\frac{T_{vap}^{1/2}}{T_s^{1/2}}\right). \tag{3.14}$$

The vapor pressure resulting from Eq. 3.14 is used to find the deflection of the polymer (Eq. 3.1) and DT (Eq. 3.2); hence the thickness and volume of the vapor layer at each time step.

3.3.5 The Effect of Evaporation Heat Flux

Several cases were executed to evaluate the effect of evaporation on the thermal response of the target. Comparisons were made between results from models including and neglecting evaporation heat flux. The mass flux due to evaporation, and hence pressure and thermal resistance increase, were included in each model.

The results showed that the evaporation heat flux did not significantly effect the thermal response of the target; thus, the evaporation heat flux is neglected from this point forward. For a description of the model that includes the evaporation heat flux, and a comparison of the results see Appendix F.

By neglecting the evaporation heat flux the model of the target becomes a single linear system and the computation time is decreased by approximately five times. When a vapor layer is present, but evaporation heat flux is neglected, h in Eq. 3.6 is given by:

$$h = \frac{k_{vap}}{\lambda} \tag{3.15}$$

where k_{vap} (W/m-K) is the thermal conductivity of the DT vapor, and λ (m) is the average distance between the surfaces over the time n to n+1. λ is calculated using Eq's. 3.1 & 3.2. If no vapor layer exists h is based on the contact resistance of the DT solid/liquid on the polymer.

When vapor is present it is necessary to account for λ changing over a time step to obtain an accurate value for h. An iteration scheme is used to determine the appropriate λ. For the first iteration an artificial heat transfer coefficient, h^* is calculated by assuming that $\lambda^{n+1}*$

$= \lambda^{n}$. An artificial solution is obtained for the system and λ^{n+1}** is calculated based on the artificial solution. The difference in the vapor layer thickness calculations is then obtained as:

$$\Delta\lambda = abs\left(\lambda^{n+1**} - \lambda^{n+1*}\right) \tag{3.16}$$

If $\Delta\lambda$ is less than a specified tolerance the solution is saved as permanent and the method continues to the next time step. If $\Delta\lambda$ is larger than a specified tolerance, λ^{n+1}** becomes λ^{n+1}* and the system is solved using λ^{n+1}*. This process is continued until convergence is reached. A listing of the code for the integrated thermomechanical model is found in Appendix N.

CHAPTER 4

Testing the Integrated Thermomechanical Model

The validity of the code was tested throughout its development by comparing the numerical results to results from exact solutions for simplified cases, i.e., constant thermal properties, and no phase change. The conservation of energy checked and satisfied by the code. To test the validity of the phase change model, an exact solution was derived.

4.1 Introduction

There are few analytical solutions to the solid-to-liquid phase change problem; however, some solutions for simplified geometries and boundary conditions do exist. These analytical solutions can be compared to the numerical model to test the validity of the apparent c_p method discussed in Chapter 3. To examine the performance of the present spherical model, an analytical solution for a solid sphere undergoing phase change was derived (see Appendix G for the derivation) and the results from the exact solution are compared to the numerical results below.

The exact solution (Appendix G) is obtained for a solid sphere of radius *b*, initially at a uniform temperature equal to the melting temperature of the solid, *Um*. At $t >= 0$ the surface of the sphere is raised to a temperature $Uo > Um$. Because the sphere is initially at the melting temperature of the solid, and the solid-liquid interface is an adiabatic surface, only heat conduction in the liquid region need be considered. Convection in the liquid layer is assumed to be negligible.

The melt layer as a function of time is used as a metric for determining the performance of the numerical model. The influence of the node spacing Δr, time step Δt, and

the phase change interval ΔT_{pc} are shown below. The surface temperature U_o = 25 K = constant. The lower bound of ΔT_{pc} is always $T_{DT,TP.}$

In the numerical model a node is considered liquid when the node temperature is greater than the average of the lower and upper temperatures that constitute ΔT_{pc}. Demarcating phase change at this temperature is somewhat arbitrary since the upper or lower bound of ΔT_{pc} could also be used. However, after running several cases it was seen that using the average of the upper and lower bounds for the calculation of the melt layer returned the best results.

4.2 Comparison of Exact and Numerical Results

Fig. 4.1 shows the effect of the node spacing on the melt layer calculation for a case where Δt = 1e-5 s, ΔT_{pc} = 0.4 K. Notice that even for large Δr the melt layer is approximated quite well at the time just before the next node changes phase.

Fig. 4.2 shows the effect of the time step, Δt, for a case where Δr = 0.5 μm, and ΔT_{pc} = 0.2 K. In this case it changing the time step from 1e-5 s to 5e-6 s changes the numerical solution very little. For these parameters it appears that Δt = 1e-5 s is sufficiently small.

Fig. 4.3 shows the effect of ΔT_{pc} when Δr = 0.5 μm, and Δt = 1e-5 s. The influence of ΔT_{pc} appears to increase with time. Regardless of ΔT_{pc} the melt layer calculation is always within the resolution of 0.5 μm.

While the melt layer results suggest that the apparent c_p approach can adequately track the melt layer, this is not the only measure of accuracy for the numerical model. Another metric is the ability to model the transient temperature field. Fig. 4.4 shows the temperature field at t = 0.0015 s for a cases where Δr = 1 μm, Δt = 1e-5 s, and ΔT_{pc} = 0.4 K or 0.2 K. Decreasing ΔT_{pc} from 0.4 K to 0.2 K increases the accuracy the temperature field on in the solid phase but decreases the accuracy of the temperature field in the liquid portion.

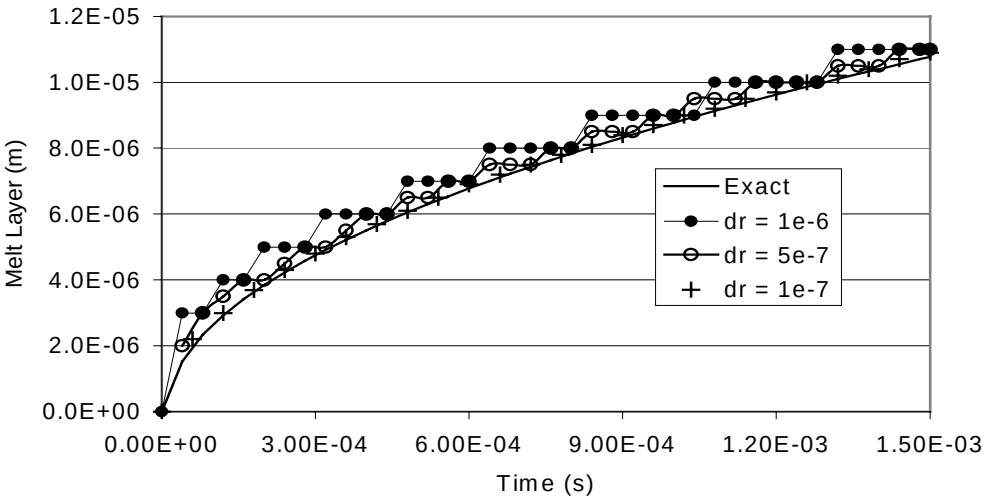

Figure 4.1. Decreasing Δr increases the accuracy of the numerical solution. Notice that for large Δr, the melt layer as a function of time is best represented at the time just before the next node changes phase.

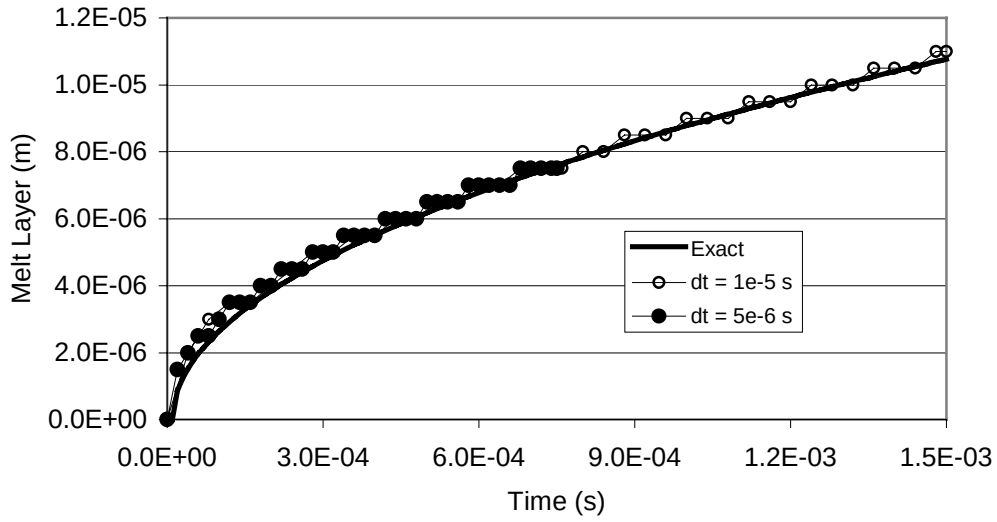

Figure 4.2. Decreasing the time step from 1e-5 s to 5e-6 s is of small consequence. This suggests that a time step of 1e-5 s is sufficiently small for the given parameters.

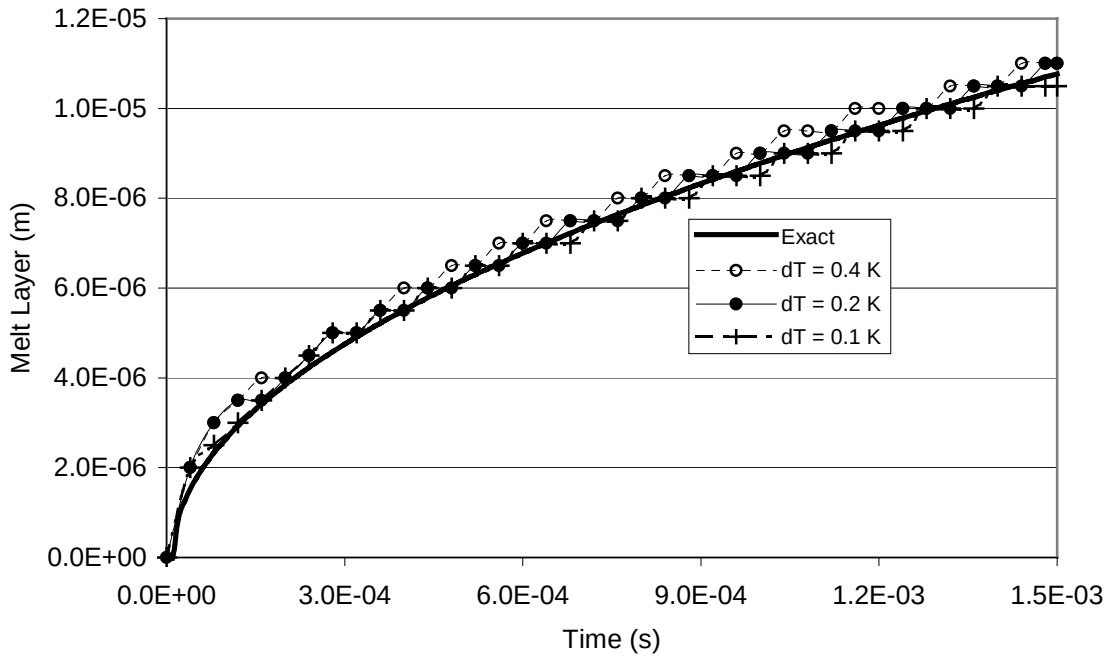

Figure 4.3. Decreasing ΔT_{pc} causes the numerical model to under predict the melt layer. However the results are always within 0.5 μm of the analytical solution.

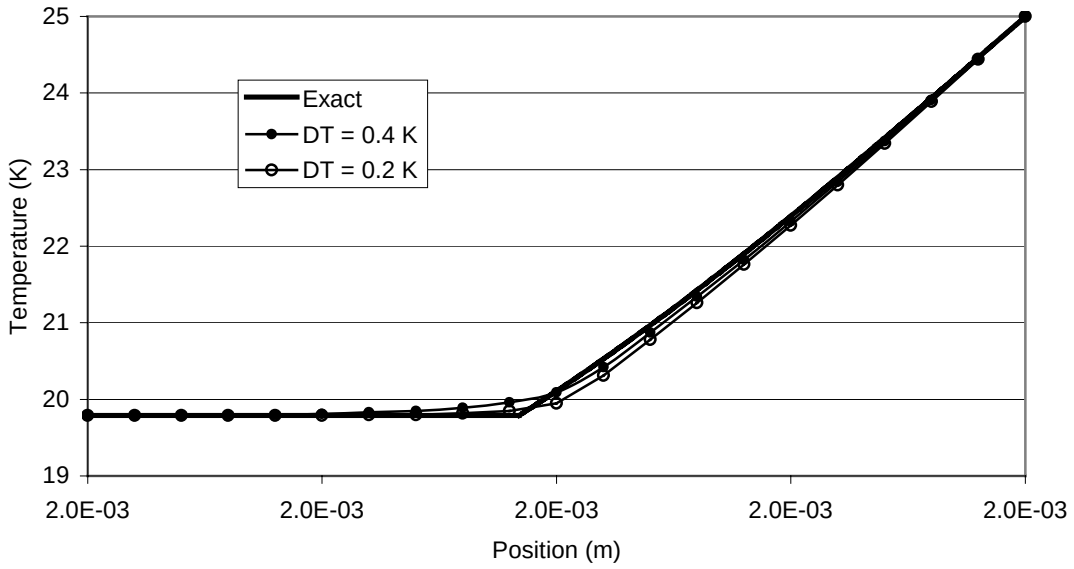

Figure 4.4. Decreasing ΔT_{pc} increases the accuracy of the temperature profile in the solid phase but decreases the accuracy in the liquid portion.

By comparing the numerical results to the analytical results for a simplified case of a melting sphere it has been shown that the numerical model is accurate. To increase the accuracy of the numerical model a different method of obtaining the properties at the n+1 time step (see Chapter 3) could be used. The temperature profile (Fig. 4.4) of the numerical model could be improved by implementing one of the more complex methods described in *Finite Difference Methods in Heat Transfer*. These methods require the tracking of the phase change front, and are thus slightly more complex.

CHAPTER 5

A Parametric Study

5.1 Introduction

By imposing a conservative criterion, that the DT temperature remains below $T_{TP,DT}$, a valuable illustration of the need for a more robust target is given. Consider a direct drive target initially at a uniform temperature of 18 K, injected into a 6.5-meter radius chamber at 400 m/s. Under these conditions, the maximum heat flux the target can be subjected to is ~ 0.6 W/cm^2. This heat flux will be achived when the density of protective gas is ~ 3.22e19 m^{-3}. This protective gas density is atleast an order of magnitude less than the projected required density [35].

Several methods have been identified that may increase the thermal robustness of a direct drive target; hence, increasing the density of protective gas and the flexibility of chamber design:

1. Decrease the initial temperature of the target.
2. Place a foam insulator on the outside of the target.
3. Allow the DT to exceed the triple point.
4. Combinations of the above options.

To investigate the potential of these design options a detailed parametric study was completed using the integrated thermomechanical model described in Chapter 3. A difficulty that is encountered in studying option 3 and 4 is the lack of acceptance criteria for determining whether a given target can be successfully imploded. Possible limitations are discussed in the sections dealing with these options.

For consistency several parameters are defined as constant.

1. The radius of the reaction chamber is 6.5 m.
2. The target is injected at 400 m/s into the chamber.
3. From (1) and (2) the time of flight of the target is 0.01625 s.

5.2 Decreasing the Initial Target Temperature

Perhaps the simplest method of increasing the robustness of a direct drive target is to decrease the initial temperature of the basic target (see Fig. 1.2). Unfortunately, as the temperature of the DT solid is decreased thermal contraction and DT surface roughness could become problematic [37].

To study the influence of the initial target temperature on a basic target (Fig. 1.2), it is assumed that the maximum DT temperature must remain below $T_{DT,TP}$. Fig. 5.1 shows the time to reach $T_{DT,TP}$, or the survival time, as a function of uniform input heat flux.

Taking the required survival time to be 0.0163 s, theses results show that decreasing the target temperature from 18 K to 16 K increases the maximum acceptable heat flux from ~ 0.6 W/cm^2 to ~ 1.5 W/cm^2. The increase in acceptable heat flux is less pronounced when transitioning form 16 K to 14 K, where the acceptable heat flux is only increased to ~ 1.9 W/cm^2.

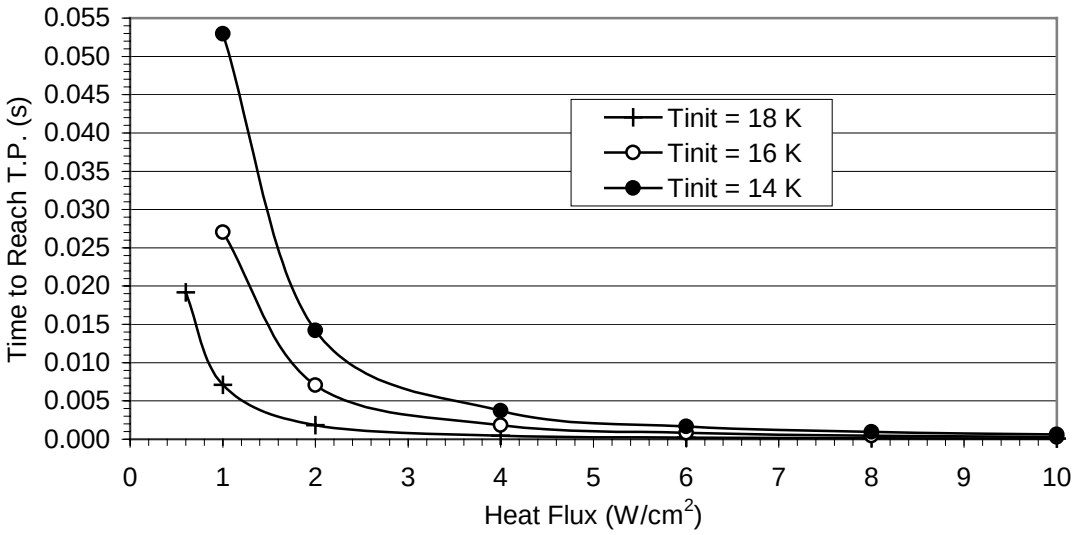

Figure 5.1. The maximum acceptable heat flux into a basic target, based on $T_{DT,TP}$, is increased significantly by lowering the initial temperature.

5.3 Insulating the Target with Porous Foam

An intuitive method for protecting the target is to insulate it with a porous foam cover (see Fig. 5.2). The thickness and porosity of the insulator could be limited by economic, implosion physics, or structural robustness considerations.

5.3.1 Insulator Configuration

Notice the presence of a *non-porous* outer polymer shell in Fig. 5.2. This outer shell serves two purposes.

1. A reflective Pd or Au coating needs to be applied at the outermost layer to decrease the absorbed radiation heat flux (see Chapter 2).

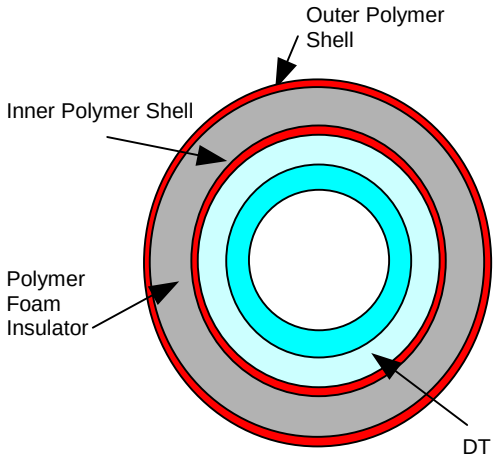

Figure 5.2. A direct-drive target with an insulating shell.

2. It provides increased heat capacity, the importance of which will be shown later.

In the numerical model, the porosity of the foam polymer is assumed to linearly transition over 10 microns, from the non-porous outer shell, to a constant porosity. A similar transition, from porous foam to non-porous shell, occurs over the 10 microns before the inner shell. A schematic of the polymer density variation is shown in Fig. 5.3. For this thesis it is assumed that the shells and the foam insulator are polystyrene. The thermal properties for polystyrene are given in Appendix B.

The base parameters for an insulated target are taken to be:

1. Inner non-porous shell thickness, t_p = 2 μm.
2. Outer shell thickness, t_o = 5 μm.
3. Initial target temperature = 16 K.

The foam insulator thickness, t_f is set to 100 or 150 μm, and the insulator density, ρ_{foam} is set to 10% or 25% of the fully dense polystyrene. The relationship between the foam density and porosity is assumed to be:

$$\rho_{foam} = \rho_{poly}\left(1-\varphi\right) \tag{5.1}$$

where ρ_{foam} (kg/m^3) is the foam density, ρ_{poly} (kg/m^3) is the density of polystyrene, and φ is the foam porosity.

It is assumed that the thermal conductivity of the polymer foam is related to the porosity of the foam by:

$$k_{foam} = (1-\varphi)k_{poly} \tag{5.2}$$

where k_{foam} (W/m-K) is the thermal conductivity of the foam, k_{poly} (W/m-K) is the thermal conductivity of the polymer, and φ is the foam porosity. The specific heat capacity (J/kg-K) of the polystyrene foam is assumed to be independent of the porosity.

5.3.2 The Effect of the Insulator Density and Thickness

$T_{TP,DT}$ is assumed to be the maximum allowable DT temperature to ensure survival. The effect of the foam insulator porosity and thickness, on the time to reach $T_{TP,DT}$, are shown in Fig. 5.4.

Increasing the insulator thickness and decreasing the insulator density (increasing in foam porosity) increases the maximum allowable heat flux for any given time to triple point (survival time). The results for a typical target without insulation, with an initial temperature of 16 K, are plotted in Fig. 5.4 for reference.

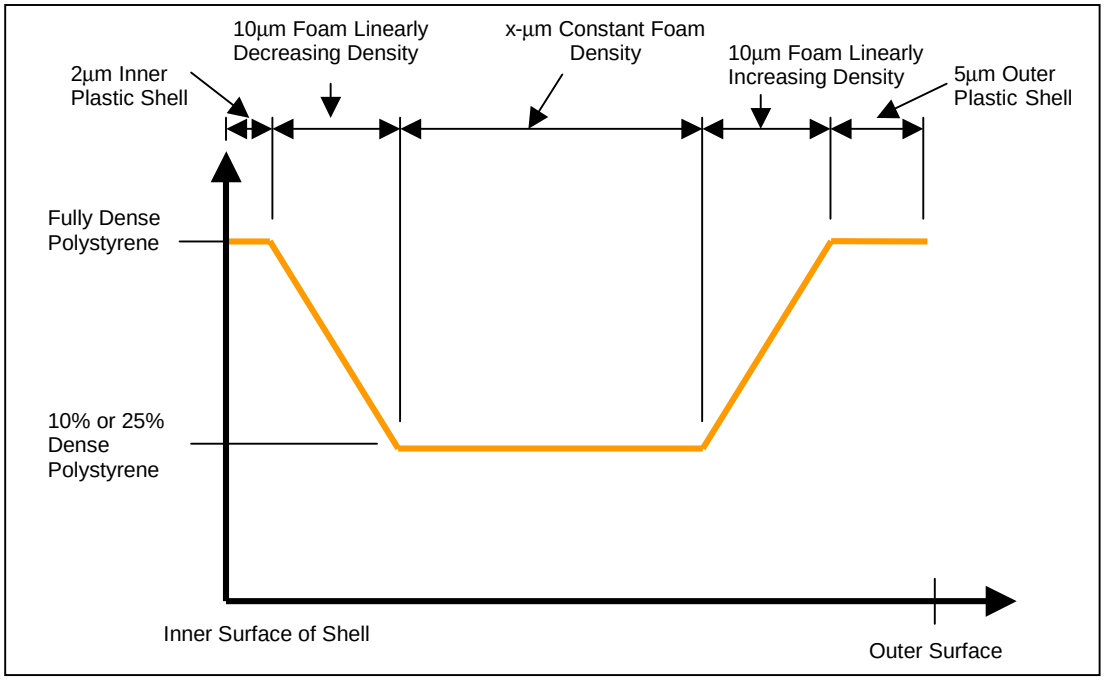

Figure 5.3. The polymer density variation as a function of position.

A comparison of the configurations will be taken at the nominal survival time of 0.0163 seconds. The maximum allowable heat flux is increased from ~ 1.5 W/cm^2 for a typical target to ~ 4 W/cm^2 for a target protected with a 100 µm, 25% dense insulator. Decreasing the insulator density to 10% increases the maximum allowable heat flux to ~ 8 W/cm^2. When the

insulation thickness is increased to 150 μm, the maximum acceptable heat flux becomes ~ 15 W/cm^2 for a 25% dense insulator, and > 18 W/cm^2 for a 10% dense insulator.

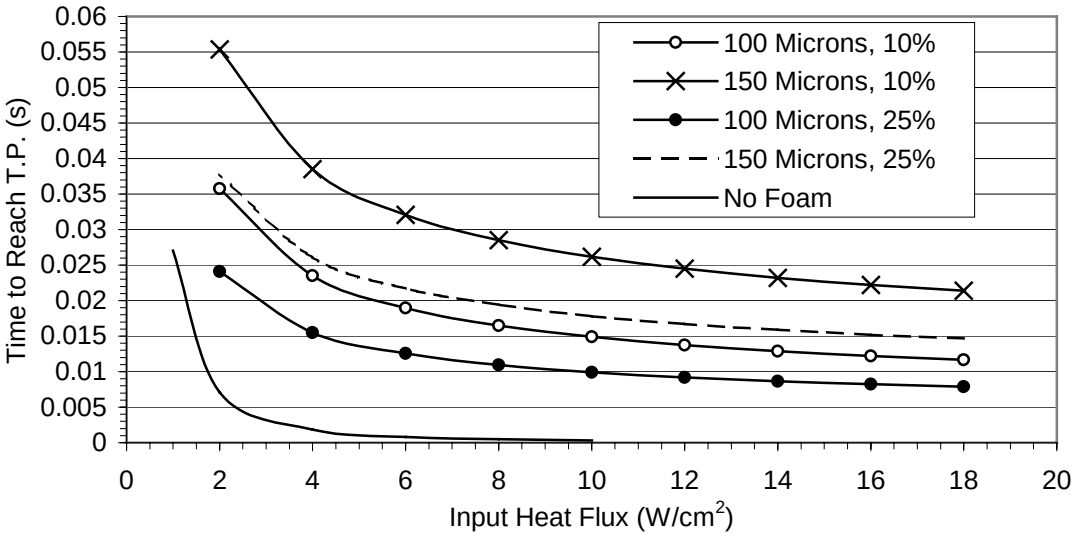

Figure 5.4. The time to reach $T_{TP,\,DT}$ as a function of insulator thickness and density. Initial Target Temperature = 16 K.

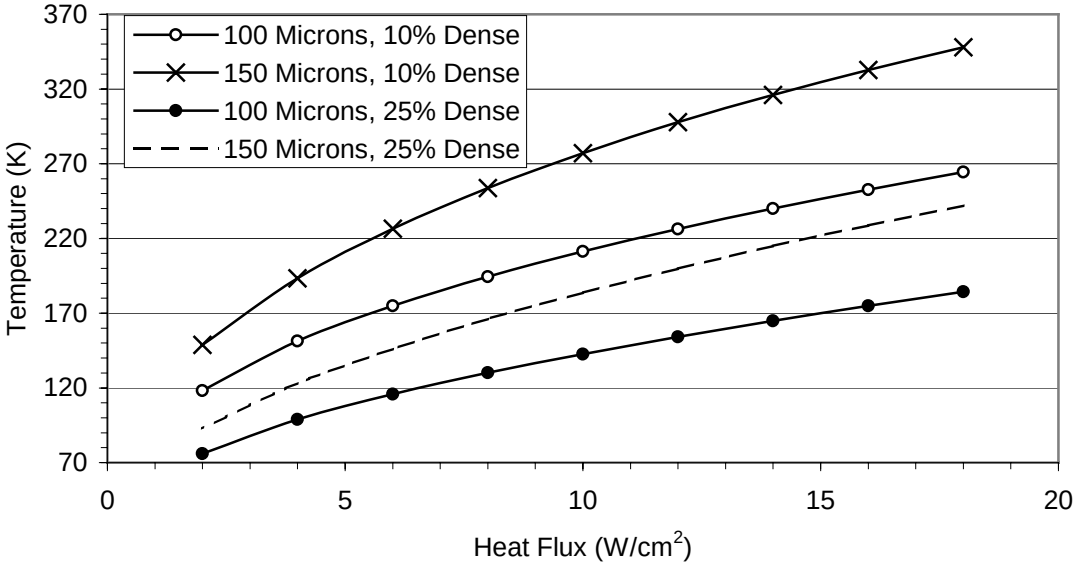

Figure 5.5. The maximum temperature in the target at the time $T_{TP,DT}$ was reached.

Fig. 5.5 shows the maximum polystyrene temperature at the time $T_{TP,DT}$ is reached (see Fig. 5.4) for the insulator configurations studied above. Based on the resulting surface temperatures (Fig. 5.5) it appears that sondensation of background gas on the surface of the insulated target will not be an issue.

5.3.3 The Effect of Outer Shell Thickness and Insulator Configuration

The influence of the outer shell thickness and the spatial foam density distribution were investigated for a 100 μm, 10% dense insulator. Fig. 5.6 shows that decreasing the outer shell thickness, from 5 μm to 1μm, decreased the maximum allowable heat flux by approximately 2 W/cm^2 at a survival time of 0.0163 s.

By eliminating the outer shell, and the spatial foam density variation, the maximum heat flux is decreased by ~ 5 W/cm^2 at a survival time of 0.0163 seconds, compared to the nominal case with a 5 μm outer shell. The change in trend seen in Fig. 5.6, when the heat flux is greater than 10 W/cm^2, is based on the assumption that the glass transition temperature of the polymer should not be exceeded.

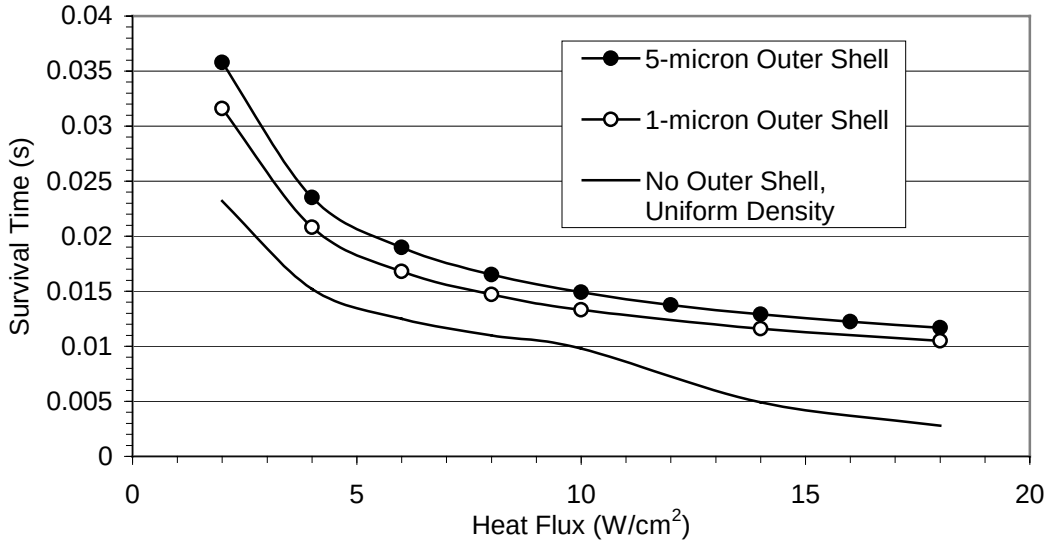

Figure 5.6. The time to reach $T_{TP,\,DT}$ or $T_{GT,P}$ as a function of insulator configuration.

5.3.4 The Effect of Decreasing Initial Temperature

Like the basic target, the initial temperature significantly influences the survival time of an insulated target. Fig. 5.7 shows the results obtained for cases with 100 μm, 10% dense insulators. At the nominal time of 0.0163 seconds, decreasing the temperature from 18 K to 16 K increases the maximum heat flux by more than 6 W/cm^2. Decreasing the initial temperature from 16 K to 14 K increases the maximum allowable heat flux by more than 7 W/cm^2.

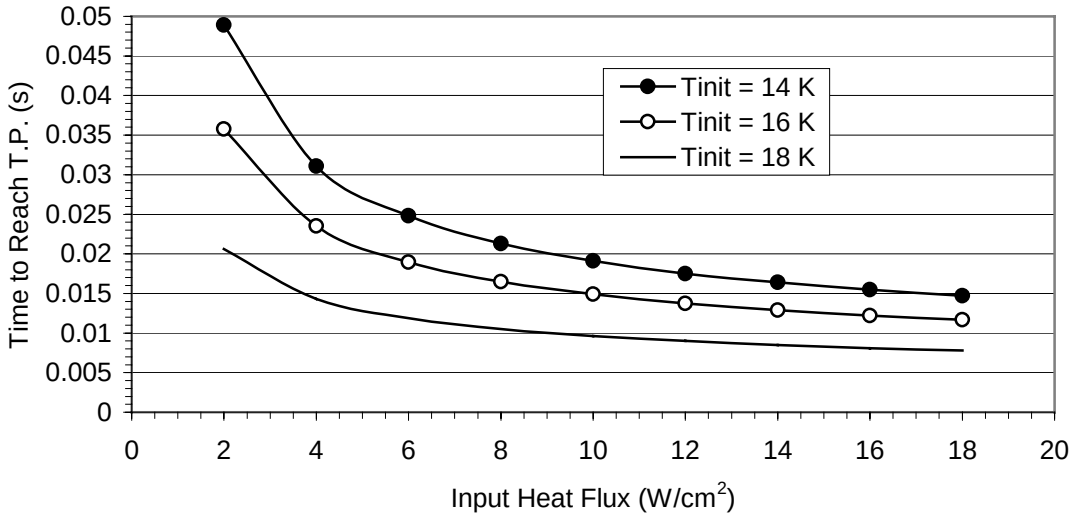

Figure 5.7. The time to $T_{TP,\,DT}$ as a function of initial temperature for a target with a 100-micron, 10% dense insulator.

5.4 Allowing Phase Change

Requiring the DT temperature to remain below $T_{TP,DT}$ is based on the assumption that any DT phase change will result in the infraction of target smoothness, symmetry, or uniform density requirements. A major motivation for creating the integrated thermomechanical model was to study the ramifications of exceeding the DT triple point.

The density changes associated with melting and vaporization could violate target symmetry, continuity, or smoothness requirements. Yet, the large difference in density between DT vapor and DT solid/liquid makes vapor formation a seemingly greater threat to target survival.

Several modes of vapor production can occur depending on the conditions. The modes of vapor growth that could occur inside of a target are homogeneous and heterogeneous nucleation. A detailed discussion of vapor growth modes is found in the *Handbook of Phase Change: Boiling and Condensation* [38].

Here, it is sufficient to note that heterogeneous nucleation occurs at a preexisting vapor filled nucleation sites. In Appendix H it is shown that the critical radius, or the minimum radius that a vapor cavity or nuclei must be before growth can occur is ~ 0.5 μm for liquid temperatures near 19 K. The intimate contact between the solid DT and the polymer shell due to the layering process, coupled with the smoothness of the polymer shell, virtually eliminate the possibility of preexisting vapor sites of radii ~ 0.5 μm. However, the critical radius decreases rapidly with increasing liquid temperature (~ 0.1 μm at 22 K). In addition, the decay of tritium to helium-3 could fertilize the nucleation sites, as the presence of helium-3 or any dissolved gas, acts to decrease the critical radius.

Homogeneous nucleation is the spontaneous creation of vapor nuclei without the aid of preexisting nucleation sites. In *Boiling Phenomena* [27] it is shown that homogeneous nucleation occurs very slowly for temperatures less than 0.9Tc (where Tc is the critical temperature). Above 0.9Tc the creation of vapor nuclei is very rapid, the rapid increase in pressure would certainly constitute a catastrophic phase change event inside of a fragile target. The presence of helium-3 will increase the rate of spontaneous nucleation according to its concentration [27]. Since the helium-3 concentration is unknown, 0.8Tc is taken as the maximum allowable DT liquid temperature.

5.4.1 Solid to Liquid Phase Change

Specific criteria for determining the viability of a target that has undergone melting are not known. Several possible limitations are:

1. Homogeneous nucleation.
2. The ultimate strength of the polymer or DT is exceeded.
3. The melt layer thickness.

As discussed above, 0.8Tc will be taken as the maximum allowable DT liquid temperature. While the polymer shell will remain intact up to its ultimate strength, the DT solid could buckle before the ultimate strength is reached. The thickness of the acceptable melt is completely unknown.

When assuming that *only* melting occurs, it is necessary to assume that the thermal expansion of the polymer is zero. If the polymer undergoes thermal expansion, the initial rate of thermal expansion of the polymer is greater than the expansion of the DT and a gap is formed between the DT and the polymer. If this occurs the gap would be filled with DT vapor at the appropriate saturation pressure. The expansion of the polystyrene occurs even below the triple point; therefore, vapor may be present in the target even without exceeding the triple point. Targets with vapor layers are considered in subsection 5.4.2.

Fig. 5.8 shows the time to reach several possible limiting factors for a target with an initial temperature of 16 K. In this case the maximum heat flux for a survival time of 0.0163 s based on 0.8Tc is 5.2 W/cm^2. This is more than triple the heat flux than obtained using $T_{TP,DT}$ as the limit. The results for targets with initial temperatures of 14 K and 18 K are shown in Appendix I. For each initial temperature the homogeneous nucleation is the first limit to be exceeded, except when the heat flux is very low. Table 5.1 gives a summary of the maximum allowable heat flux, for the nominal survival time, using 0.8Tc as the critical parameter.

Table 5.1. The maximum allowable heat flux in to a basic target if the maximum allowable temperature is taken to be 0.8Tc.

Initial Target Temperature (K)	Maximum Allowable Heat Flux (W/cm^2)
14	5.6
16	5.2
18	5.0

Fig. 5.9 shows the melt layer as a function of heat flux at the nominal time of 0.0163 s. Arbitrarily selecting 10 μm as the limit of melt layer thickness, the maximum heat flux would be reduced to 1.6, 2.5, and 3.25 W/cm^2 for targets with initial temperatures of 18, 16, and 14 K respectively.

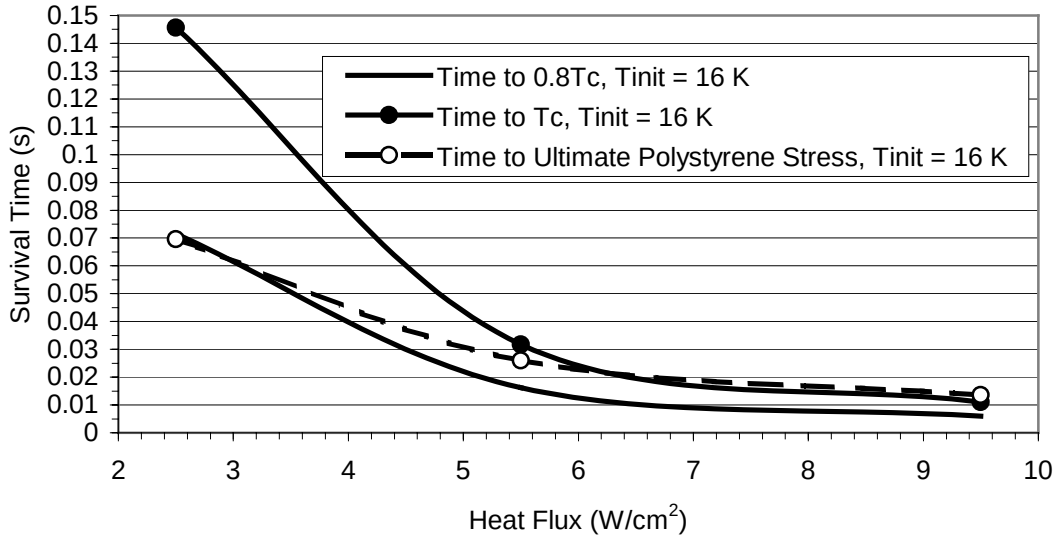

Figure 5.8. The survival of a 16 K basic target is limited by 0.8Tc for nearly all input heat fluxes.

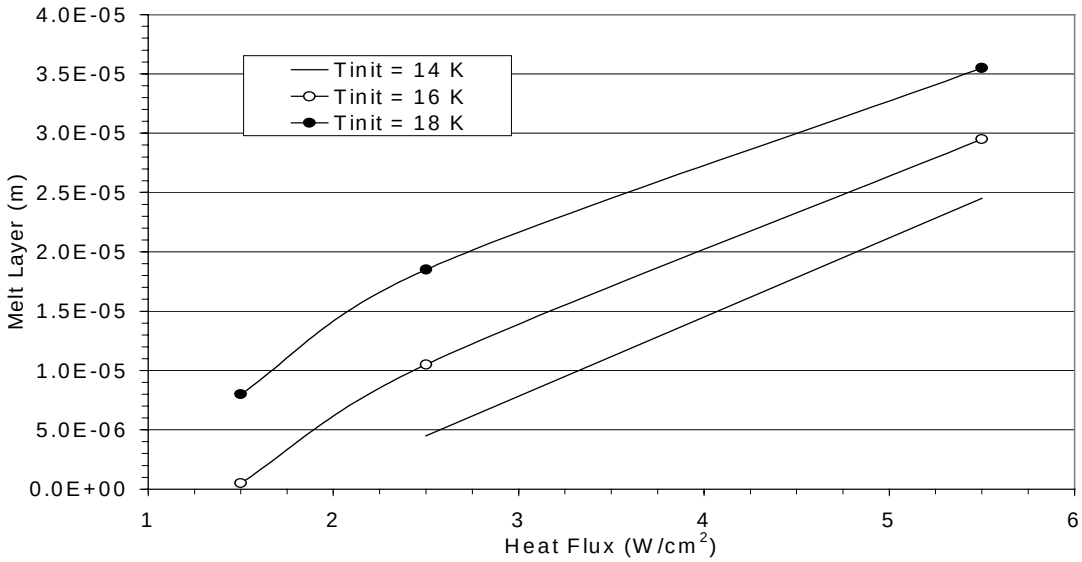

Figure 5.9. The melt layer thickness for a basic target at t = 0.01625 s as a function of heat flux and initial temperature.

5.4.2 Phase Change with Vapor

To study the influence and behavior of DT vapor it is assumed that a vapor layer initially exists between the DT solid and the polymer shell (Fig. 3.1). The thickness of this layer is determined by the deflection of the polymer shell and the DT solid due to the DT vapor pressure. The limiting criteria for this scenario could include:

1. The ultimate strength of the polymer or DT.
2. The vapor layer thickness.

Fig. 5.10 shows the time to reach the ultimate strength of the polymer as a function of heat flux, for a target with a 2-μm polymer shell. For a 2-μm shell the ultimate stress of the polymer is exceeded before the ultimate stress of the DT in every case. Based on the polymer ultimate strength, the maximum allowable heat fluxes at the nominal time are 2.1, 2.5, and 3.0 W/cm^2 for initial temperatures of 18, 16, and 14 K respectively. Note that the presence of

vapor significantly decreased the allowable heat flux compared to the cases where only melting occurs (Table 5.1).

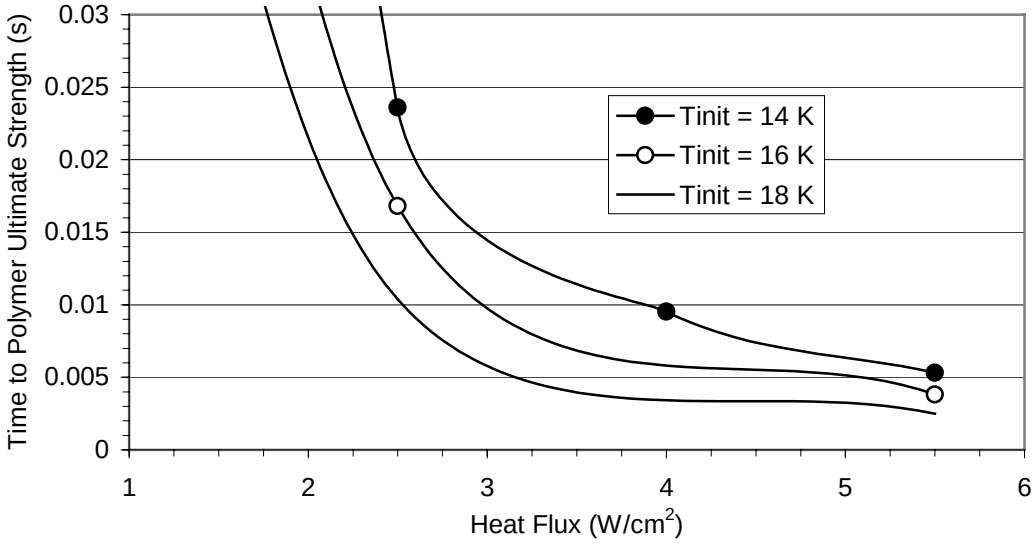

Figure 5.10. The time to reach the polymer ultimate strength.

The effect of the polymer shell thickness was investigated for a target with an initial temperature of 16 K. Fig. 5.11 shows that when the polymer shell thickness is increased from 2-μm to 5-μm the maximum allowable heat flux is increased from ~ 2.5 to 3.0 W/cm^2. When the shell thickness is increased to 10-μm the critical parameter becomes the DT ultimate strength. Notice that for low heat flux on a target with a 10-μm shell, the time to DT ultimate strength is lower than the time to polymer ultimate strength in a target with a 5-μm shell. This happens because the thick shell deflects less, leaving a smaller insulating vapor layer, thus allowing more DT melting and a subsequent decrease in DT solid thickness.

Basing target failure on the ultimate strength of the DT or polymer may be too hopeful. One must also consider the amount of vapor that is present. Fig. 5.12 shows the vapor layer thickness as a function of time for a target with an initial temperature of 18 K. The first

thing to notice is that the vapor layer is initially 2-μm thick due to the saturation pressure of DT at 18 K (see the P-T diagram for DT in Appendix D). For the high heat flux cases the vapor layer grows rapidly and the ultimate strength of the polymer is exceeded before the nominal time of 0.0163 s is achieved.

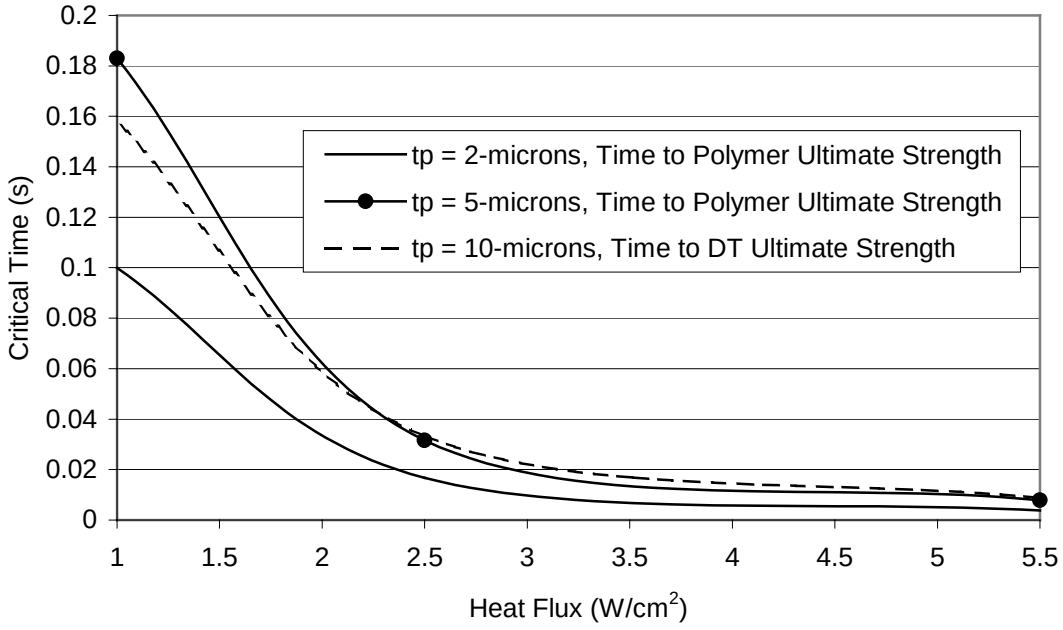

Figure 5.11. The thickness of the polystyrene shell determines whether the ultimate strength of the polystyrene or DT is exceeded first.

Figure 5.13 shows the vapor layer thickness for a target with a 2-μm shell thickness and an initial temperature of 14 K. A very interesting result occurs for this case when the input heat flux is 1 W/cm^2; the vapor layer thickness goes to zero. This result is very exciting since it suggests that vapor layers/bubbles could be eliminated or minimized under certain conditions.

Apparently this case exhibits vapor gap closure due to the low initial DT vapor pressure, and the low heat flux. Results in Appendix J, for a target with an initial temperature of 16 K, show that vapor closure can occur at this temperature if the polystyrene shell

thickness is increased to 10 μm. The results for a target with an initial temperature of 16 K and shell thickness of 2 and 5-μm are also given in Appendix J.

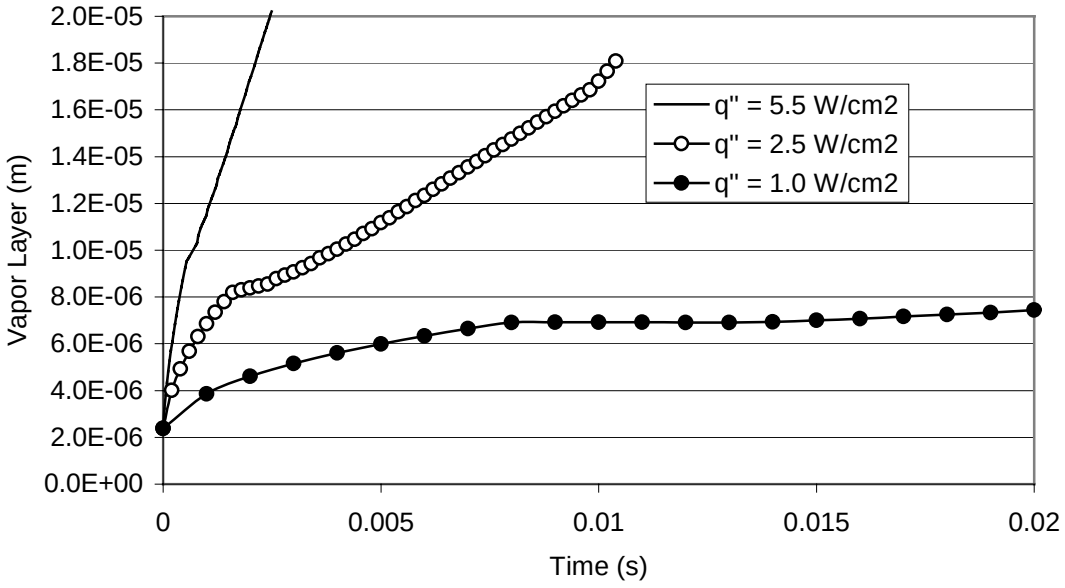

Figure 5.12. The vapor layer thickness as a function of time and heat flux for a target with an initial temperature of 18 K.

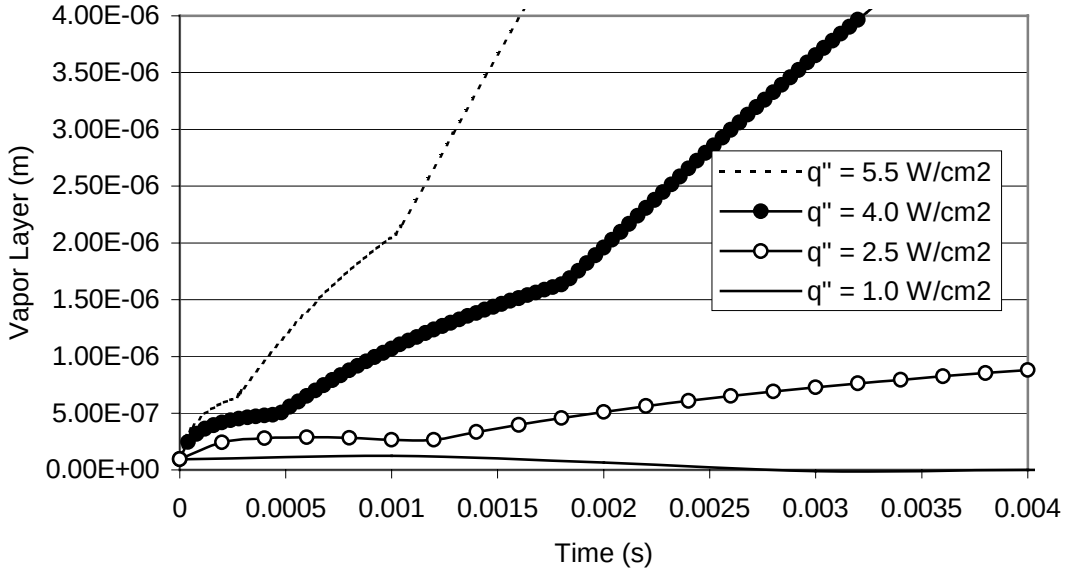

Figure 5.13. The vapor layer thickness as a function of time and heat flux for a target with an initial temperature of 14 K.

The results from the numerical parametric study were used to construct a target design plan. The details of this design plan are found Appendix M. This plan identifies the potential of various design options, idendifies critical matters that must be resolved, proposes methods for resolving the critical matters, and illustrates the coupling of the target and chamber design.

CHAPTER 6

Conclusions and Recommendations

In this chapter conclusions and recommendations for each of the target design options (decreasing the initial temperature, insulating the target, and allowing phase change) discussed in Chapter 5 are given. In addition, conclusions and recommendations regarding the chamber protective gas and its interaction with targets are presented. Based on the findings of this thesis it is concluded that a direct drive target can be designed to withstand the range of thermal loadings expected in a reaction chamber.

6.1 Decreasing the Initial Target Temperature

Decreasing the initial target temperature benefits the basic, insulated, and phase change targets (see sections 5.2, 5.3 and 5.4). For each design option the acceptable heat flux is increased. For the insulated target a lower initial temperature would translate to less insulation or more chamber protecting gas. For the phase change target the amount of phase change would decrease with decreasing temperarure and the vapor may be eliminated (see section 5.4). Thus it seems that a thorough investigation of the minimum allowable target temperature will pay off regardless of the final target design.

6.2 Allowing Phase Change

From the integrated thermomechanical model it appears that the maximum allowable heat flux (best case scenario) for a target that experiences phase change is ~ 5 W/cm^2. This translates to an allowable protective gas density of ~ $3x10^{20}$ m^{-3} = 10 mtorr @

300 K. If this is not a sufficient amount of protective gas then an insulated target or other design must be used.

While the 1-d integrated thermomenchanical model described in this thesis, has illustrated the potential of allowing phase change, it does not allow for detailed examination of the consequences of phase change. Some of the shortcomings of this model are due to its simplicity, other are due to the lack of DT properties. The following subsections detail several important recommendations for the continued study of phase change in direct drive targets.

6.2.1 DT Vapor Formation Due to Thermal Expansion

As discussed in section 5.4.1, if thermal expansion is included in the melting only model, the polystyrene shell expands faster than the solid DT and creates a layer of DT vapor. The expansion of the polymer shell is based on the assumption that there is no bond between the DT solid and the polymer. Any bond between the DT solid and polymer would impede the thermal expansion of the polymer until a sufficient stress builds at the interface.

It is recommended that a characterization of the bond between the DT solid and the polystyrene be sought. This will aid in determining the validity of assuming melting only phase change.

6.2.2 DT Bubble Formation and Growth

The results in section 5.4 suggest that vapor layers can be eliminated under certain circumstances. In reality any vapor will probably be in the form of bubbles, rather than a continuous layer. An understanding of the time dependent behavior of bubbles is needed.

In the event that DT vapor bubbles form at discrete locations in the target, an ability to predict the number, growth rate, and size of the bubbles would be essential for determining

whether a given target meets the acceptance criteria. Knowledge of the surface roughness of the polymer shell and DT solid, and the amount of helium-3 present in the DT are essential for modeling bubble growth.

It is proposed that a simplified numerical model of DT bubble growth be constructed. This model might be simplified by assuming that the presence of DT vapor bubbles does not cause significant multidimensional heat transfer, and that heat required for vaporization at the bubble interface does not significantly change the temperature field in the liquid. These assumptions allow for a numerical model to be constructed consisting of two main routines.

For each time step: the first routine would calculate the temperature field and the melt layer thickness disregarding the presence of bubbles. The second routine would use the temperature field and melt layer thickness data from the first routine to calculate the growth of a bubble(s) in an environment where the pressure is related to the bubble growth. The bubbles will grow when the liquid is superheated (the liquid pressure is less than the saturation pressure), and collapse otherwise.

Since the pressure in the liquid is dependent upon the deflection of the polymer shell and the DT, the application of a 2-d heat flux may require numerical models for the deflection of the DT and the polymer under partial loading. Collection of the correct mechanical properties for the DT is essential for an accurate bubble growth model to be constructed. Since a bond between the solid DT and the polymer would impede the flow of vapor or liquid along the DT-polymer interface, the nature of the bond must be characterized.

The fundamentals of bubble growth and collapse in an environment where the liquid pressure is related to bubble growth could also be studied experimentally using DT or a stimulant material. For the experimental setup to simulate bubble formation and growth in a direct drive target, the experiment must include solid to liquid phase change, a dissolved gas,

and at least one deformable surface to mimic the deflection of the DT solid/polymer shell. The numerical model would then be tested against this experiment. The results could be applied to understand bubble growth in direct drive targets.

6.2.3 Non-Uniform Solid to Liquid Phase Change

The integrated thermomechanical model has shown that tens of microns of melt can occur before the polymer or DT mechanically fails. For a symmetrically heated target the meltlayer thickness is uniform and the polymer and DT are uniformly loaded. In reality a 2-d heat flux will be applied at the surface of the target (see Chapter 2), which will result in a melt layer thickness that changes with position. It is recommended that a new 2-d heat transfer, and 2-d solid mechanics model be created to study the effects of non-uniform solid to liquid phase change due to a 2-d heat flux on the symmetry and continuity of the target.

6.2.4 Imploding Targets that have Undergone Phase Change

The amount and nature of acceptable phase change is uncertain. An experiment could be coupled with a numerical model to determine the effects of phase change on implosion quality. The experiment would expose targets to a uniform, or non-uniform heat flux, and then implode the target. Using the numerical model, the amount and type of phase change could be determined. The coupling of the numerical model and the data from the experiment dealing with the quality of the implosion would then allow for a better understanding of the effects of phase change. This would aid in determining the amount (if any) of allowable phase change.

6.3 Insulating Targets

Insulating the target with a foam layer appears to have enormous potential for increasing the maximum allowable heat flux on a target. An insulated target is the best option considered in this thesis if the required protective gas density is greater than ~ $3x10^{20}$ m^{-3} = 10 mtorr @ 300 K. However, several unknowns exist about the ability manufacture and implode insulated targets. In the following subsections several recommendations are made for the continued study of insulated direct drive targets.

6.3.1 Non-Uniform Thermal Expansion

An insulated target that is loaded by a 2-d heat flux could have large differences in temperature over the outer surface of the target. It is proposed that the 2-d model discussed in 6.2.3 could also be used to study the consequences of thermal expansion on an insulated target. The insulation could be limited by the asymmetry resulting from thermal expansion.

6.3.2 Structural Robustness of an Insulator

The thickness and the porosity of the insulator could be limited by the ability to withstand the acceleration of injection, and the shear stress of flight through the protective gas. The shear stress could be calculated using DS2V and an experimental/theoretical study could be done to determine the deformation of typical insulators.

6.4 The Chamber Protective Gas

It is clear from the results of Chapters 2 and 5 that the chamber protective gas density will largely determine the design of the direct drive target. However, several questions remain regarding the required protective gas density, the importance of the accommodation

and condensation (sticking) coefficients, and the consequences of condensed material on the surface of the target. The following subsections recommend: possible factors that could pose upper and lower limits on the protective gas density, future work for the determination of the accommodation and stickning coefficients, and work regarding the study of the effects of condensed material on the surface of the target.

6.4.1 The Minimum Protective Gas Density

Assuming that a gas such as Xe must be used to protect the chamber, it would be helpful to have a cost function that related the Xe density to the chamber wall life. This would allow for a determination of the cost-benefit relationship between the gas density and various target designs. It would also be beneficial to know if there is a minimum gas density, below which an unacceptable amount of chamber wall loading occurs. This density would correspond to the minimum allowable gas density, and may exclude certain target designs.

6.4.2 The Maximum Protective Gas Density

The maximum protective gas density may be limited by the ability to place a target at chamber center in an accurate and repeatable manner. DS2V could be applied to determine the drag force on a target for various gas temperatures, gas pressures, and target velocities. This data could then be used with results from simulations of the chamber environment to model the displacement of a target. Presumably there would be a maximum protective gas density, above which target placement could not be guaranteed.

6.4.3 Determining the Condensation and Accommodation Coefficients

The heat load on the target could be reduced significantly if the condensation and accommodation coefficients are determined to be less than unity. The high expected

temperatures of the protective gas would make an experimental determination of these parameters difficult. The determination of the condensation coefficient should be first, since a condensation coefficient near unity would eliminate the need to determine the accommodation coefficient.

6.4.4 Effect of Condensation on Target Surface

For a basic target (no insulation) condensation could buildup on the outer surface of the target. This buildup of condensation could pose two problems:

1. Encroachment on the smoothness or symmetry requirements of the target.
2. Decreasing the reflectivity of the surface of the target.

It seems reasonable that the same experimental setup could study each of these issues. The density of the protective gas could be limited by condensation buildup, or a decrease in reflectivity. If Ne rather than Xe were used as the chamber protecting gas, the surface temperature of the target would likely exceed the sublimation temperature of Ne (see Appendix A). This would decrease or eliminate the condensate found on the surface of the target.

6.5 Delivering a Viable Target

Using the integrated thermomechanical model, in conjuction with the thermal loading data from DS2V, it has been shown that viable direct drive targets can be designed. Ultimately the design of the target may be decided by the required amount of protective chamber gas. If the chamber can be protected with a small amount (~3.22e19 m^{-3} = 1 mtorr @ 300 K), or no protective gas at all, then a basic target would be the clear design choice based on simplicity. If the required protective density is found to be ~3.22e20 m^{-3} = 10 mtorr @ 300 K, then allowing a basic target to undergo phase change might be an

acceptable design choice. Further investigation of the effects of phase change on the target viability must be completed before this design option is considered feasible. If the required gas density is found to be between 3.22e20 m^{-3} - 3.22e21 m^{-3} (10 mtorr – 100 mtorr @ 300 K), then an insulated target would be the only viable option considered in this thesis.

APPENDIX A

Saturation Data for Rare Gas Solids

Fig. A.1 shows the sublimation temperatures for several rare gas solids. This plot was obtained using the fitted function [12]:

$$\log_{10} P = A + B/T \tag{A.1}$$

Where P (torr) is the sublimation pressure, T (K) is the temperature, and the parameters A and B are given in Table A.1.

The pressure of the background gas can be calculated using the ideal gas law:

$$P_{gas} = nk_bT \tag{A.2}$$

where n is the number density, k_b is Boltzmann's constant, and T is the temperature.

Table A.1. Coefficients used in Eq. A.1 for various rare gases [12].

Substance	***A***	***B***
Ne	6.89224	-110.809
Ar	7.66391	-414.861
Kr	7.73270	-578.320
Xe	7.78642	-806.689

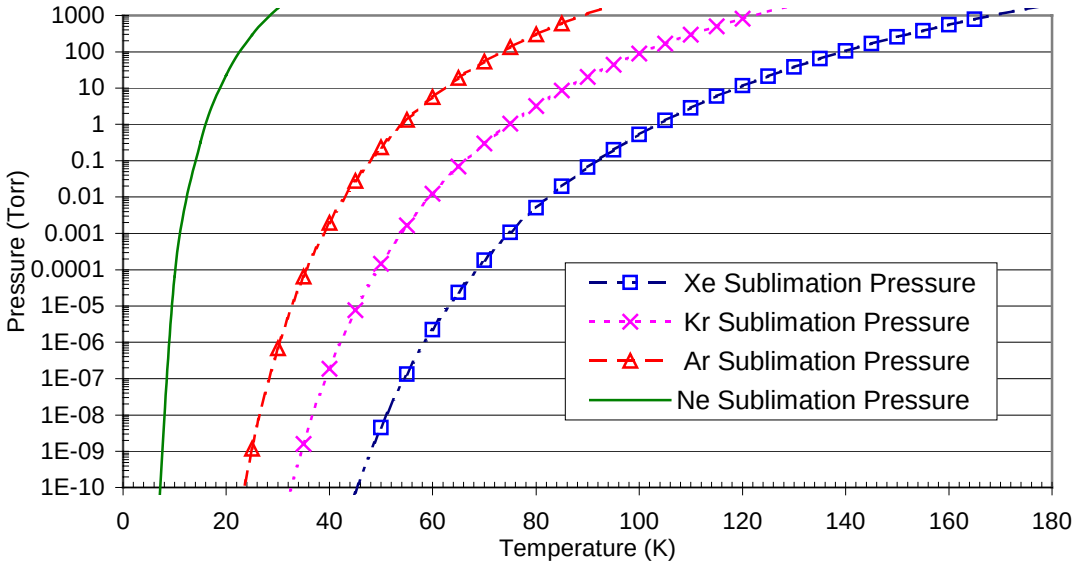

Figure A.1. The sublimation pressures for several rare gas solids [12].

APPENDIX B

Material Properties

B.1 Polystyrene Properties

Throughout this thesis it is assumed that the polymer shells and foam insulator are made of polystyrene. The thermal properties for polystyrene are shown in Tables B.1 and B.2. Note the rapid variation of thermal properties with temperature. The thermal conductivity was inferred between 4.2 K and 100 K using the data from [13] and [14] for temperatures below 4.2 K and above 100 K respectively. The following logarithmic fit resulted:

$$k = 0.0293 \cdot \ln(T) - 0.0134 \qquad \text{(B.1)}$$

where T (K) is the temperature and k is given in units of W/m-K.

Table B.1. Thermal conductivity of polystyrene [13,14].

Temperature (K)	Thermal Conductivity (W/m-K)
4.2	0.0286
10	0.0541*
20	0.0744*
40	0.0947*
60	0.1066*
80	0.1150*
100	0.1215
200	0.1418
300	0.1537
370	0.1599

* Indicates interpolated values.

Table B.2. The Specific heat of polystyrene. Data up to 100 K taken from Wunderlich [15], above 100 K from [14].

Temperature (K)	Specific Heat (J/kg-K)
10	32.18
20	102.19
50	270.15
100	460.55
200	799.68
300	1197.42
370	1842.19

Other polystyrene properties that were *assumed* to be independent of temperature are shown in Table B.3.

Table B.3. Various properties of Polystyrene.

Property	Value
Density - ρ_{Poly}	1100 kg/m^3 [16]
Ultimate Strength - Su_{Poly}	3e7 Pa
Elastic Modulus - E_{Poly}	3.4e9 Pa [16]
Coeff. of Thermal Expansion	0.222e-4 m/m/K [17]
Poisson's Ratio	0.33 [16]

B.2 DT Properties

The DT thermal properties are given in *Hydrogen Properties for Fusion Energy* [18]. A synopsis of some important thermal properties used in this study is shown in Table B.4. For a complete treatment of the properties of DT see *Hydrogen Properties for Fusion Energy* [18].

Table B.4. DT Thermal Properties [18].

Temperature (K)	**Density** (kg/m^3)	**Thermal Conductivity** (W/m-K)	**Heat Capacity** (KJ/kg-K)	**Enthalpy** (MJ/m^3)
14	257.4	0.45	1.940	1.49
16	256.06	0.38	2.269	2.61
17	255.14	0.35	3.063	3.17 (int)
18	254.04	0.33	3.540	3.86
19	252.74	0.30	3.963	4.53 (int)
19.79	251.54 *	0.29 *	4.280 *	5.13
20	221.00 **	0.10 ***	6.400	14.20
26	203.00	0.10	7.800	20.30
30	188.00	0.10	9.200	25.70
35	161.00	0.10	12.200	30.70
37	146.00	0.10	14.200	31.30

*	Indicates solid property at the triple point
**	Indicates liquid properties used thereafter
***	Following values estimated from pressurized H_2 data
int	Interpolated

The mechanical properties D2 were used to estimate the properties of DT. This estimate was made by assuming that the DT values correspond to the D2 values at a 1.1 K lower temperature [4]. This was done to reflect the 1.1 K decrease in triple point temperature of D2 compared to DT. The data from *Hydrogen Properties for Fusion Energy* [18] for the elastic modulus, and yield strength were used in the model and are shown in Table B.5. The ultimate strength (see Table B.6) was based on data in the *Handbook of Properties of Condensed Phases of Hydrogen and Oxygen* [19]. The elastic modulus and yield strength from [19] are shown in Table B.6 for reference.

Table B.5. DT Mechanical properties extrapolated from data for D2 found in [18].

Temperature (K)	Elastic Modulus (MPa)	Yield Strength (MPa)
12.7	90	0.16
16.7	80	0.08
17.5	40	0.05

The data from *Hydrogen Properties for Fusion Energy* [18] shows a rapid decrease in elastic modulus and yield strength as the triple point is approached.

Table B.6. DT Mechanical properties extrapolated from data for D2 found in [19].

Temperature (K)	Elastic Modulus (MPa)	Yield Strength (MPa)	Ultimate Strength (MPa)
13	424	0.224	0.377
15	405	0.199	0.326
17	392	0.138	0.266
18	387	0.134	0.235

The elastic modulus and yield strength from *Handbook of Properties of Condensed Phases of Hydrogen and Oxygen* [19] is much larger than the data from *Hydrogen Properties for Fusion Energy* [18] and does not decrease as rapidly as the triple point is approached. The discrepancies in elastic modulus and yield strength should be resolved by experimental work.

APPENDIX C

The Knudsen Number in a DT Vapor Layer

When studying the effect of DT vapor on the thermal and mechanical response of a target, the vapor is assumed to behave as a linear thermal resistor, where heat transfer takes place only by continuum conduction through the DT vapor. To determine if molecular effects are important the Knudsen number is calculated:

$$Kn = \frac{\varsigma}{\lambda} \tag{C.1}$$

where ζ is the mean free path of the DT vapor and λ is vapor layer thickness. ζ is taken as:

$$\varsigma = \frac{k_b T_{gas}}{\sqrt{2}\pi d_{DT}^2 p_{gas}} \tag{C.2}$$

where k_b (J/K-molecule) is Boltzmann's constant, T_{gas} (K) is the temperature of the DT vapor, p_{gas} (Pa) is DT vapor pressure, and d_{DT} is the diameter of a DT molecule and is taken to be 0.22 nm [18]. The pressure p_{gas} is assumed to be the saturation pressure corresponding to T_{gas} (See the P-T property diagram in Appendix D).

As the Knudsen number increases (see Fig. C.1) the effective thermal conductivity of the gas will decrease [36].

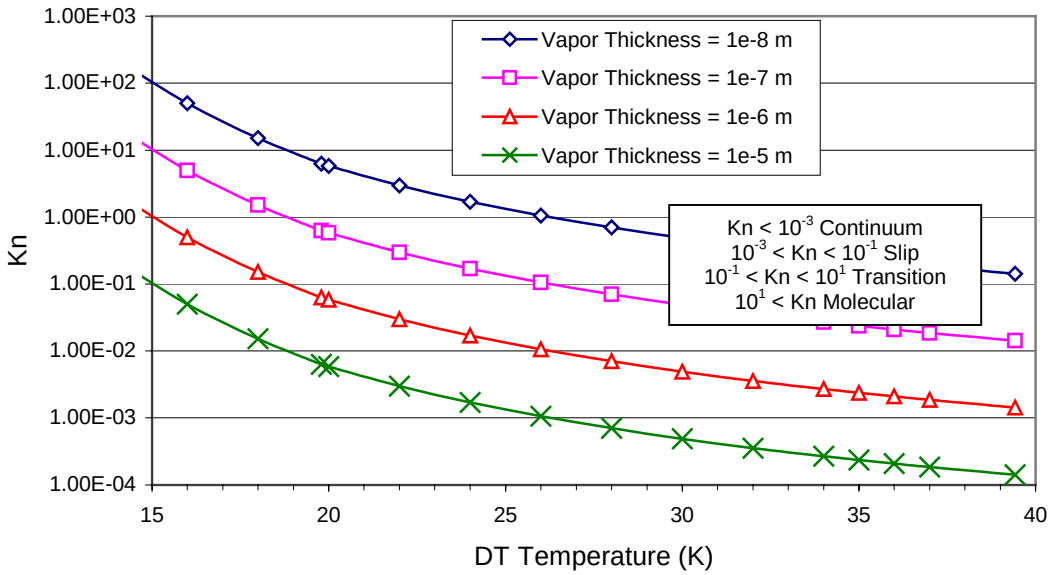

Figure C.1. Typical vapor thickness and temperature combinations could result in the DT vapor operating in any of the regimes.

APPENDIX D

P-T Property Diagram for DT

The P-T property diagram is shown in Fig. D.1. It is based on saturation data found in *Hydrogen Properties for Fusion Energy* [18].

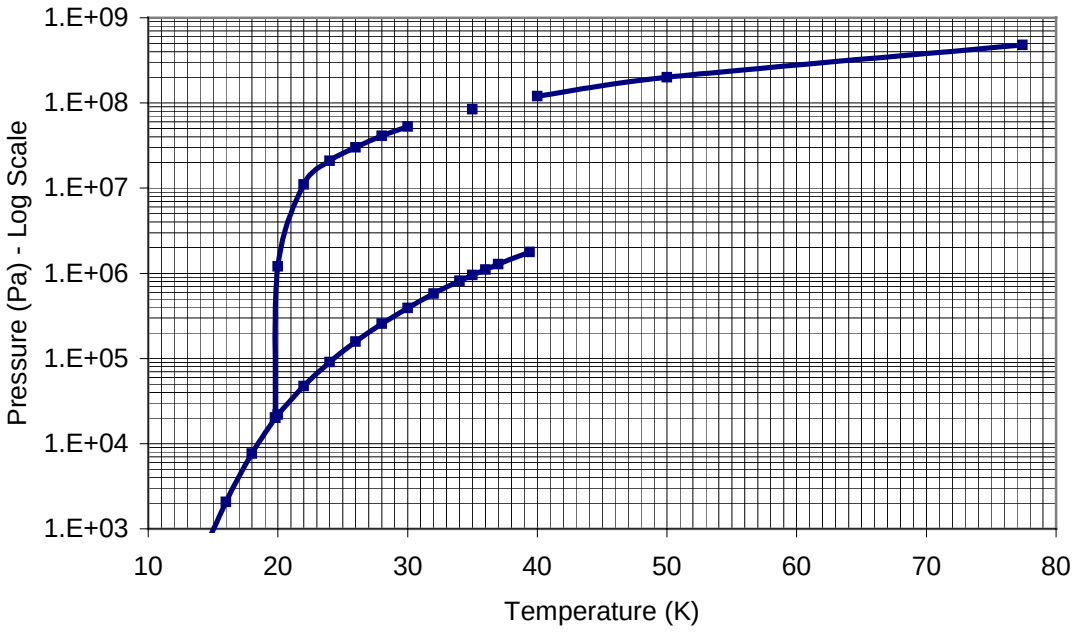

Figure D.1. The P-T property diagram for DT.

APPENDIX E

Saturation of DT Vapor in a Closed System

Recall the some of the assumptions of section 3.2 used to simplify the analysis of the vaporization of DT.

1. A vapor layer is initially present between the polymer shell and the DT.
2. The thickness of the initial vapor layer is based on the vapor pressure and the deflection of the polymer shell and the DT solid.
3. DT vapor can be modeled as an ideal gas.
4. No helium or other gases are present.

Recall Eq. 3.10 repeated here for convenience:

$$j = \left(\frac{M}{2\pi R}\right)^{1/2}\left[\frac{p_{sat}}{T_{surface}^{1/2}} - \frac{p_{vap}}{T_{vap}^{1/2}}\right]. \tag{E.1}$$

Assuming that the volume of the vapor layer is constant over a small time step Eq. E.1 is integrated in time to give the DT mass in the vapor layer at the time step n+1:

$$m^{n+1} = \underbrace{\frac{p_{sat}V}{RT_{vap}^{1/2}T_{surface}^{1/2}}}_{Equilibrium} - \underbrace{\left(\frac{p_{sat}V}{RT_{vap}^{1/2}T_{surface}^{1/2}} - m^{n}\right)}_{Change\_in\_equilibrium\_mass}\exp\left(-\triangle t\cdot\tau\right). \tag{E.2}$$

Eq. E.2 shows that the DT vapor mass, at time step n+1, is equal to the equilibrium value for the mass, minus a non-equilibrium term that includes a decaying exponential.

The time constant τ in the exponential term of Eq. E.2 is given by:

$$\tau = \frac{A}{V}\left(\frac{1}{2\pi R}\right)^{1/2}\sigma_c RT_{gas}^{1/2} \approx 10^8 \tag{E.3}$$

where A is the evaporation surface area, V is the volume of the vapor gap, R is the gas constant of DT, and T_{gas} is the maximum temperature of the DT vapor.

For the parameters appropriate for this problem it can be shown that τ is on the order of 10^8. Therefore, when Δt is larger than 0.1 μs, the product of Δt and τ is larger than 10, and the exponential term will be essentially zero. This means that the vapor equilibrium condition will occur by the time step n+1 for Δt greater than 0.1 μs.

APPENDIX F

Thermal Effect of DT Vaporization

F.1 Non-linear Boundary Conditions

The interface condition between the inner surface of the polymer shell and the vaporization surface of the DT is given by two convection condition equations. For the inner surface of the polymer shell:

$$-k_a \frac{\partial T}{\partial r} = h\left(T_a^{n+1} - T_b^{n+1}\right) \tag{F.1}$$

where k_a is the thermal conductivity of the polymer shell, h is the heat transfer coefficient based on the vapor layer, T_a is the temperature of the inner surface of the polymer shell, and T_b is the outer surface temperature of the DT solid/liquid.

If a vapor layer separates the two surfaces, h is given by:

$$h = \frac{k_{vap}}{\lambda} \tag{F.2}$$

where k_{vap} (W/m-K) is the thermal conductivity of the DT vapor, and λ (K) is the average vapor layer thickness over the time step n to n+1. If no vapor layer exists h is based on the contact resistance of the DT solid/liquid on the polymer, and is arbitrarily assigned a value of 10,000 W/m^2-K.

A similar condition is used for the surface of the DT solid/liquid:

$$-k_b \frac{\partial T}{\partial r} = h\left(T_a^{n+1} - T_b^{n+1}\right) - q''_{evap} \tag{F.3}$$

where q''_{evap} is the "apparent" heat flux due to vaporization, and k_b is the thermal conductivity of the DT solid or liquid.

In the absence of a vapor layer $q''_{evap} = 0$. When $q''_{evap} = 0$ the heat conduction equation with the appropriate boundary and interface conditions produces a linear system which is readily solved.

When a vapor layer exists between the two surfaces, and condensation or evaporation is occurring, the interface condition is complicated by three coupled factors:

1. The vapor layer thickness (λ) changes in time, thus the heat flux between the surfaces changes in time. In addition, λ at t = n+1 is not known at t = n. Note that λ is dependent on the DT vapor pressure which is a function of temperature.
2. The temperature of the inner surface of the polymer shell and the outer surface of the DT at t = n+1 are not known at t = n. Note that T_a and T_b are dependent on λ.
3. The apparent heat flux due to vaporization, $q''_{evap,}$ is a non-linear function of the temperature and pressure.

Because the apparent heat flux is non-linear, the resulting system of equations is non-linear. To solve this non-linear problem the target is separated into two portions where the vapor layer (and hence non-linear condition) defines the boundary. The first (outer) section consists of the foam insulator (if any) and polymer shell. The second (inner) section consists of the DT vapor core, DT solid and DT/Foam (Fig. 1.2).

This separation allows an artificial solution of the inner and outer sections to be obtained for artificial boundary conditions. The artificial boundary conditions are selected so

that linear systems result for the inner and outer section. The artificial boundary conditions are then adjusted using a minimization scheme to solve the non-linear problem.

For the outer section the artificial boundary condition is given as:

$$-k_a \frac{\partial T}{\partial r} = h^* \left(T_a^{n+1*} - T_b^* \right) \tag{F.4}$$

where $T_b^{n+1}*$ is a trial temperature representing the temperature at the DT interface, $h*$ is a trial heat transfer coefficient defined as the average heat transfer coefficient:

$$h^* = \frac{k_{ave}}{\lambda_{ave}} \tag{F.5}$$

where k_{ave} is the average thermal conductivity of the vapor gap over the time step, and λ_{ave} is the average vapor layer thickness over the time step.

For the first iteration k_{ave} and λ_{ave} are assumed to be equal to the values from time n. For subsequent iterations k_{ave} and λ_{ave} are the average of the time n values and the values obtained from the previous artificial conditions.

An artificial solution for the outer section is obtained by applying the artificial boundary condition (Eq. F.4). Particularly the artificial temperature $T_a^{n+1}*$ of the inner surface of the polymer shell is obtained.

Once $T_a^{n+1}*$ and $h*$ are defined, an artificial boundary condition is applied to the inner section of the form:

$$-k_b \frac{\partial T}{\partial r} = h^* \left(T_a^* - T_b^{n+1**} \right) - q_{evap}^{"*} \tag{F.6}$$

where $q"_{evap}*(T_a^{n+1}*, T_b^{n+1}*)$ is the temporary evaporation heat flux based on the artificial temperature $T_a^{n+1}*$ obtained above and the trial temperature $T_b^{n+1}*$. This boundary condition along with the conduction equation for the inner section, results in an artificial solution for the inner section of the target, namely it returns $T_b^{n+1}**$.

The difference:

$$\Delta T_b = abs\left(T_b^{n+1**} - T_b^{n+1*} \right) \tag{F.7}$$

is then minimized using Brent's method. Brent's method iterates on the above procedure changing $T_b^{n+1}*$ until ΔT_b is below a specified tolerance. When a solution is reached T_a^{n+1}, T_b^{n+1}, h , and $q"_{evap}$ are consistent.

Note that to be exact the thermal resistance should be based on the equations for thermal resistance of a sphere not a plane; however, the small gap size makes the results nearly identical.

Since each linear system of equations must be solved several times before convergence is achieved, solving this non-linear problem is computationally expensive. The expensive minimization scheme is only required due to the non-linear evaporation heat flux; therefore, the effect of the evaporation heat flux is investigated.

F.2 A Comparison Study

Using a target with a 2μm polymer shell several cases were executed to evaluate the effect of evaporation on the thermal response of the target. A thin polymer shell was used since it maximizes the evaporation heat flux by allowing rapid expansion of the polymer.

The results for cases where the evaporation heat flux was included were compared to cases where the evaporation heat flux was neglected. The mass flux due to evaporation, and hence pressure and thermal resistance increase were included in each case.

In Fig. F.1 the interface temperature histories resulting for the models with and without the vaporization heat flux are shown for an input heat flux of 9.5 W/cm^2. The similarity between the histories suggests that the vaporization heat flux does not represent a major thermal effect. When lower input heat fluxes are used the interface temperature histories are nearly identical. If the shell were constrained by due to a thick polymer shell or localized loading, the effect of evaporation heat flux would be even less.

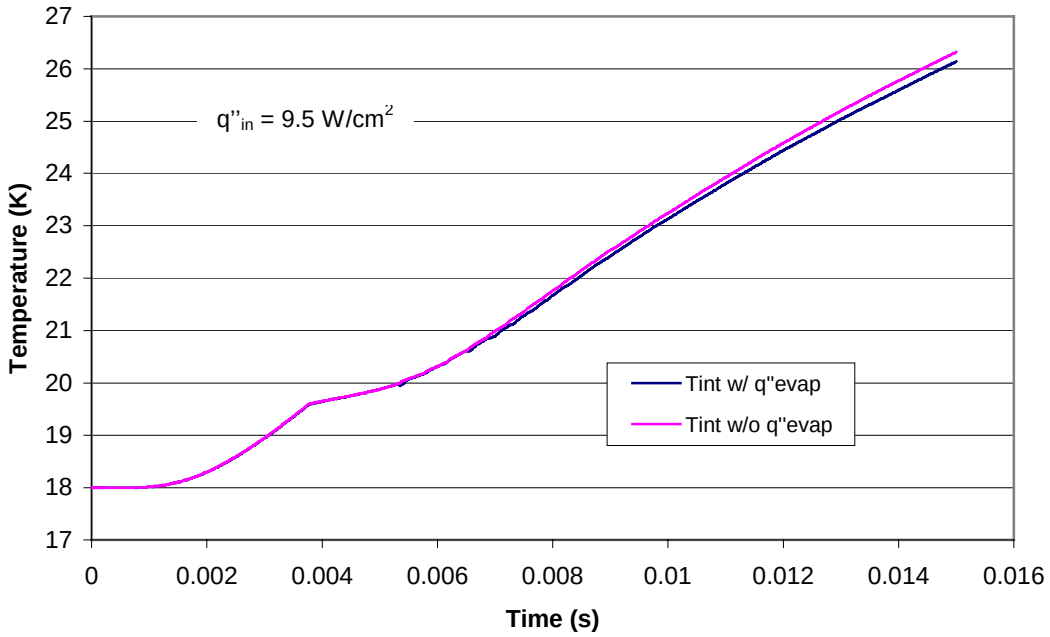

Figure F.1. The difference in the DT interface temperature histories for models with and without the evaporation heat flux is not significant for an input heat flux of 9.5 W/cm^2.

APPENDIX G

An Exact Solution for a Phase Change Problem

To obtain an analytical solution for a solid sphere the equations for the solid sphere are transformed into equations for a semi-infinite slab. The well-known solution for a semi-infinite slab is then transformed back into spherical coordinates.

To begin, the analytical solution for the temperature distribution of a solid semi-infinite slab initially at the melting temperature *Tm*, but still in the solid phase, is given by [25]:

$$T_l(x,t) = T_o + (T_m - T_o)\cdot\frac{erf\left(x/2(\kappa t)^{1/2}\right)}{erf(\gamma)} \tag{G.1}$$

where $T_l(x,t)$ is the temperature field in the liquid, T_o is the temperature to which the outer surface is raised at $t = 0$, κ is the thermal diffusivity of the liquid, t is the time since phase change began, and γ is given by the transcendental equation:

$$\gamma e^{\gamma^2} erf(\gamma) = \frac{c_p(T_o - T_m)}{L_f\sqrt{\pi}} \tag{G.2}$$

c_p is the specific heat of the liquid (assumed to be constant) and L_f is the latent heat of fusion. Using these relations the temperature profile in the slab is given as a function of space and time.

Fortunately a similar solution for a spherical geometry can be obtained by transforming the governing spherical heat conduction equations to equations for a semi-infinite slab by using the transform [25]:

$$T(x,t) = r\cdot U(r,t) \tag{G.3}$$

Eq. G.3 relates the temperature profile, *U(r,t)* of a sphere, to a temperature profile in a semi-infinite slab *T(x,t)*.

Consider a solid sphere of radius *b*, initially at a uniform temperature equal to the melting temperature of the solid, *Um*. Let *Uo* > *Um* represent the temperature to which the surface is raised to at *t >= 0*. Also, let *s* denote the position of the liquid-solid interface as a function of time. Because the sphere is initially at a uniform temperature and the solid-liquid interface is an adiabatic surface, only heat conduction in the liquid region need be considered, as no temperature gradient will exist in the solid region. Convection in the liquid layer is assumed to be negligible.

The heat conduction equation, with constant properties, in spherical coordinates is given by:

$$\frac{1}{r}\frac{\partial^2}{\partial r^2}(rU) = \frac{1}{\kappa}\frac{\partial U}{\partial t} \tag{G.4}$$

where the boundary conditions for *t >= 0* are:

$$U(b,t) = U_o \tag{G.5}$$

$$U(s,t) = U_m \tag{G.6}$$

The interface condition (the solid-liquid interface) is given by:

$$-k_l\frac{\partial U}{\partial r} = \rho L\frac{\partial s(t)}{\partial t}. \tag{G.7}$$

The thermal conductivity of the liquid is given by k_l, ρ is the liquid density, κ is the thermal diffusivity of the liquid, and L is the latent heat of fusion.

Solving Eq. G.3 for U and substituting it Eq's. G.4-7 the transformed problem is obtained:

$$\frac{\partial^2 T}{\partial r^2} = \frac{1}{\kappa}\frac{\partial T}{\partial t} \tag{G.8}$$

$$T_o(b,t) = bU_o \tag{G.9}$$

$$T_m(s,t) = sU_m \tag{G.10}$$

$$-k_l\left(\frac{\partial T}{\partial r}|_s - \frac{T}{s}\right) = \rho L s\frac{\partial s}{\partial t} \tag{G.11}$$

Define:

$$x = b - r \tag{G.12}$$

Now assume a solution of the form:

$$T(x,t) = T_0 + B\cdot erf\left[\frac{x}{2(\kappa t)^{1/2}}\right] \tag{G.13}$$

Eq. G.13 satisfies Eq. G.8 and Eq. G.9. The coefficient B in Eq. B.13 is determined such that Eq. G.10 is satisfied.

Define:

$$\gamma = \frac{x(t)}{2(\alpha t)^{1/2}} = \frac{b - s(t)}{2(\kappa t)^{1/2}} \tag{G.14}$$

Using Eq. G.13 evaluated at r = s(t), we get:

$$B = \frac{T_m - T_0}{erf(\gamma)} \tag{G.15}$$

The temperature as a function of space and time is thus given by:

$$T(x,t) = bU_o + \left(s(t)U_m - bU_o\right)\frac{erf\left[(b-r)/2(\kappa t)^{1/2}\right]}{erf[\gamma]} \tag{G.16}$$

where T_m and T_o have been transformed back into spherical coordinates. Substituting the resulting equation for $T(x,t)$ into the interface condition Eq. G.11 a relationship for γ is given by:

$$\left[\left(b - 2\gamma(\kappa t)^{1/2}\right)U_m - bU_o\right]\frac{\exp\left(-\gamma^2\right)}{erf(\gamma)} + U_m\sqrt{\pi\kappa t} + \frac{\sqrt{\pi}L}{C_p}\gamma\left(b - 2\gamma(\kappa t)^{1/2}\right) = 0 \tag{G.17}$$

Eq. G.17 is not nearly as elegant as Eq. G.2, but it is nonetheless tractable. Once results are obtained in terms of T they are transformed into U by applying Eq. G.3.

APPENDIX H

Critical Vapor Radius

The critical radius, or the minimum radius of a vapor nucleus, or nucleation site, to ensure growth, for a single component vapor nucleus in a uniform temperature liquid is given by [27]:

$$r_c = \frac{2\omega}{p_{\inf} \cdot \exp\left[\frac{\nu_l \left(p_o - p_{\inf}\right)}{RT_o}\right] - p_o} \tag{H.1}$$

where ω (N/m) is the surface tension, p_{inf} (Pa) is the equilibrium vapor pressure, ν_l (m^3/kg) is the liquid specific volume, p_o (Pa) is the pressure in the liquid, R (J/kg-K) is the gas constant, and T_o (K) is the liquid temperature.

Fig. H.1 shows the calculated critical radius for DT at liquid pressures and temperatures in the range expected for the target. At the onset of solid to liquid phase change the liquid temperature and pressure will be low. For the low temperature, low-pressure conditions that would exist shortly after solid to liquid phase change, Fig. H.1 shows that the critical radius is ~ 0.5 μm.

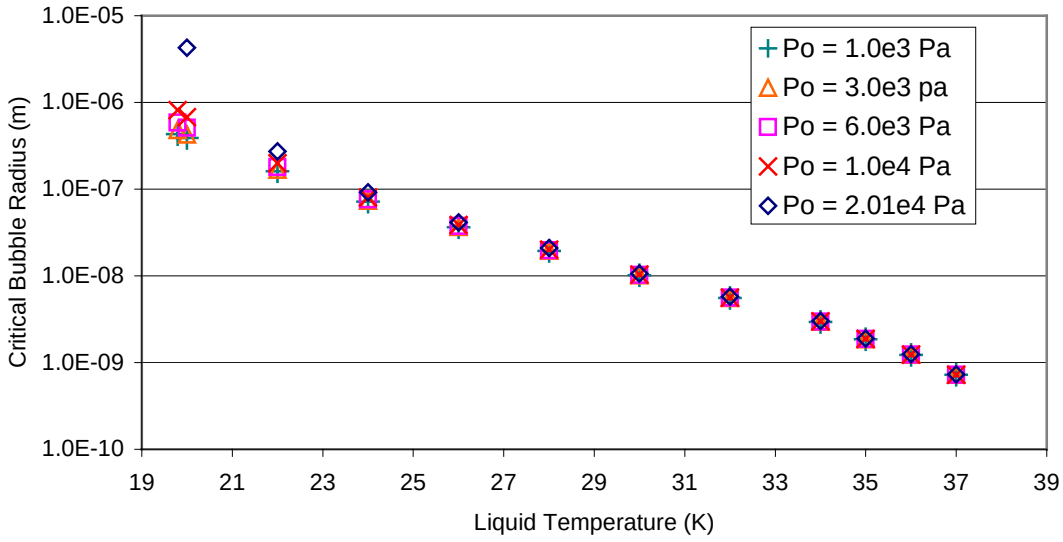

Figure H.1. The critical bubble radius for DT at pressures at temperature typical for target conditions.

The intimate contact between the solid DT and the polymer shell due to the layering process, coupled with the smoothness of the polymer shell, virtually eliminate the possibility of preexisting vapor sites of radii ~ 0.5 μm. However, the decay of tritium to helium-3 could fertilize nucleation sites, as the presence of He-3, or any dissolved gas, acts to decrease the critical radius.

APPENDIX I

Results from Melting Only Model

Fig. I.1 shows the critical time as a function of heat flux based on the DT reaching $T_{TP,\,DT}$, Tc, or the ultimate polystyrene stress for an initial temperature of 14 K. At the critical time of 0.0163 s, 0.8Tc is exceeded when the heat flux is ~ 5.6 W/cm^2. This is nearly triple the allowable heat flux based on the $T_{TP,DT}$ limit.

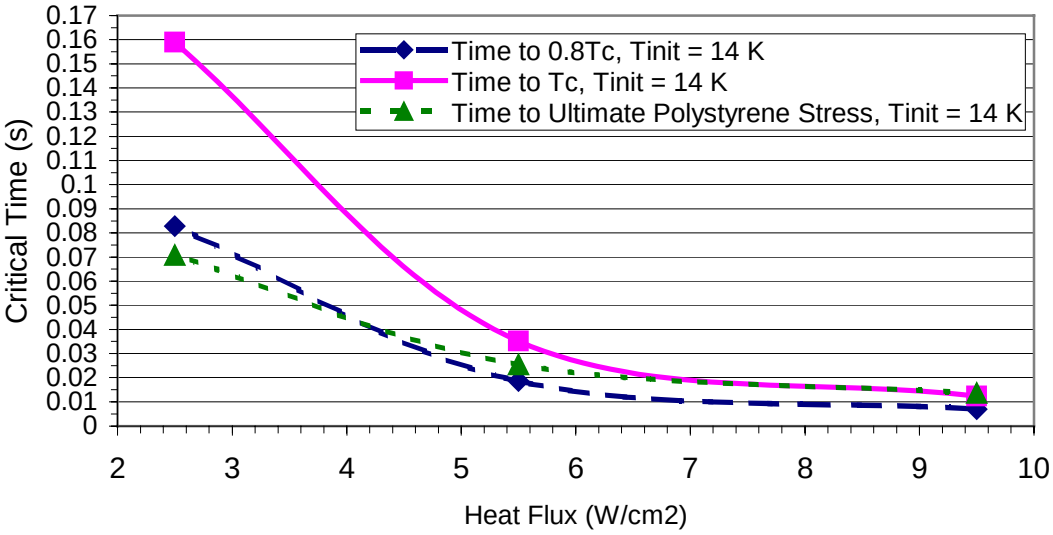

Figure I.1. The critical time is limited by 0.8Tc for nearly all heat fluxes considered.

Fig. I.2 shows the critical time for a target with an initial temperature of 18 K.

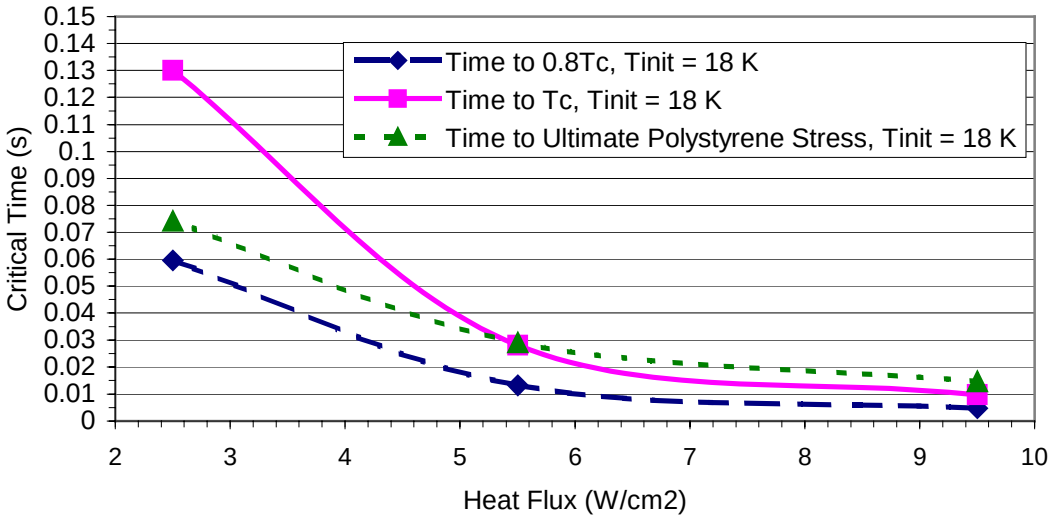

Figure I.2. The critical time is limited by 0.8Tc for all heat fluxes considered.

APPENDIX J

Results from Vapor Layer Model

Fig. J.1 shows that when the polystyrene shell thickness is increased to 10 μm the vapor layer will disappear for low heat flux.

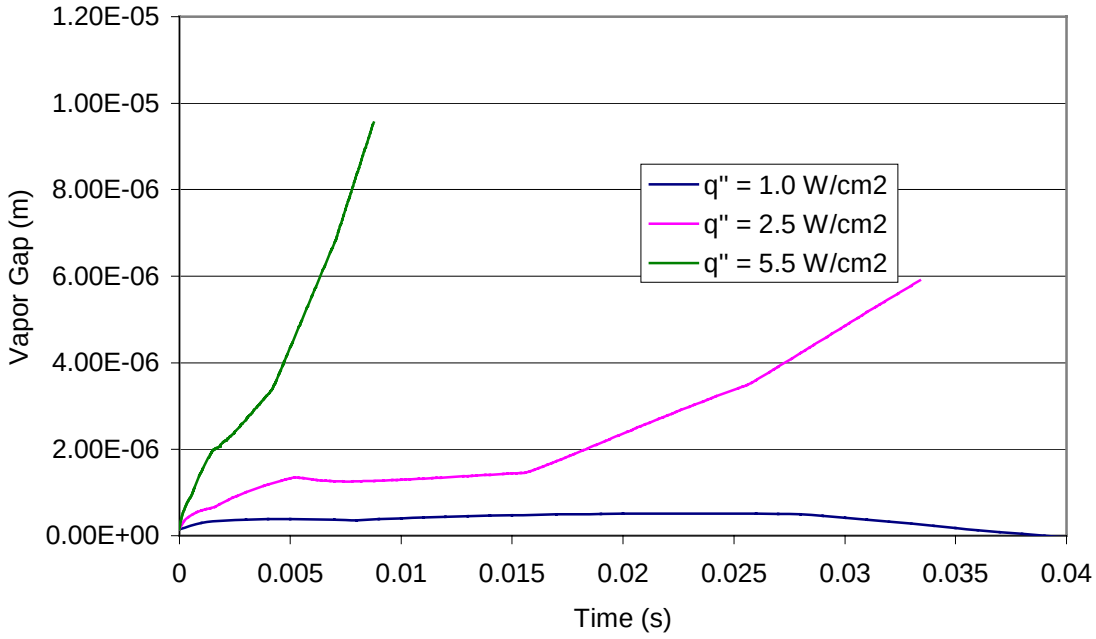

Figure J.1. The vapor layer thickness as a function of time for a target with an initial temperature of 16 K, and a polymer shell thickness of 10 μm. Notice that the vapor layer goes to zero for low heat flux after a long time.

F

Fig. J.2 shows that the vapor layer grows for each heat flux when the polystyrene shell is set to 2 μm. In Fig. J.3 it appears that the vapor layer thickness will decrease then increase when the heat flux is low for a target with a 5 μm shell. A lower heat flux would probably result in vapor layer closure.

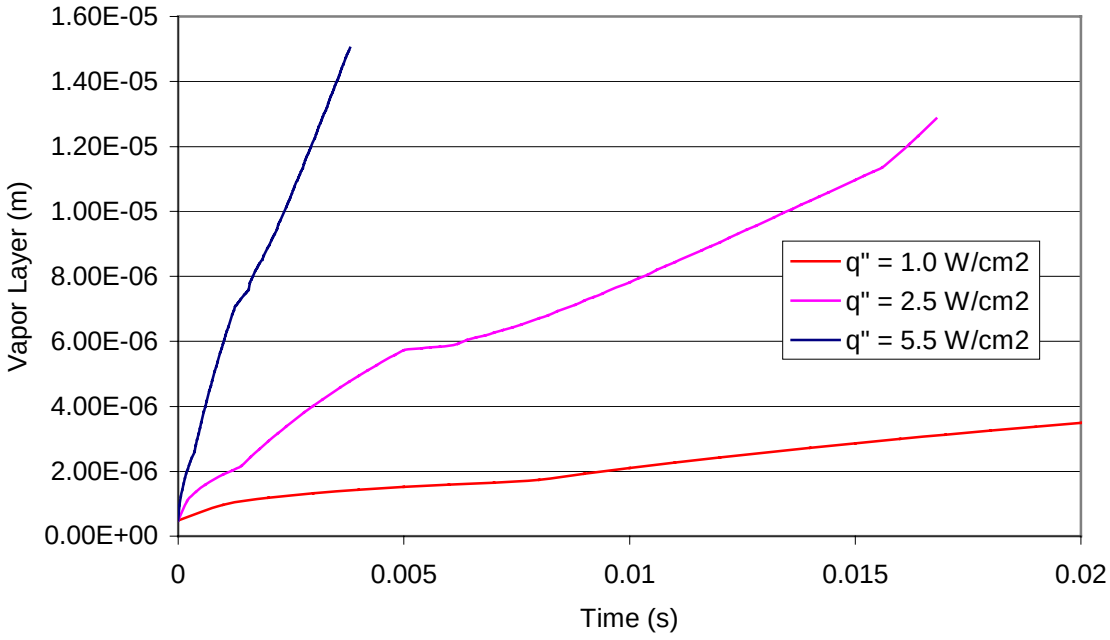

Figure J.2. The vapor layer thickness as a function of time for a target with an initial temperature of 16 K, and a polymer shell thickness of 2 μm.

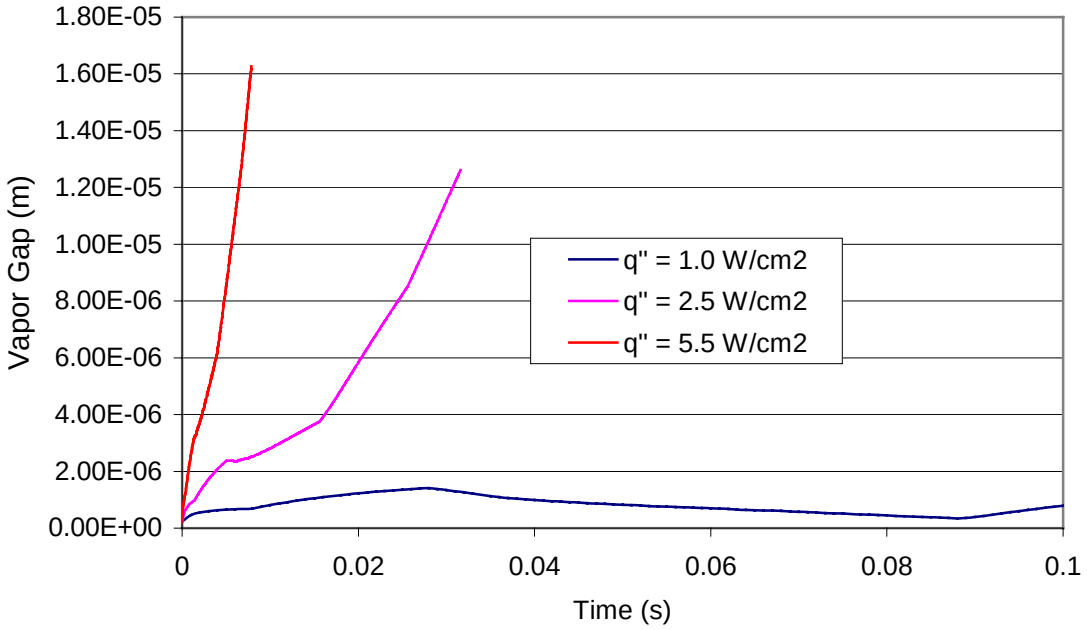

Figure J.3. The vapor layer thickness as a function of time for a target with an initial temperature of 16 K, and a polymer shell thickness of 5 μm.

APPENDIX K

Sticking and Accommodation Coefficient Literature Search

Several sources were located for the sticking (condensation) and accommodation coefficient of Xenon on various surfaces. A list of the most valuable references follows:

•Frost, W., *Heat Transfer at Low Temperatures*, 1975 Plenum Press.

•Eisenstadt, M., Condensation of Gases during Croypumping, Journal of Vacuum Science and Technology, Vol. 7, p. 479.

•Brown et. al., Condensation of 300-2500 K Gases on Surfaces at Cryogenic. Temperatures, Journal of Vacuum Science and Technology, Vol. 7, p. 241.

•Sazhin et. al., Accommodation of Tangential Momentum on Atomically Clean and Contaminated Surfaces, American Vacuum Society, Sep. 2001, p. 2499.

•Rettner et. al., Effect of Incidence Energy and Angle on the Adsorption Probability of Xe on Pt, J. Chem. Phys. 91, 1989, p. 1942.

•Madix et. al., Trapping of Ar on Well Ordered Ar, Kr, and Xe Overlayers on Pt at 30 K, Surface Science, 2000, p. 62-80.

Brown et. al. report a rapidly decreasing sticking coefficient with increasing target temperature (see Fig. K.1). The sticking coefficient also appears to be related to the number flux.

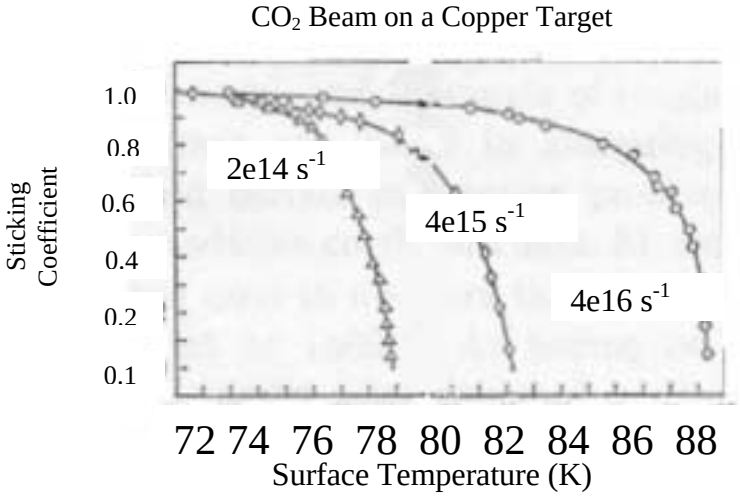

Figure K.1. The sticking coefficient is a strong function of surface temperature.

Frost reports accommodation coefficients very near unity for argon on various surfaces (see Table K.1).

Table K.1. The accommodation coefficient for a 1400 K argon beam as a function of the surface type and temperature.

Surface Description	Surface Temperature (K)	Accommodation Coefficient
Hand-polished copper	77	0.99
Hand-polished copper	276	0.97
CO2 frost on copper	77	0.99
Hand-polished copper	77	0.99
Hand-polished copper	280	0.98
CO2 frost on copper	77	0.99

APPENDIX L

2-D Heat Transfer Due to Vapor Bubbles

The following analysis was presented by A. R. Raffray at the July 2003 HAPL Target Workshop (http://aries.ucsd.edu/HAPL/).

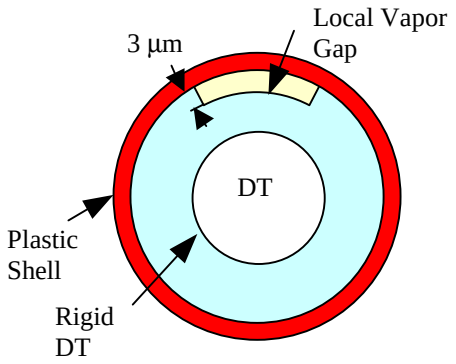

Figure L.1. The depth of the bubble was held constant at 3 μm. The arc length was varied to study the influence of bubble size.

A 2-D ANSYS model was utilized to study the effect of the bubble size on the temperature field in a target. This model neglected any thermal expansion or deflection of the polymer shell. Fig. L.1 shows the target configuration used to study the thermal effects of a vapor bubble. The bubble "depth" was held constant at 3 μm and the arc length was set to 15 μm and 50 μm. The results were then compared to a case where a 3 μm vapor layer existed over the entire target. It was determined that when the arc length was 50 μm or greater the 1-d heat transfer model would be accurate away from the edges of the bubble.

APPENDIX M

Target Design Plan

Using the results of the parametric study (Chapter 5) a target design plan was created. A major feature of this design plan is the parallel development/investigation of the basic, insulated, and phase change targets. This parallel development/investigation was selected because there are many unknowns remaining in the design options, numerical modeling can be used to evaluate design options, and it allows for flexibility in the chamber design.

The specific choice of target and chamber design can be postponed until more knowledge is obtained about the limitations of the target design options, the chamber conditions, alternate methods of chamber protection, and the costs and benefits of the acceptable target-chamber combinations. Successfully balancing the protection of the target and the protection of the chamber in an economically feasible manner is the ultimate goal.

The design plan is shown in flow chart format in Fig. M.1 to Fig. M.3. The maximum allowable heat fluxes shown for each design option are for a survival time of 0.01625 s. The design plan for the basic target is shown in Fig. M.1. Starting at the left side with the basic target, with an initial temperature of 18 K, where the triple point temperature of the DT is the failure criteria, it is seen that the maximum acceptable heat flux is ~ 0.6 W/cm^2.

The next design option on the upper path (Fig. M.1) is to decrease the initial temperature of the target. Since the minimum initial temperature is unknown, 16 K is assumed. Once agin the triple point is assumed as the failure criterion. Here the acceptable heat flux is increased to ~ 1.5 W/cm^2. However, for this design solution to work the feasibility of an initial temperature of 16 K (or lower) must be proved. This work is currently being pursued at Los Alamos National Laboratories.

The design option along the lower path (Fig. M.1) is to allow phase change to occur. In this case, the best-case scenario of only melting (no vapor formation or growth) is assumed. The failure criterion is assumed to be when the DT temperature reaches 0.8Tc. For an initial temperature of 18 K, the basic target would accept 5.2 W/cm^2 for 0.0163 s without reaching 0.8Tc. This scenario would result in a 23-μm thick liquid layer. For this design solution to be acceptable it must be shown that a target with a liquid layer (and/or vapor bubbles as the case may be) can be successfully imploded. The nucleation and growth of vapor bubbles should also be more carefully examined as this will lead to a lower acceptable heat flux. The effect asymmetric phase change, due to a 2-d input heat flux should also be considered. The proposed methods of resolution are shown on the far right.

Combining the decreased initial temperature design with the phase change design presents another design option. This option results in an acceptable heat flux (based on 0.8Tc) of 5.5 W/cm^2 with a 30-μm thick liquid layer. This option requires the resolution of the each of the issues described above.

A very similar design plan was developed for the insulated target (Fig. M.2) In this case the acceptable heat fluxes are increased dramatically, but the manufacturability, cost, and ability to successfully implode the insulated target must be resolved.

Finally Fig. M.3 is a basic diagram showing the interaction of the target and chamber design. This diagram assumes that a protective gas must be used in the chamber. Three issues could control the amount of protective gas. First, the protection of the chamber wall, this consideration will likely set a minimum on the density of the protective gas. This minimum

may be determined through numerical simulation of the fusion micro-explosion in chambers with various wall materials. Second, the ability to deliver the target accurately, as the protective gas density is increased it will become more difficult to place the target in the intended location. Using the numerical simulation data for the chamber conditions the displacement of the target could be determined for several gas densities. This data could serve to define a maximum amount of protective gas. Finally, the protective gas density must not result in a heat load higher than the target can survive.

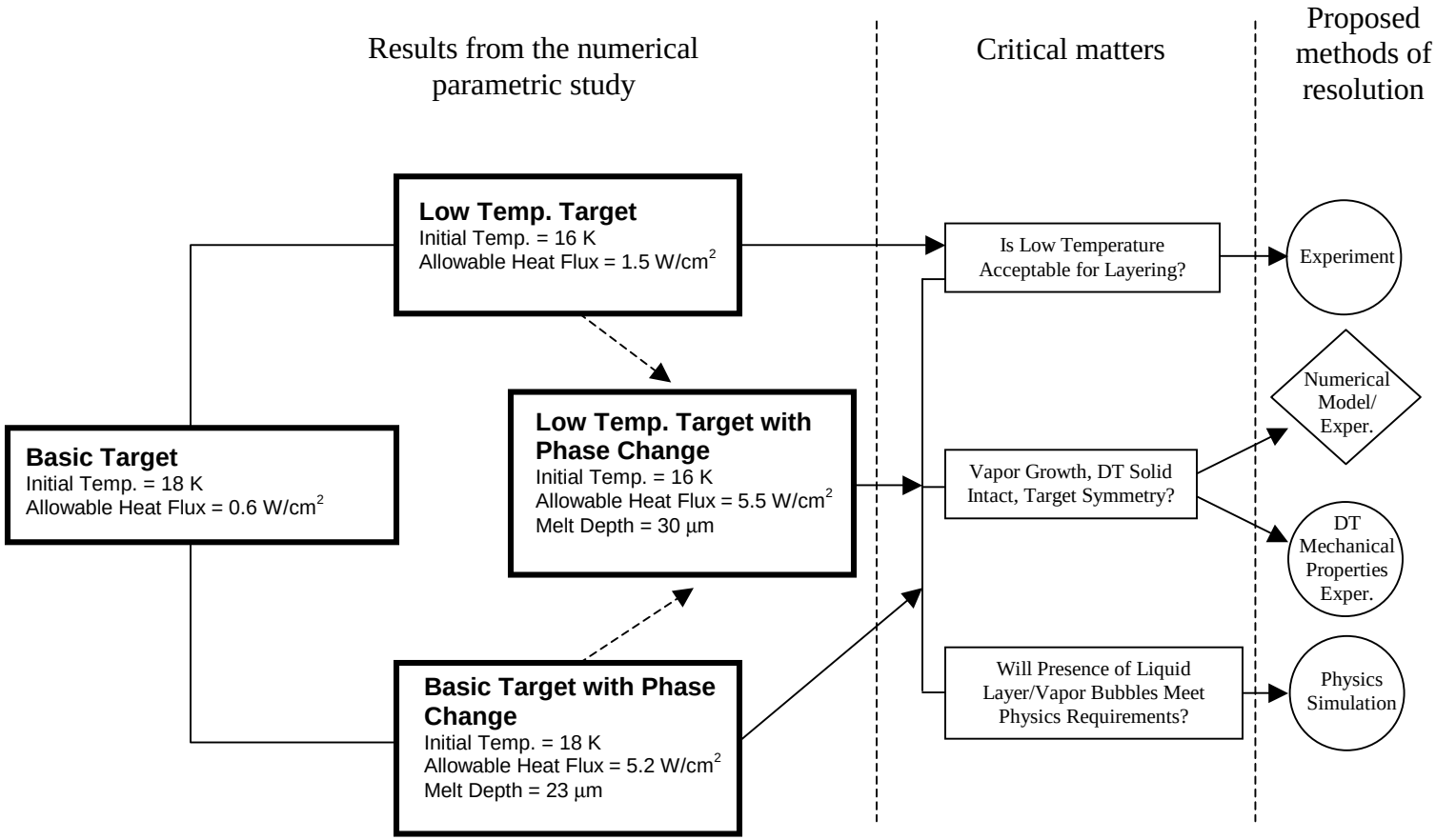

Figure M.1. The design plan for a basic target.

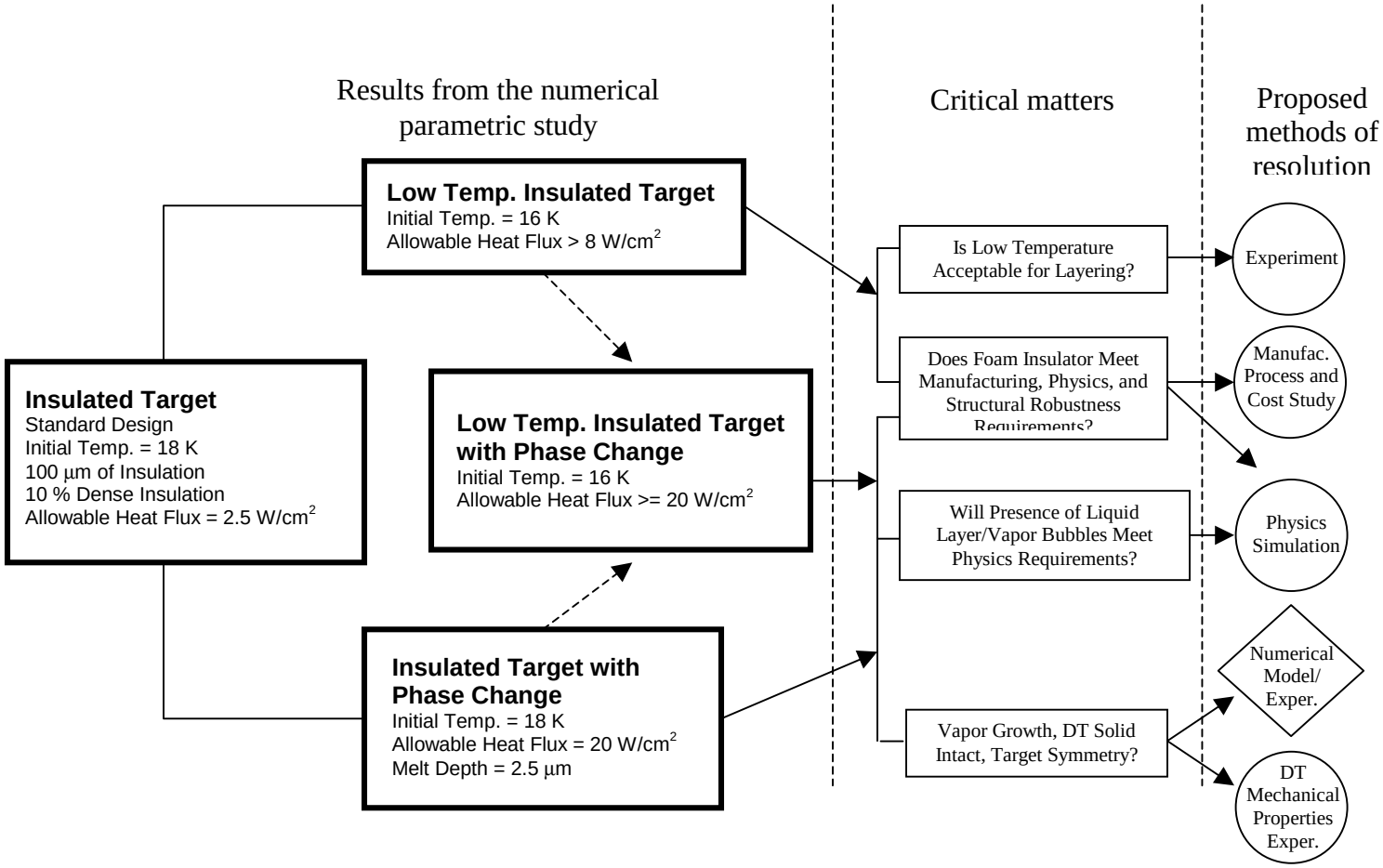

Figure M.2. The design plan for an insulated target.

Coupling of the Target Design and Chamber Design

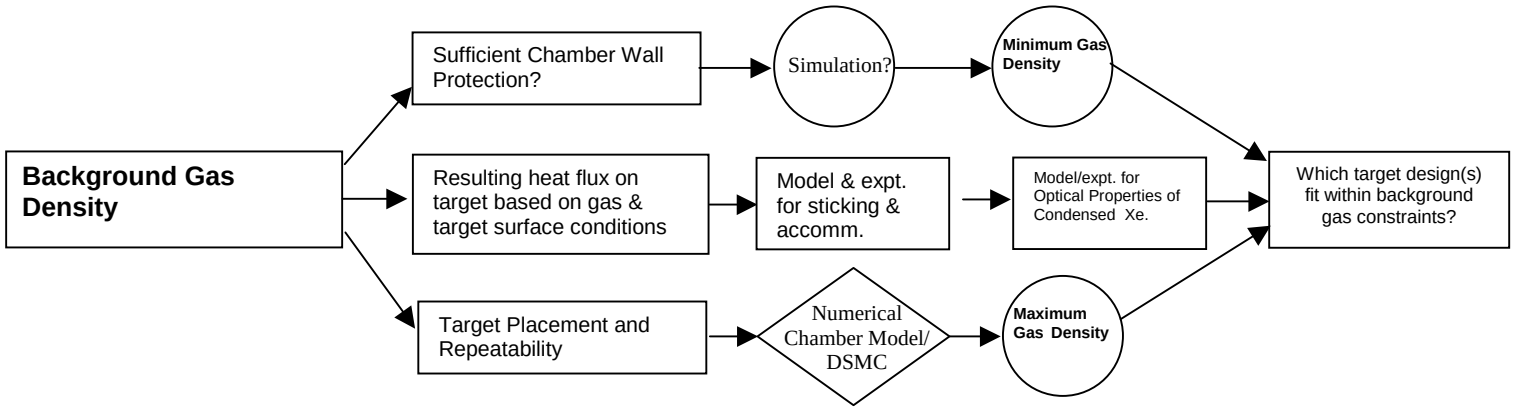

Figure M.3. A flow chart illustrating the coupling between target design and chamber design.

APPENDIX N

Integrated Thermomechanical Model Code

The following is a listing of the integrated thermomechanical model that was written in MATLAB. The subroutines that returned material properties are not show due to their simplicity and length. A hierarchy, which shows the order the subroutines are called, is shown first for reference.

Temp_adjust.m – The main program for the thermomechanical model.
- mass_plastic_foam.m – calculates the mass in the polymer shell and foam.
  - plasticdensity.m – returns the density of the polymer.
  - foamdensity.m – returns the density of the polymer foam at its fully dense value.
- mass_DT.m – calculates the mass in the DT/Foam.
  - dfdensity.m – returns the density of the DT/Foam.
- vapor_pressure.m – calculates the DT vapor pressure based on the temperature.
- condv.m – calculates the thermal conductivity of the DT vapor.
- Temp_total.m – sets up the linear system of equations representing the PDE.
  - nextdtdensity.m – extrapolates the DT density.
    - dtdensity.m – returns the DT density.
  - nextheatdt.m – extrapolates the DT specific heat.
    - heatdt.m – returns the DT specific heat.
  - nextconddt.m – extrapolates the DT thermal cond.
    - conddt.m – returns the DT thermal con.
  - nextdfdensity.m – extrapolates the DT/Foam den.
    - dfdensity.m – returns the DT/Foam den
  - nextheatdf.m
    - heatdf.m
  - nextconddf.m
    - conddf.m
  - nextplasticdensity.m
    - plasticdensity.m
  - nextheatp.m
    - heatp.m
  - nextcondp.m
    - condp.m
  - nextfoamdensity.m
    - foamdensity.m
  - nextheatf.m
    - heatf.m
  - nextcondf.m
    - condf.m
  - thomas.m – solves the set of linear equations using the Thomas algorithm.
- melt.m – calculates the thickness of the melt layer.
- brent_vol.m – minimization scheme that finds the deflection of the DT and polymer due to phase change by adjusting the pressure load until the deflection and the volume change due to phase change are consistent.
- find_triplet_vol.m – finds two pressures that are result in too much and too little deflection.
- Volume_change.m – calculates the available volume for phase change as a function of pressure.
- init_triplet_vol.m – returns three initial pressures based on an assumed pressure.
- vapor_flux.m – calculates the amount of DT vaporization.

Currently it is assumed that the DT and DT/Foam have identical properties, in this case these six subroutines could be reduced to three

If an insulator is used the difference in the foam and plastic properties are accounted for by multiplying factors. This means that these six subroutines could be reduced to three.

```
clear all;
% This is the main program for the 1-D integrated thermomechanical model
of
% an IFE direct drive target. This code can include a uniform vapor layer.

    %%%%% The incident heat flux (W/m^2) %%%%%%

    qin = -4.5e4;

    % Declare some global variables that will be need in subroutines
    global rp tp tv rv tdf tdt tdf_nm np dx dx2 dx3 nf ndf ndf_nm ndt n2
ndt
    global n4 dt qmain rdt R denc alphac alphap alphaf Tinf hb fd fcp fk
pd
    global pcp pk denc cpc kc nc ro_prev j_prev A therm_p Ep pois tp
v_layer
    global rinst revap Tp_ave tf rinst revap_inst n_change tdf_melt n1
ng_one
    global ng_two nlf npo dens_mult rb Edt pois_dt Vol_ch melt_cur n_He
kbolt
    global f_He Press_hel_guess Tplas_ave

    qmain = qin;

    %%%%%%%%%%% Define the time step (s) based on the heat flux
%%%%%%%%%%
    %% the time step should also be set according to the node spacing
%%%%%
    dt = 1e-5;%4e-4;%1e-5;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%% Define the geometry of the target %%%%%%%%%%%%%%%%
    rp = 2e-3; % Inner radius of the plastic shell (m)
    tp = 2e-6; % Thickness of the plastic shell (m)

    %%% If a foam insulator is used the subroutine Temp_total.m must be
    %%% changed to allow for the insulator
    tf = 0; % Thickness of the foam outer coat (m)
    %tg = 10e-6; % Thickness over which the foam outer coat changes
linearly
    % from fully dense to low density
    %tpo = 5e-6; % Thickness of outer plastic shell
    %tlf = tf - (2*tg) - tpo; % Thickness of low density foam
    %dens_mult = .10; % Density fraction of fully dense assigned to low
    %density foam

    tv = 1e-10; % Initial thickness of the vapor layer (m)
    tdf = 290e-6;%-tv; % Thickness of the DT/Foam region (m)
    tdt = 190e-6; % Thickness of the DT ice region (m)
    rdt = 1.52e-3; % Inner radius of the DT ice
    revap = rp; % Radius of the evaporation interface

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



        Ttot_p(i) = Tnp(i);
    end
    for i = n4+1:n4+n2
        Ttot(i) = To(i-n4);
        Ttot_p(i) = Top(i-n4);
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Create 1-D mesh
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    np = floor(tp/.1e-6)+1; % Number of nodes in the plastic shell
    dx = (tp/(np-1)); % Spacing between nodes in outer segment

    % The following are used if a foam insulator is present
    %ng_one = floor(tg/dx)+1; % number of nodes in first decreasing foam
density portion
    %nlf = floor(tlf/dx); % number of nodes in the low density foam
    %ng_two = floor(tg/dx); % The number of nodes in the increasing foam
density portion
    %npo = floor(tpo/dx); % The number of nodes in the outer plastic shell
    %nf = floor(tf/dx)+1; % Number of nodes in foam

    % The following are used for insulated and uninsulated targets
    dx2 = 0.5e-6; % Spacing between nodes in the DT
    ndt = floor(tdt/dx2); % Number in DT
    tdf_nm = 110e-6; % Thickness of the coarse mesh segment of the DT/Foam
    ndf_nm = floor(tdf_nm/dx2)+1; % Number in DT/Foam coarse mesh segment
    tdf_melt = tdf-tdf_nm; % Thickness of the fine mesh DT/Foam segment
    dx3 = .5e-6; % Spacing between nodes in fine mesh portion of the
DT/Foam
    ndf = floor(tdf_melt/dx3)+1; % Number in DT/Foam fine mesh segment
    %%% Note that it is best to use the same spacing in all of the DT
%%%%%
    n1 = ndt + ndf_nm; % Number of nodes in the coarse mesh segment of the
DT & DT/Foam
    n2 = np; % Number of nodes in the outer segment (plastic)
    %n2 = np + nf; %If an insulator is used
    n4 =ndf+ndf_nm+ndt; % Number of nodes in the inner segment (DT &
DT/Foam)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%% Initial Temperature Profiles %%%%%%%%%%%%%%%%%%%%%%
    % Initialize vectors
    To = zeros(n2,1);
    Top = zeros(n2,1);
    Tn = zeros(n4,1);
    Tnp = zeros(n4,1);
    Ttot = zeros((n4+n2),1);
    Ttot_p = zeros((n4+n2),1);
    % Outer segment
    for i=1:n2;
        To(i)=18;%18;%19.78999;
        Top(i)=18;%18;%19.78999; % Temperature at the previous time step
    end
    % Inner segment
    for i=1:n4
        Tn(i)=18;%18;%19.78999;
        Tnp(i)=18;%18;%19.78999; % Temperature at the previous time step
    end
    % The overall temperature profile
    for i = 1:n4
        Ttot(i) = Tn(i);

    %%%%%%%%%%% Define some constant properties
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    R = 1662.86; % Gas constant for DT (J/kg-K)
    kbolt = 1.3807e-23; % Boltzmann Constant (J/K)
    therm_p = .222e-4; % Thermal expansion coefficient for the plastic
shell
    Ep = 3.4e9; % Youngs modulus for the plastic shell (Pa)
```

```
    pois = 0.3; % Poison's ratio for the plastic shell
    Edt = 2.312e7; % Youngs modulus for the DT shell, adjusted according
to
    % the initial temperature (Pa)
    pois_dt = 0.3;% Poison's ratio for the DT shell

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%% If using data from file for initial temperature profile
%%%%%%%%%%%
%         fid = fopen('To.txt','r');
%         To=fscanf(fid,'%8g',inf);
%         fclose(fid);
%
%         fid = fopen('Top.txt','r');
%         Top=fscanf(fid,'%8g',inf);
%         fclose(fid);
%
%         fid = fopen('Tn.txt','r');
%         Tn=fscanf(fid,'%8g',inf);
%         fclose(fid);
%
%         fid = fopen('Tnp.txt','r');
%         Tnp=fscanf(fid,'%8g',inf);
%         fclose(fid);
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        Tp_ave = (Tn(1)+Tn(np))/2; % Average initial plastic temperature

        % Calculate the mass of each node in the plastic and foam section
        [mass_pf]=mass_plastic_foam(To,Top);
        [mass_inner] = mass_DT(Tn,Tnp); % Only the mass in the melting
section

        %%%%%%%% Calculate the inital volume on the fine mesh section
%%%%%%%
        rm = rp - tdf_melt; % inner radius of the melting section (m)
        Vm_o = 0; % Initialize the initial volume of the melting section
        Mm = 0; % Initialize the initial mass of the melting section
        for j=1:ndf-1
            dV= (4*pi/3)*((rm+(j+1)*dx3)^3-(rm+(j*dx3))^3);
            Vm_o = Vm_o + dV;
            dm(j) = dV*(dfdensity(Tn(n1+j))); % Mass of each node (Const.)
            Mm = Mm + dm(j);
        end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        Tplas(1) = To(1); % Initialize vector for storing the inner
polymer surface temperature
        Tmax_foam(1) = To(n2); % Initialize vector for storing maximum
outer temperature
        Vm(1) = Vm_o; % Volume of the melting section
        q_evaporation(1) =0;
        meltdepth(1) = 0;
        q_out_plot(1) = 0;
        q_in_plot(1) = 0;
        DeltaE_pf_tot(1) = 0;
        Total_energy_in(1) = 0;
        Energy_lv_total(1) = 0;
        DeltaE_DT_tot(1) = 0;
        Energy_error(1) = 0;
        Pressure_load(1) = Press_tot(1);
        Stress_DT (1) = 0;

        %%%%% Calculate the initial deflection of the plastic shell
%%%%%%%
        melt_cur = 0;
        Vol_ch = 0; % Change in Volume of DT due to solid-to-liquid p.c.
        Press_tot(1) = vapor_pressure(Tn(n4));
        % Calculate the deflection of the polymer shell based on total
pressure
        Def_poly(1) = Press_tot(1)*(rp^2)*(1-pois)/(2*Ep*tp);
        % Calculate the deflection of the DT shell based on total pressure
        Def_DT(1) = (Press_tot(1)*(rp-melt_cur)/Edt)*(((1-
pois_dt)*(rdt^3+2*rp^3)/(2*(rp^3-rdt^3)))-pois_dt);
        % Calculate volume of the vapor
        Vol_vap(1) = (4*pi/3)*((rp+Def_poly)^3-rp^3+(rp-melt_cur)^3-(rp-
melt_cur-Def_DT)^3);
        % Calculate the stress in the DT
        Stress_DT(1) = 3*Press_tot(1)*(rp-melt_cur)^3/(2*(rp-melt_cur)^3-
rdt^3);
        % Calculate the stress in the Polymer
        Stress_Poly(1) = Press_tot(1)*rp/(2*tp);
        rcalc(1) = rp+Def_poly(1); % Radius of the plastic shell under
load
        dr(1) = -Def_DT(1); % change in Outer radius of evaporation
interface due to expansion (melting/thermal expansion)
        r_poly_end = rp+Def_poly(1); % The radius of the plastic shell at
beginning of the ith time step
        r_DT_end = revap+dr(1);% The radius of the DT interface at the
beginning of the ith time step
        gap(1) = r_poly_end - r_DT_end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %%%%%%%%%%%%%%%%%%% Calculate the initial vapor mass in the gap
%%%%%%%%%%%%%%%%%%%%%%%%
        A = 4*pi*(revap^2);% Surface area of the vapor interface (assumed
constant troughout)
        Mo = vapor_pressure(Tn(n4))*Vol_vap(1)/(R*Tn(n4));%ro_prev*V; %
Initial mass of DT in the vapor layer (kg)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%

        %%%%%%%%%%%%%%%%%%%%%%%%%%%% Initialize some storage vectors
%%%%%%%%%%%%%%%%%%%%%%%%%%
        Time(1) = 0; % Initialize time vector
        Tint(1) = Tn(n4); % Initialize vector for storing the evaporation
surface temperature

        Stress_Poly(1) = 0;
        counter = 1;
        index=1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
        %%%%%%%Solve the heat conduction equation by marching forward in
time %%%%%%%%%%%%%%%%%%%
        for i=1:1000000
            rinst = rcalc(i); % The radius of the plastic shell at
beginning of the ith time step
```

```
            revap_inst = revap+dr(i); % The radius of the DT interface at
the beginning of the ith time step
            % Save some values for later use
            Tn_save = Tn;
            To_save = To;
            Ttot_save = Ttot;
            Tnp_save = Tnp;
            Mo_save = Mo;

            %%%%%%%%%%%% Calculate h based on data from the i-1 time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % Calculate the thermal conductivity of the vapor
            kvp = condv(To(1)); % Conductivity of vapor at polymer shell
temperature
            kvdf = condv(Tn(n4)); % Conductivity of vapor at DT outer
temperature
            kave = (kvp+kvdf)/2; % Average Conductivity based on temps
            delta_ave = rinst-revap_inst;

            if delta_ave <= 0 % no vapor
                h = 12e4;
                [Ttot_temp] = Temp_total(Ttot,Ttot_p,h);
                %%% Make Temp vectors for outer (To) and inner (Tn)
sections %%%%
                for v = 1:n4
                    Tn_temp(v) = Ttot_temp(v);
                end
                for v = 1:n2
                    To_temp(v) = Ttot_temp(n4+v);
                end

                %%%%%%%%%%%%%%%%%%% Calc. expansion of the DT based on
temporary prof. %%%%%%%%%%%%%%%%%%%%%%
                % Calculate the volume of the DT melting section
                Vm(i+1) = 0;
                for j = 1:ndf-1
                    dV = dm(j)/(dfdensity(Tn_temp(j+n1)));
                    Vm(i+1)= Vm(i+1) + dV;
                end
                Vol_ch = Vm(i+1)-Vm_o; % The change in volume of the DT
melting section from the original

                    Press_tot(i+1) = vapor_pressure(Tn(n4));
                    % Calculate the deflection of the polymer shell based
on total pressure
                    Def_poly(i+1) = Press_tot(i+1)*(rp^2)*(1-
pois)/(2*Ep*tp)+(rp*therm_p*(Tplas_ave-Tp_ave));
                    % Calculate the deflection of the DT shell based on
total pressure
                    Def_DT(i+1) = (Press_tot(i+1)*(rp-melt_cur)/Edt)*(((1-
pois_dt)*(rdt^3+2*rp^3)/(2*(rp^3-rdt^3)))-pois_dt);
                    % Calculate volume of the vapor
                    Vol_vap(i+1) = (4*pi/3)*((rp+Def_poly(i+1))^3-
rp^3+(rp-melt_cur)^3-(rp-melt_cur-Def_DT(i+1))^3);
                    % Volume available due to deflection of DT
                    Vol_def_DT = (4*pi/3)*((rp-melt_cur)^3-(rp-melt_cur-
Def_DT(i+1))^3);
                    Vol_outer = Vol_ch-Vol_def_DT; % Amount of volume
change at outer layer of DT
                    dr(i+1) = (((3*(Vol_outer)/(4*pi))+(revap^3))^(1/3))-
revap; % Change in radius of outer DT layer

                    r_evap_calc = ((rp+Def_poly(i+1))^3-
(3*Vol_vap(i+1)/(4*pi)))^(1/3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%

                % Test to see if a gap exists due to thermal expansion of
polymer shell
                % Calculate the average plastic temperature
                Tplas_ave = (To_temp(1)+To_temp(n2))/2;
                % Calculate the average change in temp
                Temp_change = Tplas_ave-Tp_ave;
                Thermalex = rp*therm_p*(Tplas_ave-Tp_ave);
                rout_dt = ((3*Vm(i+1)/(4*pi))+ (rp-tdf_melt)^3)^(1/3);
                rplas = rp+Thermalex;
                gap_test = rplas-rout_dt

                % Calc. melt layer based on temporary temp. prof.
                meltdepth(i+1) = melt(Tn);
                melt_cur = meltdepth(i+1);

                Press_tot(i+1) = brent_vol(Press_tot(i)); % Find pressure
if no vapor is present, using a minimization scheme

                % Calculate the deflection of the polymer shell based on
total pressure
                Def_poly(i+1) = Press_tot(i+1)*(rp^2)*(1-
pois)/(2*Ep*tp)+(rp*therm_p*(Tplas_ave-Tp_ave));
                % Calculate the deflection of the DT shell based on total
pressure
                Def_DT(i+1) = (Press_tot(i+1)*(rp-melt_cur)/Edt)*(((1-
pois_dt)*(rdt^3+2*rp^3)/(2*(rp^3-rdt^3)))-pois_dt);
                % Calculate volume of the vapor
                Vol_vap(i+1) = 0;

                dr(i+1) = Def_poly(i+1);
                if gap_test <= 0
                    gap(i+1) = 0;
                    M2 = 0;
                end
                if gap_test > 0;


                    r_poly_end = rp+Def_poly(i+1); % The radius of the
plastic shell at beginning of the ith time step
                    r_DT_end = revap+dr(i+1);% The radius of the DT
interface at the beginning of the ith time step
                    gap(i+1) = r_poly_end - r_DT_end;
                    delta_ave = (gap(i+1)+gap(i))/2;
                end
            end

            if delta_ave > 0;
                h = kave/delta_ave; % temporary h

                %%%%%%%%%%%%%%%%%%% Perform calculations based on
temporary h %%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%
                % Calculate a temporary temperature profile Ttot based on
assumed h from above
                [Ttot_temp] = Temp_total(Ttot,Ttot_p,h);
                %%% Make Temp vectors for outer (To) and inner (Tn)
sections %%%%
                for v = 1:n4
```

```
                    Tn_temp(v) = Ttot_temp(v);
                end
                for v = 1:n2
                    To_temp(v) = Ttot_temp(n4+v);
                end

                %%%%%%%%%%%%%%%%%%% Calc. expansion of the DT based on
temporary temp. prof. %%%%%%%%%%%%%%%%%%%%%%%
                % Calculate the volume of the DT melting section
                Vm(i+1) = 0;
                for j = 1:ndf-1
                    dV = dm(j)/(dfdensity(Tn_temp(j+n1)));
                    Vm(i+1)= Vm(i+1) + dV;
                end
                Vol_ch = Vm(i+1)-Vm_o; % The change in volume of the DT
melting section from the original

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%

                % Calc. melt layer based on temporary temp. prof.
                meltdepth(i+1) = melt(Tn);
                melt_cur = meltdepth(i+1);


                Tn_star = Ttot_temp(n4); % Estimated liq/vapor surface
temperature
                To_star = Ttot_temp(n4+1); % Estimated plastic shell inner
wall temp
                Tgas = (Tn_star+To_star)/2; % Estimated average gas temp
                % Calculate the average plastic temperature
                Tplas_ave = (To_temp(1)+To_temp(n2))/2;
                % Calculate the average change in temp
                Temp_change = Tplas_ave-Tp_ave;

                % Calculate the DT vapor pressure in the gap assuming it
is saturated at t = n+1
                Press_tot(i+1) =
(Tgas/Tn_star)^(1/2)*vapor_pressure(Tn_star);
                % Calculate the deflection of the polymer shell based on
total pressure
                Def_poly(i+1) = Press_tot(i+1)*(rp^2)*(1-
pois)/(2*Ep*tp)+rp*therm_p*(Tplas_ave-Tp_ave);
                % Calculate the deflection of the DT shell based on total
pressure
                Def_DT(i+1) = (Press_tot(i+1)*(rp-melt_cur)/Edt)*(((1-
pois_dt)*(rdt^3+2*rp^3)/(2*(rp^3-rdt^3)))-pois_dt);
                % Calculate volume of the vapor
                Vol_vap(i+1) = (4*pi/3)*((rp+Def_poly(i+1))^3-rp^3+(rp-
melt_cur)^3-(rp-melt_cur-Def_DT(i+1))^3);

                % Calculate the vapor mass, heat flux due to evaporation
based on
                % temporary temp. prof.
                [q_evap,M2] = vapor_flux(Ttot_temp,Ttot,Mo,Vol_vap(i+1));

                % Volume available due to deflection of DT
                Vol_def_DT = (4*pi/3)*((rp-melt_cur)^3-(rp-melt_cur-
Def_DT(i+1))^3);
                Vol_outer = Vol_ch-Vol_def_DT; % Amount of volume change
at outer layer of DT
                dr(i+1) = (((3*(Vol_outer)/(4*pi))+(revap^3))^(1/3))-
revap; % Change in radius of outer DT layer
```

```
                r_evap_calc = ((rp+Def_poly(i+1))^3-
(3*Vol_vap(i+1)/(4*pi)))^(1/3);
                r_poly_end = rp+Def_poly(i+1); % The radius of the plastic
shell at beginning of the ith time step
                r_DT_end = revap+dr(i+1);% The radius of the DT interface
at the beginning of the ith time step
                gap(i+1) = r_poly_end - r_DT_end;

                %%%%%% Calculate h based on average of i* (estimate) and
i-1 time %%%%%%%
                % Calculate the thermal conductivity of the vapor
                kvp = (condv(To(1))+condv(To_temp(1)))/2; % Conductivity
of vapor at polymer shell temperature
                kvdf = (condv(Tn(n4))+condv(Tn_temp(n4)))/2; %
Conductivity of vapor at DT outer temperature
                kave = (kvp+kvdf)/2; % Average Conductivity based on temps

                delta_ave = ((rinst-revap_inst)+(r_poly_end-
r_evap_calc))/2;

                if delta_ave <= 0
                    h = 12e4;
                    [Ttot_temp] = Temp_total(Ttot,Ttot_p,h);
                    %%% Make Temp vectors for outer (To) and inner (Tn)
sections %%%%
                    for v = 1:n4
                        Tn_temp(v) = Ttot_temp(v);
                    end
                    for v = 1:n2
                        To_temp(v) = Ttot_temp(n4+v);
                    end
                    % Calculate the average plastic temperature
                         Tplas_ave = (To_temp(1)+To_temp(n2))/2;
                         % Calculate the average change in temp
                         Temp_change = Tplas_ave-Tp_ave;
                    %%%%%%%%%%%%%%%%%%% Calc. expansion of the DT based on
temporary prof. %%%%%%%%%%%%%%%%%%%%%%%
                    % Calculate the volume of the DT melting section
                    Vm(i+1) = 0;
                    for j = 1:ndf-1
                        dV = dm(j)/(dfdensity(Tn_temp(j+n1)));
                        Vm(i+1)= Vm(i+1) + dV;
                    end
                    Vol_ch = Vm(i+1)-Vm_o; % The change in volume of the
DT melting section from the original

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%

                    % Calc. melt layer based on temporary temp. prof.
                    meltdepth(i+1) = melt(Tn);
                    melt_cur = meltdepth(i+1);

                    Press_tot(i+1) = brent_vol(Press_tot(i));
                    % Calculate the deflection of the polymer shell based
on total pressure
                    Def_poly(i+1) = Press_tot(i+1)*(rp^2)*(1-
pois)/(2*Ep*tp)+rp*therm_p*(Tplas_ave-Tp_ave);
                    % Calculate the deflection of the DT shell based on
total pressure
                    Def_DT(i+1) = (Press_tot(i+1)*(rp-melt_cur)/Edt)*(((1-
pois_dt)*(rdt^3+2*rp^3)/(2*(rp^3-rdt^3)))-pois_dt);
                    % Calculate volume of the vapor
                    Vol_vap(i+1) = 0;
```

```
                    dr(i+1) = Def_poly(i+1);
                    gap(i+1) = 0;
                end

                if delta_ave > 0;%1e-9;
                    h = kave/delta_ave;
                    % Calculate a temporary temperature profile Ttot based
on assumed h from above
                    [Ttot_temp2] = Temp_total(Ttot,Ttot_p,h);
                    diff = abs(Ttot_temp(n4+1)-Ttot_temp2(n4+1));

                    while diff > 1e-3
                        Ttot_temp = Ttot_temp2;
                        %%% Make Temp vectors for outer (To) and inner
(Tn) sections %%%%

                        for v = 1:n4
                            Tn_temp(v) = Ttot_temp(v);
                        end
                        for v = 1:n2
                            To_temp(v) = Ttot_temp(n4+v);
                        end

                        %%%%%%%%%%%%%%%%%%% Account for the expansion of
the DT %%%%%%%%%%%%%%%%%%%%%%%

                        % Calculate the average plastic temperature
                         Tplas_ave = (To_temp(1)+To_temp(n2))/2;
                        % Calculate the average change in temp
                        Temp_change = Tplas_ave-Tp_ave;

                        % Calculate the DT vapor pressure in the gap
assuming it is saturated at t = n+1
                        Press_tot(i+1) =
(Tgas/Tn_star)^(1/2)*vapor_pressure(Tn_star);

                        % Calculate the deflection of the polymer shell
based on total pressure
                        Def_poly(i+1) = Press_tot(i+1)*(rp^2)*(1-
pois)/(2*Ep*tp)+rp*therm_p*(Tplas_ave-Tp_ave);
                        % Calculate the deflection of the DT shell based
on total pressure
                        Def_DT(i+1) = (Press_tot(i+1)*(rp-
melt_cur)/Edt)*(((1-pois_dt)*(rdt^3+2*rp^3)/(2*(rp^3-rdt^3)))-pois_dt);
                        % Calculate volume of the vapor
                        Vol_vap(i+1) = (4*pi/3)*((rp+Def_poly(i+1))^3-
rp^3+(rp-melt_cur)^3-(rp-melt_cur-Def_DT(i+1))^3);

                        % Calculate the vapor mass, heat flux due to
evaporation based on
                        % temporary temp. prof.
                        [q_evap,M2] =
vapor_flux(Ttot_temp,Ttot,Mo,Vol_vap(i+1));

                        Vol_def_DT = (4*pi/3)*((rp-melt_cur)^3-(rp-
melt_cur-Def_DT(i+1))^3);
                        Vol_outer = Vol_ch-Vol_def_DT; % Amount of volume
change at outer layer of DT
                        dr(i+1) =
(((3*(Vol_outer)/(4*pi))+(revap^3))^(1/3))-revap;

                        r_evap_calc = ((rp+Def_poly(i+1))^3-
(3*Vol_vap(i+1)/(4*pi)))^(1/3);
```

```
                        %$$$$$$$ Calculate the volume of the DT melting
section  %%%%%%%%%%%%%%%%%%%%%
                        Vm(i+1) = 0;
                        for j = 1:ndf-1
                            dV = dm(j)/(dfdensity(Tn_temp(j+n1)));
                            Vm(i+1)= Vm(i+1) + dV;
                        end
                        Vol_ch = Vm(i+1)-Vm_o; % The change in volume of
the DT melting section from the original

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

                        % Determine how much of the DT has melted
                        meltdepth(i+1) = melt(Tn);
                        melt_cur = meltdepth(i+1);

                        % Define some average temeratures over the time
step
                        Tn_star = Ttot_temp(n4); % The liq/vapor surface
temperature
                        To_star = Ttot_temp(n4+1); % plastic shell inner
wall temp
                        Tgas = (Tn_star+To_star)/2; % Gas temp

                        r_poly_end = rp+Def_poly(i+1); % The radius of the
plastic shell at beginning of the ith time step
                        r_DT_end = revap+dr(i+1);% The radius of the DT
interface at the beginning of the ith time step
                        gap(i+1) = r_poly_end - r_DT_end;

                        %%%%%% Calculate h based on averages from the i
(estimate) and i-1 time %%%%%%%              % Calculate the thermal
conductivity of the vapor
                        kvp = (condv(To(1))+condv(To_temp(1)))/2; %
Conductivity of vapor at polymer shell temperature
                        kvdf = (condv(Tn(n4))+condv(Tn_temp(n4)))/2; %
Conductivity of vapor at DT outer temperature
                        kave = (kvp+kvdf)/2; % Average Conductivity based
on temps
                        delta_ave = ((rinst-revap_inst)+(r_poly_end-
r_evap_calc))/2;

                        if delta_ave <= 0
                            h = 12e4;
                            [Ttot_temp] = Temp_total(Ttot,Ttot_p,h);
                            %%% Make Temp vectors for outer (To) and inner
(Tn) sections %%%%
                            for v = 1:n4
                                Tn_temp(v) = Ttot_temp(v);
                            end
                            for v = 1:n2
                                To_temp(v) = Ttot_temp(n4+v);
                            end

                            %%%%%%%%%%%%%%%%%%% Calc. expansion of the DT
based on temporary prof. %%%%%%%%%%%%%%%%%%%%%
                            % Calculate the volume of the DT melting
section
                            Vm(i+1) = 0;
                            for j = 1:ndf-1
                                dV = dm(j)/(dfdensity(Tn_temp(j+n1)));
                                Vm(i+1)= Vm(i+1) + dV;
                            end
```

```
                            Vol_ch = Vm(i+1)-Vm_o; % The change in volume
of the DT melting section from the original

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%

                            % Calc. melt layer based on temporary temp.
prof.
                            meltdepth(i+1) = melt(Tn);
                            melt_cur = meltdepth(i+1);

                            % Calculate the average plastic temperature
                            Tplas_ave = (To_temp(1)+To_temp(n2))/2;
                            % Calculate the average change in temp
                            Temp_change = Tplas_ave-Tp_ave;

                            Press_tot(i+1) = brent_vol(Press_tot(i));

                            h = kave/delta_ave;
                            % Calculate a temporary temperature profile
Ttot based on assumed h from above
                            [Ttot_temp2] = Temp_total(Ttot,Ttot_p,h);

                            diff = abs(Ttot_temp(n4+1)-Ttot_temp2(n4+1));
                        end
                    end
                end
            end

        Ttot = Ttot_temp; % Update the temperature vector
        rcalc(i+1) = rp+Def_poly(i+1); % The new radius of the plastic
shell
        Stress_DT(i+1) = 3*Press_tot(i+1)*(rp-melt_cur)^3/(2*(rp-
melt_cur)^3-rdt^3);
        Stress_Poly(i+1) = Press_tot(i+1)*rp/(2*tp);
        qin = -h*(Ttot(n4)-Ttot(n4+1)); % Calculate the heat flux into the
inner portion
        q_net_in = qin ;%+ q_evap; % The net heat flux into the inner
portion
        Mo = M2; % Update the mass in the vapor gap
        q_in_plot(i+1)=q_net_in;

        %%% Update the vectors To and Tn %%%%
        for v = 1:n4
            Tn(v) = Ttot(v);
        end
        for v = 1:n2
            To(v) = Ttot(n4+v);
        end
        Tnp = Tn_save; % Update "previous time" vector
        Top = To_save; % Update
        Ttot_p = Ttot_save;
        Time(i+1) = Time(i) + dt; % Time vector
        Tint(i+1) = Tn(n4); % Interface Temperature
        Tplas(i+1) = To(1); % Plastic/Vapor Interface Temperature

        % Calculate the change in energy of the plastic/foam since the
last time step
        [DeltaE_pf_dt]=Energy_change_pf(To,Top,mass_pf);
        DeltaE_pf_tot(i+1) = DeltaE_pf_tot(i) + DeltaE_pf_dt; % The total
change in energy of the p/f section since t=0

        % Calculate the change in energy of the DT/DT-Foam since the last
time step
```

```
                            % Calculate the deflection of the polymer
shell based on total pressure
                            Def_poly(i+1) = Press_tot(i+1)*(rp^2)*(1-
pois)/(2*Ep*tp)+rp*therm_p*(Tplas_ave-Tp_ave);
                            % Calculate the deflection of the DT shell
based on total pressure
                            Def_DT(i+1) = (Press_tot(i+1)*(rp-
melt_cur)/Edt)*(((1-pois_dt)*(rdt^3+2*rp^3)/(2*(rp^3-rdt^3)))-pois_dt);
                            % Calculate volume of the vapor
                            Vol_vap(i+1) = 0;

                            dr(i+1) = Def_poly(i+1);
                            gap(i+1) = 0;
                            break
                        end

                        if delta_ave > 0;%1e-9;
        [DeltaE_DT_dt]=Energy_change_DT(Tn,Tnp,mass_inner);
        DeltaE_DT_tot(i+1) = DeltaE_DT_tot(i) + DeltaE_DT_dt; % The total
change in energy of the p/f section since t=0

%             % Calculate the energy used for the liquid-vapor phase
change
%             Tn_ave = (Tn(n4)+Tnp(n4))/2;
%             % Determine the latent heat of vaporization
%             % % Determine the avereage latent heat of fusion (J/kg)
%             if Tn_ave < 19.78
%                 hlv = -
5.449134*(Tn_ave^3)+79.50222*(Tn_ave^2)+3764.932*(Tn_ave)+250197.3;
%             else
%                 if (Tn_ave >= 19.78 & Tn_ave < 40.01)
%                 hlv = -20.04217*(Tn_ave^3)+1150.727*(Tn_ave^2)-
23450.92*(Tn_ave)+441241.6;
%                 else
%                     if Tn_ave  > 40.01
%                     hlv = 0;
%                     end
%                 end
%
%             end
%             Energy_lv = 0;%hlv*(M2-Mo_save);
%             Energy_lv_total(i+1) = Energy_lv_total(i)+Energy_lv;

        % Calculate the energy input into the target over this time step
        Af = 4*pi*((rp+tp+tf)^2); % Surface area of foam shell
        Energy_in_dt = Af*qmain*dt;
        Total_energy_in(i+1) = Energy_in_dt+Total_energy_in(i);

        %Energy_error(i+1) = abs(abs(Total_energy_in(i+1))-
(DeltaE_pf_tot(i+1)+DeltaE_DT_tot(i+1)+Energy_lv_total(i+1)))/abs(Total_en
ergy_in(i+1));
        %Energy_error(i+1) = abs(abs(Total_energy_in(i+1))-
(DeltaE_pf_tot(i+1)+DeltaE_DT_tot(i+1))))/abs(Total_energy_in(i+1));

        %%%%%%%%%%%%%%%%%%%%%% Create plotting vectors
%%%%%%%%%%%%%%%%%%%%%%%%
%             if counter ==10; % Save data for plotting when true
%                 for j=1:n2
%                     y = n2+1;
%                     Tt(j,index)=To(y-j);
%                 end
%                 v = n4+(n2+1);
%                 for j=n2+1:n4+n2
%                     Tt(j,index)=Tn(v-j);
```

```
%                 end
%                 counter = 0;
%                 index = index+1;
%             end
        counter = counter+1

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        Tmax_foam(i+1) = Ttot(n4+n2);
        melt_test = Ttot(n4);

                Tt(j)=Tn(v-j);
            end
            counter = 0;
            index = index+1;
            % End loop DT Ultimate stress has been reached
            break;
        end
        if Stress_Poly(i+1) >= 3e7;%19.79
            % Save data for plotting
            for j=1:n2
                y = n2+1;
                Tt(j)=To(y-j);
            end
            v = n4+(n2+1);
            for j=n2+1:n4+n2
                Tt(j)=Tn(v-j);
            end
            counter = 0;
            index = index+1;
            % End loop DT Ultimate stress has been reached
            break;
        end
        if melt_test >= 39.4;%
            % Save data for plotting
            for j=1:n2
                y = n2+1;
                Tt(j)=To(y-j);
            end
            v = n4+(n2+1);
            for j=n2+1:n4+n2
                Tt(j)=Tn(v-j);
            end
            counter = 0;
            index = index+1;
            % End loop Triple point has been reached
            break;
        end
        if plas_test >= 370
                % Save data for plotting
            for j=1:n2
                y = n2+1;
                Tt(j)=To(y-j);
            end
            v = n4+(n2+1);
            for j=n2+1:n4+n2
                Tt(j)=Tn(v-j);
            end
             counter = 0;
             index = index+1;
             % End loop plastic melting temp reached
            break;
        end
        %output_count = output_count+1;
    end % End time time step
```

```
        plas_test = Ttot(n4+n2);
        if Stress_DT(i+1) >= 3.52e5;%2.96e5;%2.35e5;%2.96e5;%19.79
            % Save data for plotting
            for j=1:n2
                y = n2+1;
                Tt(j)=To(y-j);
            end
            v = n4+(n2+1);
            for j=n2+1:n4+n2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%


    % Data %%%%%%%-----------------------------------------------


     Out =
[Time;Tint;meltdepth;Tmax_foam;Press_tot;Tplas;Vol_vap;gap;Stress_DT;Stres
s_Poly];
     fid = fopen('Output_q4p5.txt','wt');
     fprintf(fid, '%8.4g\t %8.4g\t %8.4g\t %8.4g\t %8.4g\t %8.4g\t %8.4g\t
%8.4g\t %8.4g\t %8.4g\n', Out);
     fclose(fid);

     fid = fopen('Tprof_q4p5.txt','wt');
     fprintf(fid,'%8.4g\n',Tt);
     fclose(fid);

% end Temp_adjust.m
```

```
% Subroutine used to calculate the mass of each node in the
% plastic and foam region
function[mass_pf]=mass_plastic_foam(To,Top);

global rp np nf n2 dx ng_one ng_two nlf dens_mult npo

Volume_pf_total=0;
Mass_pf_total = 0;
Vpf = zeros(n2,1);
mass_pf = zeros(n2,1);
% Volume of each node in the plastic
for i = 1:1
    Vpf(i) = (4*pi/3)*((rp+(dx/2))^3-(rp)^3);
end
for i = 2:np-1
    Vpf(i)=(4*pi/3)*((rp+((i-1)*dx)+(dx/2))^3-(rp+((i-1)*dx)-(dx/2))^3);
end
for i = np:np
    Vpf(i)=(4*pi/3)*((rp+((i-1)*dx))^3-(rp+((i-1)*dx)-(dx/2))^3);
end

% Total Volume of Plastic and foam
for i = 1:np
Volume_pf_total = Volume_pf_total + Vpf(i);
end

% Mass of each plastic node
for j = 1:np
    mass_pf(j) = Vpf(j)*plasticdensity(To(j));
end

%%% Use if a foam insulator is included %%%%%%%%%%%%%%%%%%%%%%
% % Mass of each foam node in decreasing density portion
% index = 0;
% dens_step = (1-dens_mult)/(ng_one-1); % The amount that the density
decreases at each node
% for j = np+1:np + ng_one
%     mass_pf(j) = Vpf(j)*(1-(index*dens_step))*plasticdensity(To(j));
%     %dens_plot(j) = (1-(index*dens_step))*plasticdensity(To(j));
%     index = index+1;
% end
% % Mass of each node in the constant low density portion
% for j = np+ng_one+1:np+ng_one+nlf
%     mass_pf(j) = Vpf(j)*dens_mult*plasticdensity(To(j));
%     %dens_plot(j) =dens_mult*plasticdensity(To(j));
% end
% % Mass of each node in the increasing foam density portion
% dens_step = (1-dens_mult)/(ng_two);
% index = 1;
% for j = np+ng_one+nlf+1:np+ng_one+nlf+ng_two
%     mass_pf(j) =
Vpf(j)*(dens_mult+(index*dens_step))*plasticdensity(To(j));
%     %dens_plot(j) =(dens_mult+(index*dens_step))*plasticdensity(To(j));
%     index = index +1;
% end
% % Mass of each node in the outer plastic shell
% for j = np+ng_one+nlf+ng_two+1: np+ng_one+nlf+ng_two+npo
%     mass_pf(j) = Vpf(j)*plasticdensity(To(j));
%     %dens_plot(j) =plasticdensity(To(j));
% end

% end mass_plastic_foam.m
```

```
% Subroutine used to set-up linear set of eq's
function[Ttot]=Temp_total(Ttot,Ttot_p,h);

% Declare some global variables
global tdf tdt tdf_nm dx2 dx3 ndf ndf_nm ndt ndt n4 dt qmain rdt
```

```
% Subroutine used to calculate the mass of each node in the DT
%%%% Only the mass in the melting section %%%%%
function[mass_inner]=mass_DT(Tn,Tnp);

global rdt ndt n1 dx2 np nf n2 dx tdf_nm dx3 ndt ndf_nm n4 tdt

Volume_dt_total = 0;
Mass_dt_total = 0;

%%% Only fine mesh section %%%%
Vdt = zeros((n4-n1),1);
mass_inner = zeros((n4-n1),1);
rchange = rdt+tdf_nm+tdt;
index =1;
for w = 1:1
    Vdt(w)=(4*pi/3)*((rchange+(index*dx3)-(dx3/2))^3-(rchange+(index-
1)*dx3)^3);
    index = index+1;
end
index = 2;
for w = 2:n4-1-n1
    Vdt(w)=(4*pi/3)*((rchange+(index*dx3)-(dx3/2))^3-((rchange+((index-
1)*dx3)-(dx3/2))^3));
    index = index+1;
end
for w = n4-n1:n4-n1
    Vdt(w)=(4*pi/3)*((rchange+((index-1)*dx3))^3-(rchange+(index-1)*dx3-
(dx3/2))^3);
end
for z = 1:n4-n1
    Volume_dt_total = Volume_dt_total + Vdt(z);
end

% Mass and of each node
for j = 1:n4-n1
    mass_inner(j) = Vdt(j)*dfdensity(Tn(j));
end

for v = 1:n4-n1
    Mass_dt_total = Mass_dt_total + mass_inner(v);
end
% end mass_DT.m
```

```
global R denc alphac denc cpc kc nc
global rp np dx n2 dt qmain Tinf hb
global fd fcp fk pd pcp pk alphap alphaf n_change
global rp np nf n2 dx ng_one ng_two nlf dens_mult npo

hcont = 1e5; % Contact resistance coefficient for foam plastic boundary
hdf = 1e12; % coefficient used for coarse/fine boundary heat flux
conservation
ntot = n2+n4;
```

```
% Initialize e,f,g vectors
e = zeros(ntot,1);
f = zeros(ntot,1);
g = zeros(ntot,1);

% For the boundary between the DT vapor core and the DT (ndt) [First node]
dtd = nextdtdensity(Ttot(1),Ttot_p(1));
dtcp = nextheatdt(Ttot(1),Ttot_p(1));
dtk = nextconddt(Ttot(1),Ttot_p(1));
H = (2*dtk/(rdt/dx2))+((nextconddt(Ttot(2),Ttot_p(2))-dtk)/2);
G = 2*dt*dtk/(dtd*dtcp*(dx2^2));
e(1) = 0;
f(1)=1+(G);
g(1) = -(G);

% Fill the portion for the DT Ice
dt_by_dx2 = rdt/dx2;
dx2_sq = dx2^2;
for i=2:ndt
    dtd = nextdtdensity(Ttot(i),Ttot_p(i));
    dtcp = nextheatdt(Ttot(i),Ttot_p(i));
    dtk = nextconddt(Ttot(i),Ttot_p(i));
    H = (2*dtk/((dt_by_dx2)+(i-1)))+((nextconddt(Ttot(i+1),Ttot_p(i+1))-
nextconddt(Ttot(i-1),Ttot_p(i-1)))/2);
    G = dt/(dtd*dtcp*(dx2_sq));
    e(i) = (G)*((H/2)-dtk);
    f(i)=1+(2*G*dtk);
    g(i) = -(G)*((H/2)+dtk);
end

% For the boundary between the DT and the DT/Foam (ndt) the density and
the
% specific heat will be an average value, and the thermal conductivity
will change
% with position
n3 = ndt+1;
avedensndt =
(nextdtdensity(Ttot(n3),Ttot_p(n3))+nextdtdensity(Ttot(n3),Ttot_p(n3)))/2;

    R = dt/(dfd*dfcp*(dx2_sq));
    e(i) = -2*dfk*R;
    f(i) = 1+(2*R*dfk)+(R*G*2*dx2*hdf/dfk);
    g(i) = -R*G*2*dx2*hdf/dfk;
end

% Fill the first node in the fine mesh section
ns = ndt+ndf_nm+1;
nf = ndt+ndf_nm+1 ;
rchange = tdt+tdf_nm;
rch_by_dx3 = (rdt+rchange)/dx3;
dx3_sq = dx3^2;
for i = ns:nf;%i=ndt+ndf_nm+1:ndt+ndf_nm+1
    dfd = nextdfdensity(Ttot(i),Ttot_p(i));
    dfcp = nextheatdf(Ttot(i),Ttot_p(i));
    dfk = nextconddf(Ttot(i),Ttot_p(i));
    H = ((nextconddf(Ttot(i+1),Ttot_p(i+1))-
dfk)/4)+dfk*((1/((rch_by_dx3)+i-2))-1);
    G = ((nextconddf(Ttot(i+1),Ttot_p(i+1))-
dfk)/4)+dfk*((1/((rch_by_dx3)+i-2))+1);
    R = dt/(dfd*dfcp*(dx3_sq));
    e(i) = R*H*2*dx3*hdf/dfk;
    f(i) = 1+(2*R*dfk)-(R*H*2*dx3*hdf/dfk);
    g(i) = -2*dfk*R;

aveheatndt =
(nextheatdf(Ttot(n3),Ttot_p(n3))+nextheatdt(Ttot(n3),Ttot_p(n3)))/2;
dfk = nextconddf(Ttot(n3),Ttot_p(n3));
dtk = nextconddt(Ttot(n3),Ttot_p(n3));
aveconddt = (dtk+dfk)/2;
H = (2*dfk/((dt_by_dx2)+(n3-1)))+((nextconddf(Ttot(n3+1),Ttot_p(n3+1))-
nextconddt(Ttot(n3-1),Ttot_p(n3-1)))/2);
G = dt/(avedensndt*aveheatndt*(dx2_sq));
e(n3) = (G)*((H/2)-dtk);
f(n3)=1+(2*G*dtk);
g(n3) = -(G)*((H/2)+dtk);

% Fill the portion for the DT/foam up to the node before the fine mesh
ns = ndt + 2;
nf = ndt + ndf_nm-1;
for i = ns:nf;%i=ndt+2:ndt+ndf_nm-1
    dfd = nextdfdensity(Ttot(i),Ttot_p(i));
    dfcp = nextheatdf(Ttot(i),Ttot_p(i));
    dfk = nextconddf(Ttot(i),Ttot_p(i));
    H = (2*dfk/((dt_by_dx2)+(i-1)))+((nextconddf(Ttot(i+1),Ttot_p(i+1))-
nextconddf(Ttot(i-1),Ttot_p(i-1)))/2);
    G = dt/(dfd*dfcp*(dx2_sq));
    e(i) = (G)*((H/2)-dfk);
    f(i)= 1+(2*G*dfk);
    g(i) = -(G)*((H/2)+dfk);
end

% Fill the last node in the coarse mesh
ns = ndt+ndf_nm;
nf = ndt+ndf_nm;
for i= ns:nf;
    dfd = nextdfdensity(Ttot(i),Ttot_p(i));
    dfcp = nextheatdf(Ttot(i),Ttot_p(i));
    dfk = nextconddf(Ttot(i),Ttot_p(i));
    H = ((dfk-nextconddf(Ttot(i-1),Ttot_p(i-
1)))/4)+dfk*((1/((dt_by_dx2)+i-1))-1);
    G = ((dfk-nextconddf(Ttot(i-1),Ttot_p(i-
1)))/4)+dfk*((1/((dt_by_dx2)+i-1))+1);


end

% Fill the portion for the DT/foam in the fine mesh section
ns = ndt+ndf_nm+2;
nf = n4-1;
for i = ns:nf;%i=ndt+ndf_nm+2:n4-1
    rchange = tdt+tdf_nm;
    dfd = nextdfdensity(Ttot(i),Ttot_p(i));
    dfcp = nextheatdf(Ttot(i),Ttot_p(i));
    dfk = nextconddf(Ttot(i),Ttot_p(i));
    H = (2*dfk/((rch_by_dx3)+(i-1)))+((nextconddf(Ttot(i+1),Ttot_p(i+1))-
nextconddf(Ttot(i-1),Ttot_p(i-1)))/2);
    G = dt/(dfd*dfcp*(dx3_sq));
    e(i) = (G)*((H/2)-dfk);
    f(i)= 1+(2*G*dfk);
    g(i) = -(G)*((H/2)+dfk);
end

% Account for the DT/Foam outer boundary node
dfd = nextdfdensity(Ttot(n4),Ttot_p(n4));
dfcp = nextheatdf(Ttot(n4),Ttot_p(n4));
dfk = nextconddf(Ttot(n4),Ttot_p(n4));
H = ((dfk-nextconddf(Ttot(n4-1),Ttot_p(n4-
1)))/4)+dfk*((1/((rch_by_dx3)+n4-1))-1);
```

```
G = ((dfk-nextconddf(Ttot(n4-1),Ttot_p(n4-
1)))/4)+dfk*((1/((rch_by_dx3)+n4-1))+1);
R = dt/(dfd*dfcp*(dx3_sq));
e(n4)= -2*dfk*R;
f(n4)= 1+(2*R*dfk)+(R*G*2*dx3*h/dfk);
g(n4)= -R*G*2*dx3*h/dfk;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% The plastic shell and foam
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rp_by_dx = rp/dx;
dx_sq = dx^2;
% first node
for i=n4+1
    pd = nextplasticdensity(Ttot(i),Ttot_p(i));
    pcp = nextheatp(Ttot(i),Ttot_p(i));
    pk = nextcondp(Ttot(i),Ttot_p(i));
    H = ((nextcondp(Ttot(i+1),Ttot_p(i+1))-
nextcondp(Ttot(i),Ttot_p(i)))/4)+pk*((1/((rp_by_dx)+i-1))-1);
    G = ((nextcondp(Ttot(i+1),Ttot_p(i+1))-
nextcondp(Ttot(i),Ttot_p(i)))/4)+pk*((1/((rp_by_dx)+i-1))+1);
    R = dt/(pd*pcp*(dx_sq));
    e(i) = R*H*2*dx*h/pk;
    f(i)= 1+(2*R*pk)-(R*H*2*dx*h/pk);
    g(i) = -2*pk*R;
end

% Fill the portion for the plastic shell
for i=n4+2:n4+np-1
    pd = nextplasticdensity(Ttot(i),Ttot_p(i));
    pcp = nextheatp(Ttot(i),Ttot_p(i));
    pk = nextcondp(Ttot(i),Ttot_p(i));

%     fcp = nextheatf(Ttot(i),Ttot_p(i));
%     fk = nextcondf(Ttot(i),Ttot_p(i));
%     H = ((nextcondf(Ttot(i+1),Ttot_p(i+1))-fk)/4)+fk*((1/((rp_by_dx)+i-
2))-1);
%     G = ((nextcondf(Ttot(i+1),Ttot_p(i+1))-fk)/4)+fk*((1/((rp_by_dx)+i-
2))+1);
%     R = dt/(fd*fcp*(dx_sq));
%     e(i) = R*H*2*dx*hcont/fk;
%     f(i) = 1+(2*R*fk)-(R*H*2*dx*hcont/fk);
%     g(i) = -2*fk*R;
% end
%
% % Fill the decreasing density portion of the foam up to the last node in
% % the decreasing portion
% index = 1;
% dens_step = (1-dens_mult)/(ng_one-1); % The amount that the density
decreases at each node
% for i = n4+np+2:n4+np+ng_one-1
%     fd = (1-(index*dens_step))*nextfoamdensity(Ttot(i),Ttot_p(i));
%     fcp = nextheatf(Ttot(i),Ttot_p(i));
%     fk = (1-(index*dens_step))*nextcondf(Ttot(i),Ttot_p(i));
%     H = (((1-((index+1)*dens_step))*nextcondf(Ttot(i+1),Ttot_p(i+1))-(1-
((index-1)*dens_step))*nextcondf(Ttot(i-1),Ttot_p(i-
1)))/4)+fk*((1/((rp_by_dx)+i-2))-1);
%     G = (((1-((index+1)*dens_step))*nextcondf(Ttot(i+1),Ttot_p(i+1))-(1-
((index-1)*dens_step))*nextcondf(Ttot(i-1),Ttot_p(i-
1)))/4)+fk*((1/((rp_by_dx)+i-2))+1);
%     R = dt/(fd*fcp*(dx_sq));
%     e(i) = R*H;
%     f(i)=1+(2*R*fk);
%     g(i) = -R*G;
%     index = index +1;
    H = ((nextcondp(Ttot(i+1),Ttot_p(i+1))-nextcondp(Ttot(i-1),Ttot_p(i-
1)))/4)+pk*((1/((rp_by_dx)+i-1))-1);
    G = ((nextcondp(Ttot(i+1),Ttot_p(i+1))-nextcondp(Ttot(i-1),Ttot_p(i-
1)))/4)+pk*((1/((rp_by_dx)+i-1))+1);
    R = dt/(pd*pcp*(dx_sq));
    e(i) = R*H;
    f(i)=1+(2*R*pk);
    g(i) = -R*G;
end

% The boundary node for the plastic shell if it is the heat input node
for i = n4+np:n4+np
    % Get the properties at the current node
    pd = nextplasticdensity(Ttot(i),Ttot_p(i));
    pcp = nextheatp(Ttot(i),Ttot_p(i));
    pk = nextcondp(Ttot(i),Ttot_p(i));
    H = (2*pk/((rp_by_dx)+(i-1)))+((pk-nextcondp(Ttot(i-1),Ttot_p(i-
1)))/2);
    G = 2*dt*pk/((dx_sq)*pd*pcp);
    e(i) = -G;
    f(i) = 1+G;
    g(i) = 0;
    Ttot(i)=Ttot(i)-((dt*2*qmain/(pcp*pd*pk*dx))*((H/2)+pk));
end

% For use when a foam insulator is applied, note that the equations for
% last plastic shell node would have to be modified if a insulator were
used.
% % Fill the first node in the foam
% for i = n4+np+1:n4+np+1
%     fd = nextfoamdensity(Ttot(i),Ttot_p(i));
% end
%
% for i = n4+np+ng_one:n4+np+ng_one
%     fd = dens_mult*nextfoamdensity(Ttot(i),Ttot_p(i));
%     fcp = nextheatf(Ttot(i),Ttot_p(i));
%     fk = dens_mult*nextcondf(Ttot(i),Ttot_p(i));
%     H = ((dens_mult*nextcondf(Ttot(i+1),Ttot_p(i+1))-
(dens_mult+dens_step)*nextcondf(Ttot(i-1),Ttot_p(i-
1)))/4)+fk*((1/((rp_by_dx)+i-2))-1);
%     G = ((dens_mult*nextcondf(Ttot(i+1),Ttot_p(i+1))-
(dens_mult+dens_step)*nextcondf(Ttot(i-1),Ttot_p(i-
1)))/4)+fk*((1/((rp_by_dx)+i-2))+1);
%     R = dt/(fd*fcp*(dx_sq));
%     e(i) = R*H;
%     f(i)=1+(2*R*fk);
%     g(i) = -R*G;
% end
%
% % Fill the constant low density portion of the foam up to the last
constant
% % node
% for i = n4+np+ng_one+1:n4+np+ng_one+nlf-1
%     fd = dens_mult*nextfoamdensity(Ttot(i),Ttot_p(i));
%     fcp = nextheatf(Ttot(i),Ttot_p(i));
%     fk = dens_mult*nextcondf(Ttot(i),Ttot_p(i));
%     H = ((dens_mult*nextcondf(Ttot(i+1),Ttot_p(i+1))-
dens_mult*nextcondf(Ttot(i-1),Ttot_p(i-1)))/4)+fk*((1/((rp_by_dx)+i-2))-
1);
%     G = ((dens_mult*nextcondf(Ttot(i+1),Ttot_p(i+1))-
dens_mult*nextcondf(Ttot(i-1),Ttot_p(i-1)))/4)+fk*((1/((rp_by_dx)+i-
2))+1);
%     R = dt/(fd*fcp*(dx_sq));
%     e(i) = R*H;
```

```
%     f(i)=1+(2*R*fk);
%     g(i) = -R*G;
%
% end
%
% % Fill the last constant low density node
% dens_step = (1-dens_mult)/(ng_two);
% for i = n4+np+ng_one+nlf:n4+np+ng_one+nlf
%     fd = dens_mult*nextfoamdensity(Ttot(i),Ttot_p(i));
%     fcp = nextheatf(Ttot(i),Ttot_p(i));
%     fk = dens_mult*nextcondf(Ttot(i),Ttot_p(i));
%     H = (((dens_mult+dens_step)*nextcondf(Ttot(i+1),Ttot_p(i+1))-
dens_mult*nextcondf(Ttot(i-1),Ttot_p(i-1)))/4)+fk*((1/((rp_by_dx)+i-2))-
1);
%     G = (((dens_mult+dens_step)*nextcondf(Ttot(i+1),Ttot_p(i+1))-
dens_mult*nextcondf(Ttot(i-1),Ttot_p(i-1)))/4)+fk*((1/((rp_by_dx)+i-
2))+1);
%     R = dt/(fd*fcp*(dx_sq));
%     e(i) = R*H;
%     f(i)=1+(2*R*fk);
%     g(i) = -R*G;
%
% end

%
% % Fill the outer plastic shell up to the boundary node
% for i = n4+np+ng_one+nlf+ng_two+1:n4+np+ng_one+nlf+ng_two+npo-1
%     fd = nextfoamdensity(Ttot(i),Ttot_p(i));
%     fcp = nextheatf(Ttot(i),Ttot_p(i));
%     fk = nextcondf(Ttot(i),Ttot_p(i));
%     H = ((nextcondf(Ttot(i+1),Ttot_p(i+1))-nextcondf(Ttot(i-1),Ttot_p(i-
1)))/4)+fk*((1/((rp_by_dx)+i-2))-1);
%     G = ((nextcondf(Ttot(i+1),Ttot_p(i+1))-nextcondf(Ttot(i-1),Ttot_p(i-
1)))/4)+fk*((1/((rp_by_dx)+i-2))+1);
%     R = dt/(fd*fcp*(dx_sq));
%     e(i) = R*H;
%     f(i)=1+(2*R*fk);
%     g(i) = -R*G;
% end
%
 % Account for the Foam outer boundary node
% fd = nextfoamdensity(Ttot(n2+n4),Ttot_p(n2+n4));
% fcp = nextheatf(Ttot(n2+n4),Ttot_p(n2+n4));
% fk = nextcondf(Ttot(n2+n4),Ttot_p(n2+n4));
% H = (2*fk/((rp_by_dx)+(n2-2+n4)))+((fk-nextcondf(Ttot(n2-
1+n4),Ttot_p(n2-1+n4)))/2);
% G = 2*dt*fk/((dx_sq)*fd*fcp);
% e(n2+n4)= -G;
% f(n2+n4)= 1+(G);
% g(n2+n4)=0;
% Ttot(n2+n4)=Ttot(n2+n4)-((dt*2*qmain/(fcp*fd*fk*dx))*((H/2)+fk));

ntot = n2+n4;
% Solve the linear system with the Thomas algorithm
Ttot = thomas(e,f,g,Ttot,ntot);


%end temp_total.m
```

```
%
% % Fill the increasing foam density portion
%     dens_step = (1-dens_mult)/(ng_two);
%     index = 1;
% for i = n4+np+ng_one+nlf+1:n4+np+ng_one+nlf+ng_two
%     fd =
(dens_mult+(index*dens_step))*nextfoamdensity(Ttot(i),Ttot_p(i));
%     fcp = nextheatf(Ttot(i),Ttot_p(i));
%     fk = (dens_mult+(index*dens_step))*nextcondf(Ttot(i),Ttot_p(i));
%     H =
(((dens_mult+((index+1)*dens_step))*nextcondf(Ttot(i+1),Ttot_p(i+1))-
(dens_mult+((index-1)*dens_step))*nextcondf(Ttot(i-1),Ttot_p(i-
1)))/4)+fk*((1/((rp_by_dx)+i-2))-1);
%     G =
(((dens_mult+((index+1)*dens_step))*nextcondf(Ttot(i+1),Ttot_p(i+1))-
(dens_mult+((index-1)*dens_step))*nextcondf(Ttot(i-1),Ttot_p(i-
1)))/4)+fk*((1/((rp_by_dx)+i-2))+1);
%     R = dt/(fd*fcp*(dx_sq));
%     e(i) = R*H;
%     f(i)=1+(2*R*fk);
%     g(i) = -R*G;
%     index = index +1;
% end
```

```
% thomas.m
% Solves the system Ax=g for x using the Thomas algorithm,
% assuming A is tridiagonal and diagonally dominant.  It is
% assumed that (a,b,c,g) are previously-defined vectors of
% length n, where a is the subdiagonal, b is the main diagonal,
% and c is the superdiagonal of the matrix A.  The vectors
% (a,b,c) are replaced by the m_i and U on exit, and the vector
% g is replaced by the solution x of the original system.

%  -------------- FORWARD SWEEP --------------
function[g]=thomas(a,b,c,g,n);
for j = 1:n-1,   % For each column j<n,

   % Compute m_(j+1).  Note that we can put m_(j+1) in the location
   % (below the diagonal!) that a_(j+1) used to sit without disrupting
   % the rest of the algorithm, as a_(j+1) is set to zero by construction
   % during this iteration.

   a(j+1)     = - a(j+1) / b(j);

   % Add m_(j+1) times the upper triangular part of the j'th row of
   % the augmented matrix to the (j+1)'th row of the augmented
   % matrix.

   b(j+1) = b(j+1) + a(j+1) * c(j);
```

```
   g(j+1) = g(j+1) + a(j+1) * g(j);
end


%  ------------ BACK SUBSTITUTION ------------
```

```
% Function used to find the node where melting is occuring
function[Lm] = melt(Tn)

global n4 n1 dx3 dx2
T_test = 21;
count = 0;
while T_test >= 19.99;
    count = count +1;
    T_test = Tn(n4-(count-1));
end
Lm = (count-1)*dx3;
```

```
% brent_vol.m
% Input assumes {x1,x2,x3} are a bracketing triple with function
% values {J1,J2,J3}.  On output, x2 is the best guess of the minimum.
%
```

```
g(n) = g(n) / b(n);
for i = n-1:-1:1,
   g(i) = ( g(i) - c(i) * g(i+1) ) / b(i);
end
% end thomas.m
```

```
% The variables x_tol must also be predefined to indicate the desired
% tolerance of the answer in x.
function [Pres_final] = brent_vol(Pressure);
[x1, x2, x3, J1, J2, J3] = find_triplet_vol(Pressure);
x_tol = 10;
evals = 0;
      CGOLD=.3819660;  ITMAX=50;  D=0;
      FW=min(J1,J3);
      if FW == J1
         W =x1;
         V =x3;
         FV=J3;
      else
         W =x3;
         V =x1;
         FV=J1;
      end
      X =x2;
      FX=J2;
      A =min(x1,x3);
      B =max(x1,x3);
      FLAG3 = 0;
      for iter=1:ITMAX,
        if iter <= 2
            E=2.*(B-A);
        end
           XM=0.5*(A+B);
        if abs(X-XM)<=(2.*x_tol-.5*(B-A))
           FLAG3=1;
           break;
        end
        FLAG2 = 0;
        if abs(E) > x_tol | iter <= 2
          R=(X-W)*(FX-FV);
          Q=(X-V)*(FX-FW);
          P=(X-V)*Q-(X-W)*R;
          Q=2.*(Q-R);
          if Q > 0.
             P=-P;
          end
          Q=abs(Q);
          ETEMP=E;
          E=D;
          if ~(abs(P) >= abs(0.5*Q*ETEMP) | P <= Q*(A-X) | P >= Q*(B-X))
             D=P/Q;
             U=X+D;
             if U-A < 2.*x_tol | B-U < 2.*x_tol
                D=abs(x_tol)*sign(XM-X);
             end
             FLAG2 = 1;
          end
        end
        if FLAG2 == 0
           if X >= XM
             E=A-X;
           else
             E=B-X;
           end
```

```
            D=CGOLD*E;
          end
          if abs(D) >= x_tol
             U=X+D;
          else
             U=X+abs(x_tol)*sign(D);
          end
          FU= Volume_change(U);%compute_J(U);
          evals=evals+1;
          %plot(U,FU,'ko');  pause;
          if FU <= FX
            if U >= X
              A=X;

              B=U;
            end
            if FU <= FW | W == X
              V=W;
              FV=FW;
              W=U;
              FW=FU;
            elseif FU <= FV | V == X | V == W
              V=U;
              FV=FU;
            end
          end
        end
        if FLAG3==0
           t='Line minimization algorithm did not converge to prescribed
tolerance.'
        end
        x2=X; J2=FX;
        Pres_final = x2;

% end brent.m
```

```
            else
              B=X;
            end
            V=W;
            FV=FW;
            W=X;
            FW=FX;
            X=U;
            FX=FU;
          else
            if U < X
              A=U;
            else
```

find_triplet_vol.m

```
% Initialize and expand a triplet until the minimum is bracketed.
% Should work if J -> inf as |x| -> inf.
function [x1, x2, x3, J1, J2, J3] = find_triplet_vol(Pressure);

[x1,x2,x3] = init_triplet_vol(Pressure);

J1=Volume_change(x1);  J2=Volume_change(x2);  J3=Volume_change(x3);

    while (J2>J1)
        % Compute a new point x4 to the left of the triplet
        x4=x1-2.0*(x2-x1);
        J4=Volume_change(x4);
        % Center new triplet on x1
        x3=x2;      J3=J2;
        x2=x1;      J2=J1;
        x1=x4;      J1=J4;
    end
    while (J2>J3)
        % Compute new point x4 to the right of the triplet
        x4=x3+2.0*(x3-x2);
        J4=Volume_change(x4);
        % Center new triplet on x3
        x1=x2;      J1=J2;
        x2=x3;      J2=J3;
        x3=x4;      J3=J4;
    end
xf=x3;
% end find_triplet.m
```

```
global Edt pois_dt Ep pois rdt rp tp Vol_ch melt_cur therm_p Tp_ave
Tplas_ave

% Calculate the deflection of the DT shell based on guessed pressure
Def_DT_star = (Pressure_load_star*(rp-melt_cur)/Edt)*(((1-
pois_dt)*(rdt^3+2*rp^3)/(2*(rp^3-rdt^3)))-pois_dt);
% Calculate temporary change in volume due to deflection and guessed
pressure
% Calculate the deflection of the polymer shell based on guessed pressure
Def_poly_star = Pressure_load_star*(rp^2)*(1-
pois)/(2*Ep*tp)+(rp*therm_p*(Tplas_ave-Tp_ave

Vol_def = (4*pi/3)*((rp+Def_poly_star)^3-rp^3+(rp-melt_cur)^3-(rp-
melt_cur-Def_DT_star)^3);

% Calculate difference in calculated volume change
delta_Vol = abs(Vol_def-Vol_ch);
```

```
% init_triplet_T.m Initializes guess for bracketing
% triplet

function [x1, x2, x3] = init_triplet_vol(x);

    x1=x;
    x2=x+200;
    x3=x+300;

% end init_triplet.m
```

```
function[delta_Vol] = Volume_change(Pressure_load_star)
```

```
% Function used to evaluate the mass flux and heat flux
% due to evaporation at the solid/vapor or liquid vapor
% interface
function[q_vap, M2]=vapor_flux(Ttot,Ttot_p,Mo,V);

global R dt A n4 np Tp_ave therm_p rp Ep pois revap_inst tp f_He n_He
kbolt Press_hel_guess

% Define some average temeratures over the time step
Tn_star = Ttot(n4); % The liq/vapor surface temperature
To_star = Ttot(n4+1); % plastic shell inner wall temp
Tgas = (Tn_star+To_star)/2; % Gas temp

% Calculate the DT vapor pressure in the gap assuming it is saturated at t
= n+1
p_DT = (Tgas/Tn_star)^(1/2)*vapor_pressure(Tn_star);
p_sat = vapor_pressure(Tn_star);

sigma_e = 1; % Evaporation coefficient
sigma_c = 1; % Condensation conefficient

M2 = p_sat*V/(R*Tgas^(1/2)*Tn_star^(1/2));

%%%% Calculate the average mass flux over the time step
j = (M2-Mo)/(A*dt);

% % % Determine the avereage latent heat of fusion (J/kg)
% if Tn_ave < 19.78
%     h = -
5.449134*(Tn_ave^3)+79.50222*(Tn_ave^2)+3764.932*(Tn_ave)+250197.3;
% else
%     if (Tn_ave >= 19.78 & Tn_ave < 40.01)
%     h = -20.04217*(Tn_ave^3)+1150.727*(Tn_ave^2)-
23450.92*(Tn_ave)+441241.6;
%     else
%         if Tn_ave  > 40.01
%             h = 0;
%         end
%     end
%
% end
h =0;
q_vap = 0;%h*j; % Net heat flux out by evaporation/condensation
% end vapor_flux.m
```

REFERENCES

1. A.A. Harms, et. al., *Principles of Fusion Energy*, World Scientific Publishing, London, 2000.

2. R. Cook, "Creating Microsphere Targets for Inertial Confinement Fusion Experiments," Energy and Technology Review, April 1995.

3. R.W. Petzolt, et. al., "Direct Drive Target Survival During Injection in an Inertial Fusion Energy Power Plant," submitted to Nuclear Fusion, August 2002.

4. N. P. Siegel, "Thermal Analysis of Inertial Fusion Energy Targets," Master of Science Thesis, San Diego State University, May 2000.

5. J.G. Collier, *Convective Boiling and Condensation*, p. 317-322, McGraw-Hill International Book Co., New York ; London, 1981.

6. R. F. Brown, D. M. Trayer, and M. R. Busby, "Condensation of 300 – 2500 K Gases on Surfaces at Cryogenic Temperatures," The Journal of Vacuum Science and Technology, Vol. 7, No. 1, 1969.

7. J. P. Dawson, and J. D. Haygood, " Cryopumping," Cryogenics, Vol. 5, No. 2, April 1965.

8. C. R. Arumainayagam, et. al., "Adsorbate-assisted adsorption: Trapping dynamics of Xe on Pt(111) at nonzero coverages," Journal of Chemical Physics, Vol. 97, No. 7, October 1991.

9. W. G. Vincenti, C. H. Kruger, *Introduction to Physical Gas Dynamics*, Krieger Publishing, Florida, 2002.

10. W. Frost, *Heat Transfer at Low Temperatures*, Plenum Press, New York, 1975.

11. DS2V Version 2.1, GAB Consulting.

12. M. L. Klein and J. A. Venables, *Rare Gas Solids*, Academic Press, New York; London, 1976-1977.

13. G. Hartwig, *Polymer Properties at Room and Cryogenic Temperatures*, Plenum Press, New York, 1994.

14. Proceedings of the 7th Symposium on Thermophysical Properties, V.7, 1977.

15. B. Wunderlich and H. Baur, "Heat Capacities of Linear High Polymers," Advances in Polymer Science, Vol. 7, 1970.

16. J. E. Mark, *Polymer Data Handbook*, Oxford University Press, New York, 1999.

17. J. E. Mark, *Physical Properties of Polymers Handbook*, AIP Press, Woodbury, New York, 1996.

18. P. C. Souers, *Hydrogen Properties for Fusion Energy*, University of California Press, Berkley, 1986.

19. B. I. Verkin, *Handbook of Properties of Condensed Phases of Hydrogen and Oxygen*, Hemisphere Publishing Corp., New York, 1991.

20. A. C. Ugural, *Stresses in Plates and Shells*, WCB/McGraw Hill, Boston, 1999.

21. W. C. Young, *Roark's Formulas for Stress & Strain*, McGraw-Hill, New York, 1989, p. 640.

22. Y. A. Cengel and M. A. Boles, *Thermodynamics an Engineering Approach*, WCB/McGraw-Hill, Boston, 1998.

23. M. N. Özisik, *Finite Difference Methods in Heat Transfer*, CRC Press, Boca Raton, 1994.

24. C. Bonacina, et. al., "Numerical Solution of Phase-Change Problems," Int. J. Heat Mass Transfer, Vol. 16, pp. 1825-1832, 1973.

25. M. N. Özisik, *Heat Conduction*, Wiley, New York, 1993.

26. Bulgakova, N.M., Bulgakov, A.V., "Pulsed laser ablation of solids: transition from normal vaporization to phase explosion," Appl. Phys. A, 73, 2001, pp. 199-208.

27. S. Van Stralen, R. Cole, *Boiling Phenomena*, Hemisphere Publishing Corp., Washington, 1979.

28. R. W. Petzoldt, et al., "Direct-Drive Target Survival During Injection in an Inertial Fusion Energy Power Plant," submitted to Nuclear Fusion, August 2002.

29. A. R. Raffray, et al., "Enhancing Target Survival," Presentation given at the December 2002 HAPL meeting, http://aries.ucsd.edu/HAPL/.

30. A. R. Raffray, J. Pulsifer, M. S. Tilack, "Target Thermal Response to Gas Interactions," UCSD Technical Report, UCSD-ENG-092, http://aries.ucsd.edu/FERP/reports.shtml.

31. S. E. Bodner , D. G. Colombant, A. J. Schmitt, and M. Klapisch, "High-Gain Direct-Drive Target Design for Laser Fusion," Physics of Plasmas, 7(6), June 2000, pp. 2298-2301.

32. J. Grun, et al., "Rayleigh-Taylor Instability Growth Rates in Targets Accelerated with a Laser Beam Smoothed by Induced Spatial Incoherences," Phys. Rev. Lett. 58(25), June 1987, pp. 2672-2676.

33. C. J. Pawley, et al., “Observation of Rayleigh-Taylor growth to short wavelengths on NIKE,” Phys. Plasmas, 6(2), February 1999, pp. 565-570.

34. J. P. Knauer, et al., “Single-mode, Rayleigh-Taylor growth-rate measurements on the OMEGA laser system,” Phys. Plasmas, 7(1), January 2000.

35. A. R. Raffray, et al., “IFE chamber walls: requirements, design options, and synergy with MFE plasma facing components,” Journal of Nuclear Materials, 313-316 (2003), pp. 23-31.

36. K. Denpoh, “Modeling of Rarefied Gas Heat Conduction Between Wafer and Susceptor,” IEEE Transactions on Semiconductor Manufacturing, 11(1), February 1998, pp. 25-29.

37. J. Hoffer, “Update on Solid DT Studies,” Presentation given at the April 2002 HAPL meeting, http://aries.ucsd.edu/HAPL/MEETINGS/0304-HAPL/hoffer.ppt.

38. S. G. Kandlikar, M. Shoji, V. K. Dhir, *Handbook of Phase Change*, Taylor & Francis, Philadelphia, PA , 1999.

BIBLIOGRAPHY

Boiling Phenomena, S. Van Stralen, R. Cole, Hemisphere Publishing Corp., Washington, 1979.

Handbook of Phase Change, Satish G. Kandlikar, Masahiro Shoji, Vijay K. Dhir, Taylor & Francis, Philadelphia, PA , 1999.

Finite Difference Methods in Heat Transfer, M. N. Özisik, CRC Press, Boca Raton, 1994.