*OPTIMIZATION TECHNIQUES APPLIED TO PASSIVE MEASURES FOR IN-ORBIT SPACECRAFT SURVIVABILITY: CONTRACT NAS8-37378*

*~~FINAL~~ REPORT*
INTERIM

**PREPARED FOR:**

**GEORGE C. MARSHALL SPACE FLIGHT CENTER**

**MARSHALL SPACE FLIGHT CENTER, AL 35812**

**PREPARED BY:**


**ROBERT A. MOG**

**D. MARVIN PRICE**


**JUNE 1991**


(NASA-CR-184198) OPTIMIZATION TECHNIQUES APPLIED TO PASSIVE MEASURES FOR IN-ORBIT SPACECRAFT SURVIVABILITY Interim Report, period ending 30 Jun. 1991 (Science Applications International Corp.) 192 p

N91-31204

Unclas

G3/18 0040115



SAIC An Employee-Owned Company

**Science Applications International Corporation**

*6725 Odyssey Drive, Huntsville, AL 35806-3301 • (205) 971-6400*

SAIC
Science Applications International Corporation
An Employee-Owned Company

28 June 1991

NASA/MSFC
Marshall Space Flight Center, Alabama 35812

Attention: AP52-D

Subject: Final Report for Period Ending 6-30-91

Reference: Contract Number NAS8-37378

Gentlemen:

Enclosed is a copy of "Optimization Techniques Applied to Passive Measures for In-Orbit Spacecraft Survivability" Final Report, with distribution as noted below. Questions should be directed to the undersigned at 971-6736.

Sincerely,

SCIENCE APPLICATIONS INTERNATIONAL CORPORATION

Robert A. Mog

Robert A. Mog, Ph. D.
Principle Investigator

RAM/vem

Attachment a/s

cc: ED52 (1)
AP52-D (1)
AS24D (3)
AT01 (1)
CC01/Wofford (1)
KA11 (3)
KA02 (2)
NASA Scientific & Technical
Information Facility (2)

BM 1149 VM

6725 Odyssey Drive, Huntsville, Alabama 35806, (205) 971-6400
Other SAIC Offices: Albuquerque, Atlanta, Boston, Chicago, Colorado Springs, Dayton, Denver, El Paso, Las Vegas, Los Angeles, Norfolk, Oak Ridge, Omaha, Orlando, St. Ann, San Diego, San Francisco, Santa Barbara, Shalimar, Tucson and Washington, DC

## ACKNOWLEDGMENTS

We would like to thank Sherman Avans, Jennifer Robinson, Joel Williamsen, and Ben Hayashida of the Marshall Space Flight Center for their direction and contributions to these developments under NASA Contract NAS8-37378. In particular, we thank Sherman Avans for suggesting many of the initial features of PSDOC, Jennifer Robinson for recommending the approach and numerous trades for the advanced shielding concepts, Joel Williamsen for suggesting many of the features and development approaches for the Monte Carlo simulation tool, and Ben Hayashida for suggesting various improvements to PSDOC.

# Table of Contents

## LIST OF SYMBOLS

$a_{ij}$ = exponent for objective function term i and variable j

$a_{ijl}$ = exponent for term i, variable j, in constraint l

$A$ = spacecraft space debris area

$A_l$ = acceleration factor of primal penalty function for constraint l

$B$ = spacecraft orientation factor

$c_i$ = coefficient for objective function term i

$c_{il}$ = coefficient for term i in constraint l

$c_{ir}$ = coefficient for posyseparable term i

$C$ = bumper material speed of sound

$D$ = projectile diameter

$DOD$ = geometric programming degree of difficulty

$f$ = non-normalized impact velocity distribution

$f_n$ = normalized impact velocity distribution

$F$ = space debris flux

$F_h$ = fraction of hyperspace for random search

$g_l$ = constraint l

$h$ = spacecraft altitude

$i$ = spacecraft inclination

$k$ = number of independent variables

$K_l$ = right hand side of primal constraint l

$L_2$ = wall material constant

$m$ = projectile mass

$m_k$ = number of random search points

$m_l$ = number of terms in constraint l

$n$ = number of terms in objective function

$n_j$ = positive integer value corresponding to variable j

$N$ = cumulative space debris flux

$N_1$ = number of walls penetrated (normal impact)

$N_t$ = total meteoroid flux

$p$ = number of constraints

$P$ = space debris growth rate

$P_k$ = required confidence for random search

$P_0$ = spacecraft probability of no penetration

$q$ = number of discrete variables

$r_j$ = discrete availability factor for variable j

$s$ = solar flux

$S$ = bumper/wall separation

$t_1$ = bumper thickness

$t_2$ = wall thickness

$T$ = mission duration

$V$ = projectile impact velocity

$V_{max}$ = maximum space debris impact velocity

$W$ = structure mass per unit area or weight

$\alpha_l$ = acceleration factor of primal penalty function for discrete constraint l

$\delta_i$ = dual variable corresponding to objective function term i

$\delta'_{jl}$ = dual variable corresponding to term j in constraint l

$\delta_l$ = binary factor of primal penalty function for constraint l

$\delta_{1j}$ = first dual variable for discrete constraint of variable j

$\delta_{2j}$ = second dual variable for discrete constraint of variable j

$\Delta_l$ = binary factor of primal penalty function for discrete constraint l

$\varepsilon$ = convergence parameter for penalty function

$\varepsilon_1$ = initial exploratory step size for Hooke and Jeeves

$\varepsilon_2$ = final exploratory step size for Hooke and Jeeves

$\theta$ = impact angle from surface normal

$\mu_l$ = dual objective function variable in constraint l

$\nu$ = dual objective function

$\phi$ = primal penalty function

$\rho_1$ = bumper density

$\rho_2$ = wall density

$\rho_p$ = projectile mass density

$\psi$ = spacecraft inclination factor

$[\ ]_{n.i.}$ = nearest integer of quantity in brackets

A 0 subscript denotes optimal value for a primal variable.

A * superscript denotes optimal value for a dual variable.

# 1 INTRODUCTION

## 1.1 Problem Statement

Spacecraft designers have been concerned since the 1960's about the effects of meteoroid impacts on mission safety. Recent concerns have extended to the space debris environment, which typically displays more massive particles than the meteoroid environment for the same risk level. Additionally, the higher exposure area-time product of future space missions (e.g., Space Station) poses a more critical design problem than current short term missions. Finally, the inherent uncertainties in projectile mass, velocity, density, shape, and impact angle make the traditional deterministic design approach impractical.

The engineering solution to this design problem has generally been to erect a bumper or shield placed outboard from the spacecraft wall to disrupt/deflect the incoming projectiles. This passive measure has resulted in significant structural weight savings relative to a single wall concept with the same protective capability. The problem, then, is how to efficiently design these protective structures so that the bumper disrupts the projectile without posing a lethality problem to the wall protecting the crew and equipment.

Spacecraft designers have a number of tools at their disposal to aid in the design process. These include hypervelocity impact testing, analytic impact predictors, and hydrodynamic codes. Perhaps the most widely accepted of these tools is impact testing, which has the advantage of providing actual spacecraft design verification. On the other hand, maximum test velocities are currently limited (8 km/sec) relative to maximum space debris (about 15 km/sec) and meteoroid (about 72 km/sec) velocities. Also, extensive testing is required to develop statistically significant trends for the large number of parameters associated with hypervelocity impact. Hydrodynamic code analysis can overcome the velocity limitation problem. However, this method is very computer (and time) intensive, and there is a fair amount of controversy involved in the selection of appropriate codes and code-specific parameters.

Analytic impact predictors generally provide the best quick-look estimate of design tradeoffs. Their use is constrained by the limitations of the testing from which they are experimentally derived, the assumptions used in their theoretical derivation, or the regression analysis used in their statistical formation. However, analytic predictors may provide information that is clearer than that obtained from the examination of experimental results.

The most complete way to determine the characteristics of an analytic impact predictor is through (nonlinear) optimization of the protective structures design problem formulated with the predictor of interest. Optimization techniques provide analytic or numerical solutions depending on the nature of the predictor, the problem formulation, and the technique used.

## 1.2 Contract Purpose

The purpose of this contract is to provide Space Station FREEDOM protective structures design insight through the coupling of design/material requirements, hypervelocity impact phenomenology, meteoroid and space debris environment sensitivities, optimization techniques and operations research strategies, and mission scenarios. Major findings from contract inception to the beginning of this study are detailed in References 100-105 and are shown below:

| | |
|---|---|
| | ***PROTECTIVE STRUCTURES DESIGN*** |
| *1* | The Nysmith Predictor Has a Systemic Inequality Constraint. |
| *2* | All Predictors Investigated Show a Large Relative Incentive for Increasing Bumper/Wall Separation from 10 to 15 cm. (Shift to Knee) |
| *3* | All Predictors Investigated Show a Large Relative Disincentive for Increasing System Probability of No Penetration. (Already at Knee) |
| *4* | Optimal Design Ratios Vary With Mission, Requirements, Environment, and Materials Parameters. |
| *5* | Variations in Bumper/Wall Materials Show Large Design/Weight Differentials. (Even Among Aluminum Alloys only) |
| *6* | Optimal Bumper Thickness is Most Heavily Influenced by Projectile Melt/Vaporization Region While Optimal Wall Thickness is Most Heavily Influenced by Projectile Shatter Region. |
| *7* | Posynomial Programming May Be Useful in Predicting Design Trends as Functions of Estimated Regression Parameters Before Testing. |
| | ***ENVIRONMENT SENSITIVITY*** |
| *8* | Debris Dominates Meteoroids From a Design Standpoint, But Optimal Ratios are Considerably Different. |
| *9* | Debris Growth Rate, Mission Altitude, Schedule, Safety, and Duration All Have Significant Effects on Optimal Design Values and Ratios. |
| | ***OPTIMIZATION APPROACHES*** |
| *10* | Global Protective Structures Design Optimization Is Achievable Using Many Hypervelocity Impact Predictors (e.g. Nysmith, Burch, Wilkinson, Madden, Maiden, 42 Test Sub-database). |
| *11* | Differences Between Global and Local Design Optimization May Result in Large Weight Differentials. |
| *12* | The Power of the Geometric Programming Optimization Method Increases With Increasing Design Complexity (More Bumpers, Materials, etc.). |
| *13* | Material Properties Optimization Can Be Achieved Using a Hooke and Jeeves Pattern Search Approach. |
| *14* | Discrete Protective Structures Design Optimization Can Be Efficiently Performed Using Dual and Primal Methods. |
| | ***STATISTICAL ANALYSES*** |
| *15* | Posynomial Regression Can Be Performed To a Statistically Significant Level for Hypervelocity Impact Test Databases. |

## 1.3 Study Goals

The goals of this study are to:

1. Perform Space Station protective structures design sensitivities relative to the "new" space debris environment definition.
2. Incorporate the unique methodology developed into a user-friendly, menu-driven PC tool.
3. Begin development of a Monte Carlo simulation tool which will provide top level insight for Space Station protective structures designers.
4. Assess the hypervelocity impact test samples from a damage/penetration standpoint.
5. Analyze projectile shape effects on protective structures design.

The period of performance for this effort is 2-28-90 through 6-30-91.

Additional goals not included in the Scope of Work are:

6. Perform preliminary advanced shielding development work in the area of multiple bumper configurations.
7. Develop discrete protective structures design optimization methods.

## 1.4 Study Approach

The methodology presented in this study is sufficiently general for application to various spacecraft configurations and impact environments. The baseline scenario investigated is for the Space Station Core Module Configuration and space debris environment with the following specifications: 5% space debris growth rate; Space Station operation period from 1995-2004; 460 km Space Station altitude; 28.5 degree Space Station inclination; 0.97 total Core Module Configuration probability of no penetration; 588 $m^2$ total Core Module Configuration debris area; 10 cm bumper/wall separation; 0 degree impact angle (normal); 6061-T6 aluminum alloy bumper; 2219-T87 aluminum alloy wall; and 9 km/sec average impact velocity.

Because other approaches involve the analysis of existing protective structures designs, the design methodology presented here is unique. The process begins with the definition of the space debris environment to determine the critical design projectile diameter and density. The design problem is then formulated in terms of a hypervelocity impact predictor as a weight minimization function of the independent (or designer controllable) variables. These variables generally include bumper/wall material properties and thicknesses. The protective structures system is then globally optimized using the Geometric Programming technique. Sensitivity analyses are performed to investigate the effect of changes in the system parameters on the optimal design. Several hypervelocity impact predictors are analyzed, including the Wilkinson, Burch, PEN4, and Nysmith models, as well as combinations of these models.

## 1.5 Study Results

1. Goal 1 was completed using technology-specific optimization techniques. Results are given in Section 2.
2. The user-friendly, menu-driven PC tool of Goal 2 is called PSDOC (Protective Structures Design Optimization Code) and was delivered with documentation in August 1990. Several updated versions have been delivered in the interim. An overview of this tool is presented in Section 3.
3. The Monte Carlo simulation tool of Goal 3 has been planned and is currently under development. This is discussed in Section 4.
4. Hypervelocity impact samples (Goal 4) have been evaluated as discussed in Section 7. SAIC has also performed additional posynomial regression analyses on hypervelocity impact test data which has been delivered in a White Paper and as part of a Dissertation.
5. SAIC has developed appropriate regression and optimization tools to satisfy Goal 5. This is presented in Section 8.
6. The development of advanced shielding concepts has been performed (Goal 6), and will continue to be performed as part of future work on this contract. Section 5 includes results from this effort.
7. The development and applications of discrete protective structures design optimization techniques (Goal 7) is complete and is presented in Section 6.

## 1.6 Major Findings of This Study

| |
|---|
| 1. Global analytic nonlinear design optimization can be performed for the projectile melt/vaporization region (Wilkinson), for normal impacts in the projectile shatter region (Burch), and for the Nysmith predictor using Geometric Programming. |
| 2. For the predictors investigated, the optimal ratio of bumper mass per unit area to total mass per unit area may vary with mission, environment, projectile mass, and velocity regime. |
| 3. There is a large incentive for increasing the bumper/wall separation from 10 to 15 cm for all predictors investigated. |
| 4. All predictors reflect increasing design sensitivity to projectile diameter and decreasing design sensitivity to bumper/wall separation. |
| 5. The Wilkinson and Nysmith predictors reflect increasing design sensitivity to projectile velocity, while the Burch relationship is decreasing. |
| 6. For the combined predictors, 2011-T8 is the preferable aluminum alloy bumper choice for the baseline parameters. |
| 7. For the combined predictors, increasing the bumper/wall separation from 10 to 15 cm reduces the minimum module weight by 25%. |
| 8. Minimum CMC weight is very sensitive to space debris growth rate above 7% and Space Station altitude below 1000 km for the combined predictors. |
| 9. CMC protective structures design depends greatly on mission duration for the combined predictors. |
| 10. For the combined predictors, increasing the CMC mission risk from 3% to 5% reduces the minimum module weight by about 30%. |
| 11. Global (and sometimes analytic) optimization of discrete posynomial programs can be performed using dual approaches coupled with partial invariance techniques. |
| 12. Primal methods require less "pencil and paper" effort than dual methods and are more easily applied to most problems. |
| 13. Primal methods do not generally obtain global solutions for the discrete posynomial program. |
| 14. The dual method may be advantageous in cases where the objective function may be sufficiently separable, since posyseparable programs do not require solutions of coupled nonlinear equations in the dual-to-primal variable transformation. |
| 15. The Monte Carlo simulation tool is feasible from a development standpoint and appears to have advantages over current expected value models. |
| 16. An approximation to a nonstationary Poisson arrival process for impact events appears to be sufficient. |
| 17. Both the Wilkinson and ballistic PEN4 predictors may be extended to multiple bumper models. |
| 18. The multiple bumper Wilkinson predictor optimization problem is a 0 degree of difficulty posynomial programming formulation. |

# 2 ANALYSIS OF NEW SPACE DEBRIS ENVIRONMENTS

## 2.1 Earth Orbital Space Debris and Meteoroid Environs

The space debris environment model chosen for this study is due to Kessler[75]. The major dependencies considered involve space debris growth rate, spacecraft operational period, mission altitude and inclination, spacecraft debris area, orientation, and probability of no penetration.

The space debris flux is given by Kessler as

$$F(D,h,i,t,s) = B\phi(h,s)\psi(i)(F_1(D)g_1(t) + F_2(D)g_2(t)) \qquad [1]$$

where

$$\phi(h,s) = \phi_1(h,s)/(\phi_1(h,s)+1) \qquad [2]$$

$$\phi_1(h,s) = 10^{(h/200 - s/140 - 1.5)} \qquad [3]$$

$$F_1(D) = 1.05(10^{-5})/D^{2.5} \qquad [4]$$

$$F_2(D) = 7.0(10^{10})/(D+700)^6 \qquad [5]$$

$$g_1(t) = (1+2P)^{t-1985} \qquad [6]$$

$$g_2(t) = (1+P)^{t-1985} \qquad [7]$$

The spacecraft inclination factor for 28.5 degrees is 0.9135.

The cumulative flux N is given by

$$N = \int_0^T FA\,dt \qquad [8]$$

which may be approximated using one year intervals by

$$N = A \sum_{t=t_i}^{t_f} F(D,h,i,t,s(t)) \qquad [9]$$

A Poisson arrival rate for space debris gives

$$P_0 = e^{-N} \qquad [10]$$

A closed form solution for D may be accurately found for particle diameters much smaller than 700 cm. This is given by

$$D = \left( \frac{1.05(10^{-5})(G_1)}{-5.9499(10^{-7})G_2 - \frac{\ln(P_0)}{AB\psi(i)}} \right)^{0.4} \qquad [11]$$

where

$$G_j = \sum_{t=t_i}^{t_f} \phi(h, s(t))g_j(t) \text{ for j=1,2.} \qquad [12]$$

The average projectile mass density is given in gm/cm$^3$ by Kessler as

$$\rho_p = 2.8 \text{ for } D \le 1\text{cm} \qquad [13]$$

$$\rho_p = 2.8/D^{0.74} \text{ for } D > 1\text{cm} \qquad [14]$$

This relationship is shown in **Figure 2.1-1.**

For an orbital inclination of 28.5 degrees, the non-normalized impact velocity distribution is given by

$$f(V) = (14.46V - V^2)\left(18.7e^{-((V-18.07)/3.614)^2} + 0.67e^{-((V-9.505)/3.925)^2}\right) + 0.0116(28.91V - V^2) \qquad [15]$$

The normalized impact velocity distribution is given by

$$f_n(V) = \frac{f(V)}{\int_0^{\infty} f(V)dV} \qquad [16]$$

This distribution is shown in **Figure 2.1-2** for i = 28.5 degrees. Finally, the impact angle is given as a function of impact velocity as

$$\theta = \cos^{-1}(-V/15.4) \qquad [17]$$

This relationship is shown (with uncertainty bounds) in **Figure 2.1-3** for a surface parallel to the CMC velocity vector.

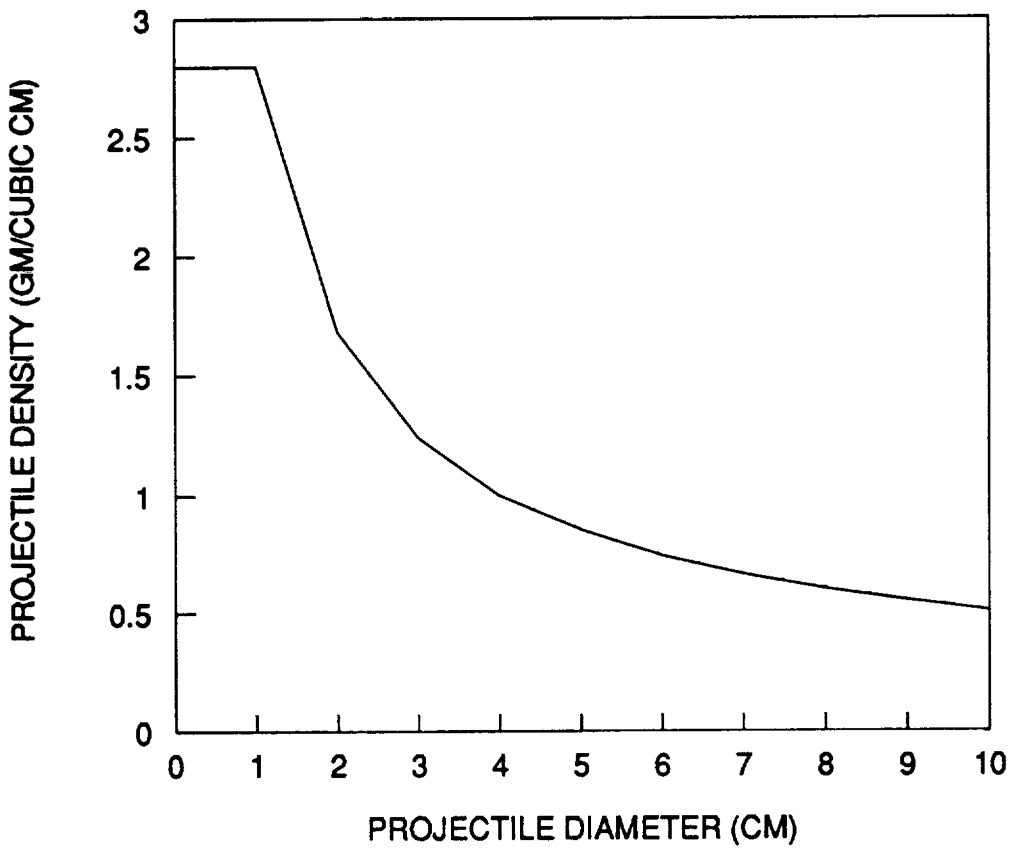



**Figure 2.1-1. Space Debris Particle Density vs Diameter**

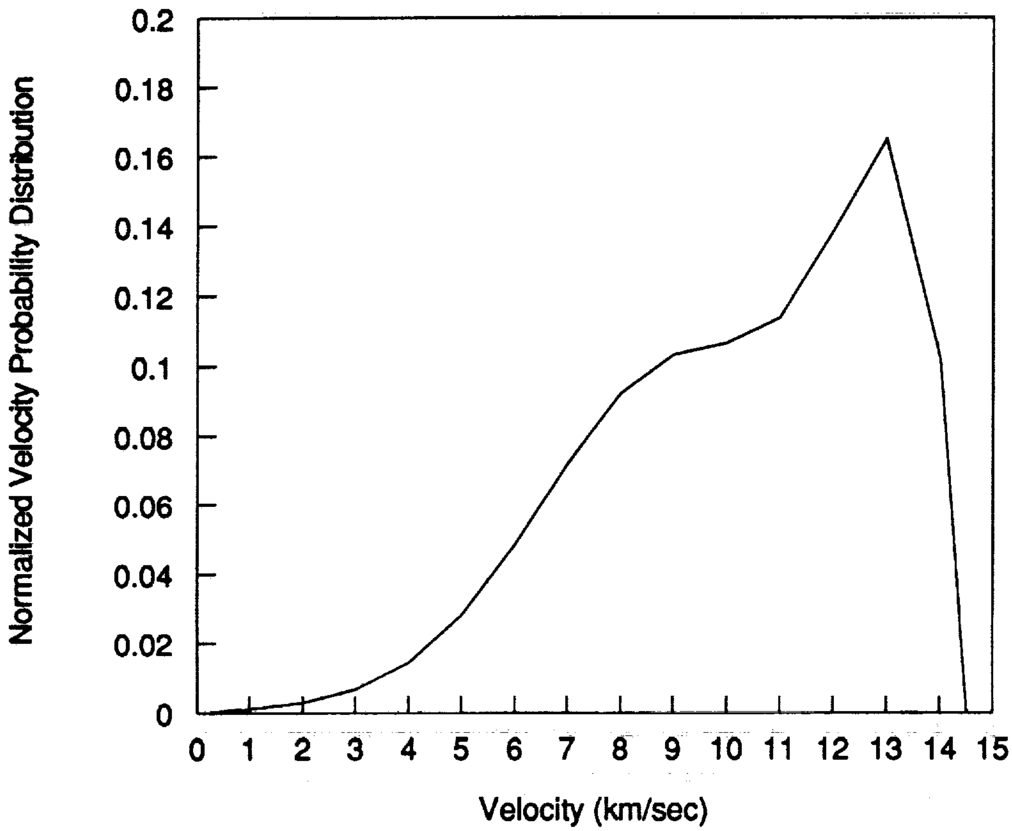



**Figure 2.1-2. Velocity Probability Distribution for 28.5 Degrees Inclination**

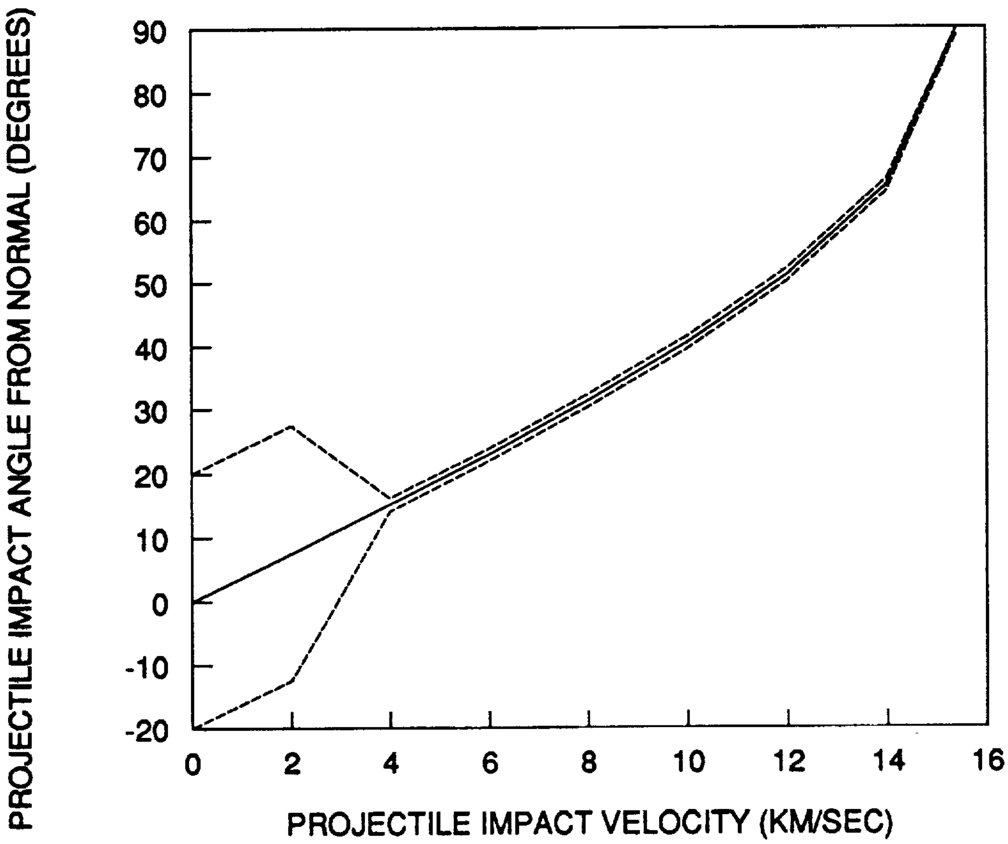



**Figure 2.1-3. Projectile Impact Angle From Normal of Surface Oriented Parallel to CMC Velocity Vector vs Impact Velocity**

The total meteoroid environment flux-mass model is given by Cour-Palais[32] as

$$\mathrm{Log}_{10}(N_t) = -14.339 - 1.584\,\mathrm{Log}_{10}(m) - 0.063(\mathrm{Log}_{10}(m))^2 \qquad [18]$$

for

$$m \in [10^{-12}, 10^{-6}]$$

and

$$\mathrm{Log}_{10}(N_t) = -14.37 - 1.213\,\mathrm{Log}_{10}(m) \qquad [19]$$

for

$$m \in [10^{-6}, 1]$$

with shielding factor

$$\eta = \frac{1+\cos(\phi)}{2} \qquad [20]$$

$$\sin(\phi) = \frac{R}{R+h} \qquad [21]$$

and gravitational defocussing factor

$$G = \frac{0.43}{N_R} + 0.57 \qquad [22]$$

where R is the radius of the shielding body (= 6378 km for Earth), and $N_R$ is the spacecraft range from the Earth's center in Earth radii. The velocity probability distribution for meteoroids is shown in **Figure 2.1-4**. The average mass density is 0.5 gm/cm$^3$, with average particle velocity of 20 km/sec.

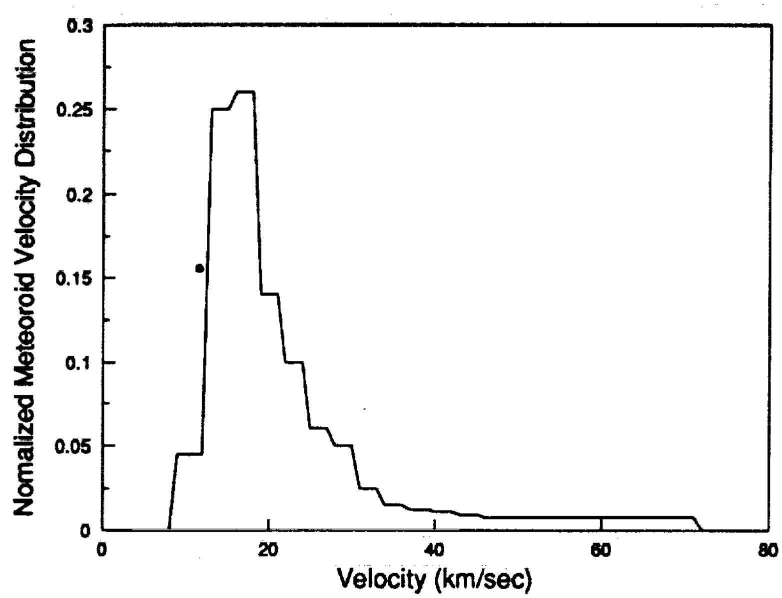



**Figure 2.1-4. Meteoroid Velocity Probability Distribution**

## 2.2 Measures of Design Effectiveness

The traditional measure of protective structures design effectiveness is the probability of no penetration of the pressure wall. This measure generally accounts for the risk associated with the particle size, impact velocity, and impact angle. It may also include spall factors to account for impact scenarios where penetration does not occur, but spallation does.

The probability of no penetration of a protective structure generally does not reflect uncertainties in the environment, in particular, the particle shape, density, and diameter. These uncertainties may be estimated by establishing confidence intervals about the expected probability of no penetration.

## 2.3 Potential Protective Structures Design Approaches

### Active Design

Active design includes debris mitigation and removal. Debris mitigation is the design of spacecraft and launch vehicles to minimize the amount of debris generated through operations. Debris removal includes the entrapment and/or possible destruction or disposal of debris.

### Passive Design

Passive protective structures design is the placement of shields permanently spaced outboard from the pressure wall to disrupt the incoming particle. One approach to providing design insight is through the use of Geometric Programming (GP).

GP is a particular nonlinear programming (NLP) technique formalized by Duffin, Peterson, and Zener[42] in 1967. It is practiced by engineers, scientists, and mathematicians alike. To appreciate the elements of GP requires a short mathematical presentation.

The prototype Geometric Programming problem is formulated in terms of posynomials -

polynomials with positive coefficients, positive-valued independent variables, and real exponents. The problem is to

$$\min f = \sum_{i=1}^{n} c_i \prod_{j=1}^{k} x_j^{a_{ij}} \qquad [23]$$

subject to

$$g_l = \sum_{i=1}^{m_l} c_{i_l} \prod_{j=1}^{k} x_j^{a_{ijl}} \le 1 \qquad l = 1,2,\ldots,p \qquad [24]$$

Obviously, this is a great restriction in applicability, since not all NLP problems may be formulated in terms of [23] and [24]. For problems of this form, including nonconvex programming problems, GP provides the globally optimal solution.

One approach to solving this problem is to consider the dual problem, as justified by the Arithmetic-Geometric Inequality. The dual Geometric Programming problem is given by

$$\max v(\delta) = \prod_{i=1}^{n}\left(\frac{c_i}{\delta_i}\right)^{\delta_i} \left(\prod_{l=1}^{p} \mu_l^{\mu_l} \left(\prod_{j=1}^{m_l} \left(\frac{c_{jl}}{\delta'_{jl}}\right)^{\delta'_{jl}}\right)\right) \qquad [25]$$

with

$$\sum_{i=1}^{n} \delta_i a_{ij} + \sum_{l=1}^{p}\left(\sum_{j=1}^{m_l} \delta'_{jl} a_{jql}\right) = 0 \qquad q = 1,2,\ldots,k \qquad [26]$$

$$\sum_{i=1}^{n} \delta_i = 1 \qquad [27]$$

$$\mu_l = \sum_{j=1}^{m_l} \delta'_{jl} \qquad l = 1,2,\ldots,p \qquad [28]$$

Clearly, equations [26]-[28] represent k+p+1 equations in $n+p+m_1+m_2+\ldots+m_p$ unknowns. If

$$k + 1 > n + \sum_{l=1}^{p} m_l \qquad [29]$$

then the system is overspecified. If, in addition, the system is inconsistent, then the problem formulation or model selection must be reconsidered. If

$$k+1=n+\sum_{l=1}^{p} m_l \tag{30}$$

and the system has nontrivial determinant, then a unique solution for the dual variables exists. If

$$k+1<n+\sum_{l=1}^{p} m_l \tag{31}$$

then the system is underspecified. The Geometric Programming degree of difficulty is given by

$$DOD=n-k-1+\sum_{l=1}^{p} m_l \tag{32}$$

Optimal dual variables for systems with positive degree of difficulty may be found by using a number of techniques, including search methods. Once the optimal dual variables are determined, they must be converted back to the primal variables using the relationships

$$f_0=\nu(\delta^*) \tag{33}$$

$$c_i \prod_{j=1}^{k} x_j^{a_{ij}}=\delta_i^* f_0 \qquad i=1,2,\ldots,n \tag{34}$$

$$\mu_l c_{il} \prod_{j=1}^{k} x_j^{a_{ijl}}=\delta_{il}^{*} \qquad l=1,2,\ldots,p \tag{35}$$

Note that this dual-to-primal conversion involves n+p nonlinear equations, and therefore represents a potentially difficult problem to solve in its own right.

Now, if the number of terms in the objective function (n) is large, and the number of independent variables (k) is small, a large degree of difficulty problem often ensues (particularly in a problem with few constraints). In these cases, solution of the dual problem may be quite lengthy, and a primal method may be in order. This strategy is further justified when gradient methods are used, because the first and second (and higher-ordered) partial derivatives of the independent variables are easily given as:

$$\frac{\partial f}{\partial x_l} = \sum_{i=1}^{n} c_i a_{il} x_l^{-1} \prod_{j=1}^{k} x_j^{a_{ij}} \qquad [36]$$

$$\frac{\partial^2 f}{\partial x_l \partial x_q} = \sum_{i=1}^{n} c_i a_{il} a_{iq} (x_l x_q)^{-1} \prod_{j=1}^{k} x_j^{a_{ij}} \qquad [37]$$

Based on the relatively large number of recent applied Geometric Programming articles, it is apparent that GP possesses a fairly high utility, particularly in the area of structural design. Because GP is the only NLP technique which offers the guarantee of a globally optimal solution for certain nonconvex problems, it should be considered more widely in practice. Additionally, for zero degree of difficulty problems, GP can provide an analytic optimal solution for the objective function and independent variables. This attribute provides greater insight for the system designer than that obtainable by other NLP techniques. Finally, the values of the dual variables may provide very crucial design information alone in terms of the physical parameters of the problem at hand.

Since its inception, GP has been widely applied to structural design optimization problems. These problems may involve dynamic and static loadings, both determinate and indeterminate. The posynomial property of weight minimization for structural design problems matches nicely with the GP technique. Additionally, since many structural design optimization problems include a large number of independent variables, this reduces the degree of difficulty for the GP process (see equation [32]).

Recently, GP has been found to be widely applicable to the optimization of spacecraft protective structures using analytic hypervelocity impact models.[100-105] The posynomial nature of these predictors is not unusual, since many physical phenomena may be attributed to a geometric model.

The basic optimization problem is a weight minimization problem of the protective structures. It has been shown[105] that for spacecraft structures with low curvature and relatively large diameter, it is sufficient to minimize the total mass per unit area given by

$$W = \sum_{i=1}^{2} \rho_i t_i \qquad [38]$$

In particular, this is true for the Space Station Core Module Configuration. Increasing the complexity of the weight objective function by accounting for specific configurations only serves to increase the complexity of the optimization technique and convergence time unnecessarily. No improvement in accuracy is achieved.

Three hypervelocity impact predictors, developed in the 1960's and displaying different attributes of Geometric Programming are due to Wilkinson[160], Burch[29] and Nysmith.[115]

The Wilkinson predictor is a piecewise differentiable model given by

$$t_2 = \frac{0.364 D^3 \rho_p V \cos(\theta)}{L_2 S^2 \rho_2} \quad \text{for} \quad \frac{D\rho_p}{\rho_1 t_1} \le 1, \qquad [39]$$

$$t_2 = \frac{0.364 D^4 \rho_p^2 V \cos(\theta)}{L_2 S^2 \rho_1 t_1 \rho_2} \quad \text{for} \quad \frac{D\rho_p}{\rho_1 t_1} > 1. \qquad [40]$$

Under condition [40], the dual Geometric Programming objective function is given by

$$v(\delta) = (\rho_1/\delta_1)^{\delta_1} (c_1/\delta_2)^{\delta_2} \qquad [41]$$

$$c_1 = \frac{0.364 D^4 \rho_p^2 V \cos(\theta)}{L_2 S^2 \rho_1} \qquad [42]$$

$$\delta_1 + \delta_2 = 1 \quad [43]$$

$$\delta_1 - \delta_2 = 0 \quad [44]$$

Equations [43] and [44] together imply

$$\delta_1 = \delta_2 = 1/2 \quad [45]$$

The minimum weight and globally optimal thicknesses are given by

$$W_0 = \frac{1.207 D^2 \rho_p \left( \frac{V\cos(\theta)}{L_2} \right)^{1/2}}{S} \quad [46]$$

$$t_{1_0} = \frac{0.604 D^2 \rho_p \left( \frac{V\cos(\theta)}{L_2} \right)^{1/2}}{S\rho_1} \quad [47]$$

$$t_{2_0} = \frac{0.604 D^2 \rho_p \left( \frac{V\cos(\theta)}{L_2} \right)^{1/2}}{S\rho_2} \quad [48]$$

Thus, the globally optimal algorithm for the Wilkinson Predictor is

1. Determine $t_{1_0}$ and $t_{2_0}$ from equations [47] and [48].
2. Compute $\frac{D\rho_p}{\rho_1 t_{1_0}}$.
3. If $\frac{D\rho_p}{\rho_1 t_{1_0}} > 1$, then quit. The optimal design is $(t_{1_0}, t_{2_0})$.
4. If $\frac{D\rho_p}{\rho_1 t_{1_0}} \leq 1$, the optimal design is $\left( t_{1_0}, t_{2_0} / \left( \frac{D\rho_p}{\rho_1 t_{1_0}} \right) \right)$.

**Figures 2.3-1, 2, and 3** show the optimal design values of minimum system mass per unit area, and optimal bumper and wall thicknesses vs projectile diameter, bumper/wall separation, and projectile velocity, respectively, for the Wilkinson predictor. In **Figure 2.3-1**, the projectile density varies with diameter according to equations [13] and [14]. In **Figure 2.3-3**, the impact angle remains constant at 0 degrees (normal). The optimal bumper and wall thicknesses for the Wilkinson predictor are approximately equal due to the similarity in bumper and wall material densities (see equations [47] and [48]).

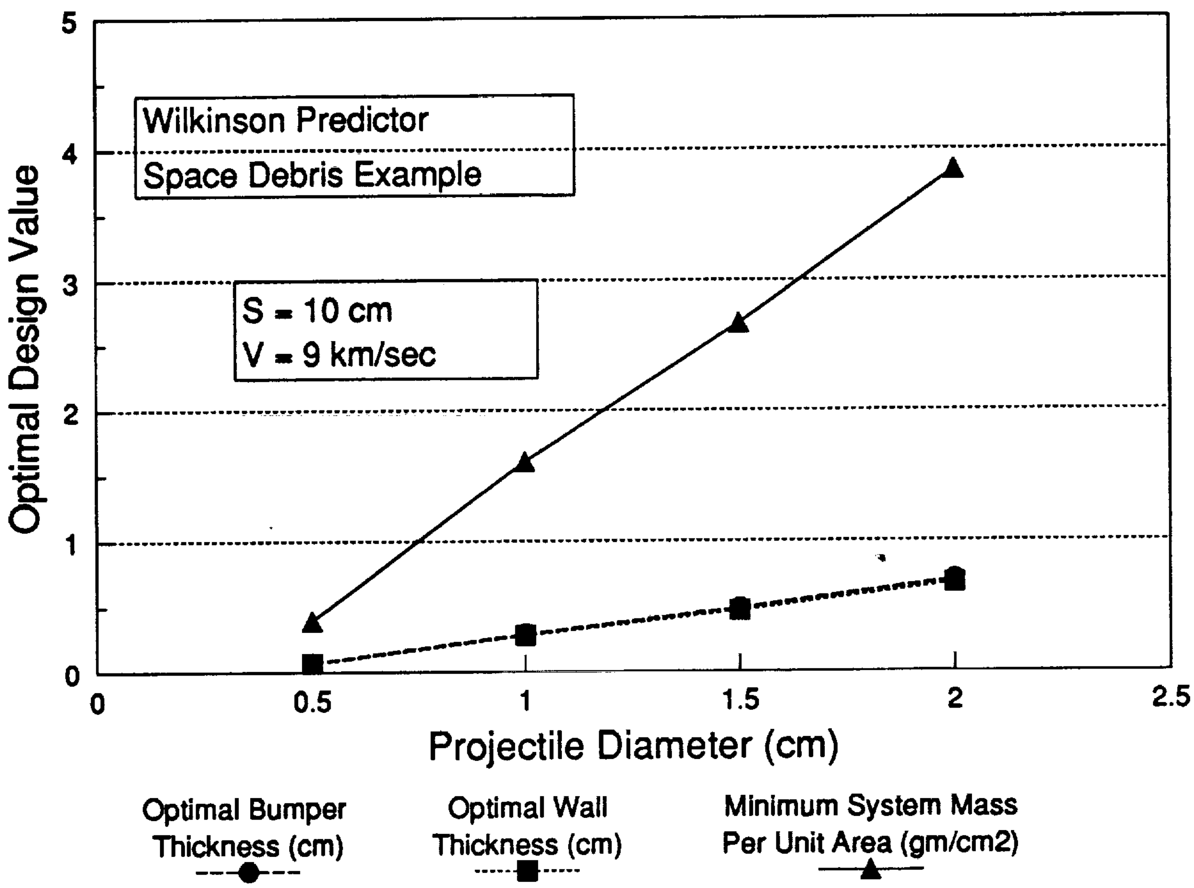



**Figure 2.3-1. Optimal Design Value vs Projectile Diameter for Wilkinson Predictor**

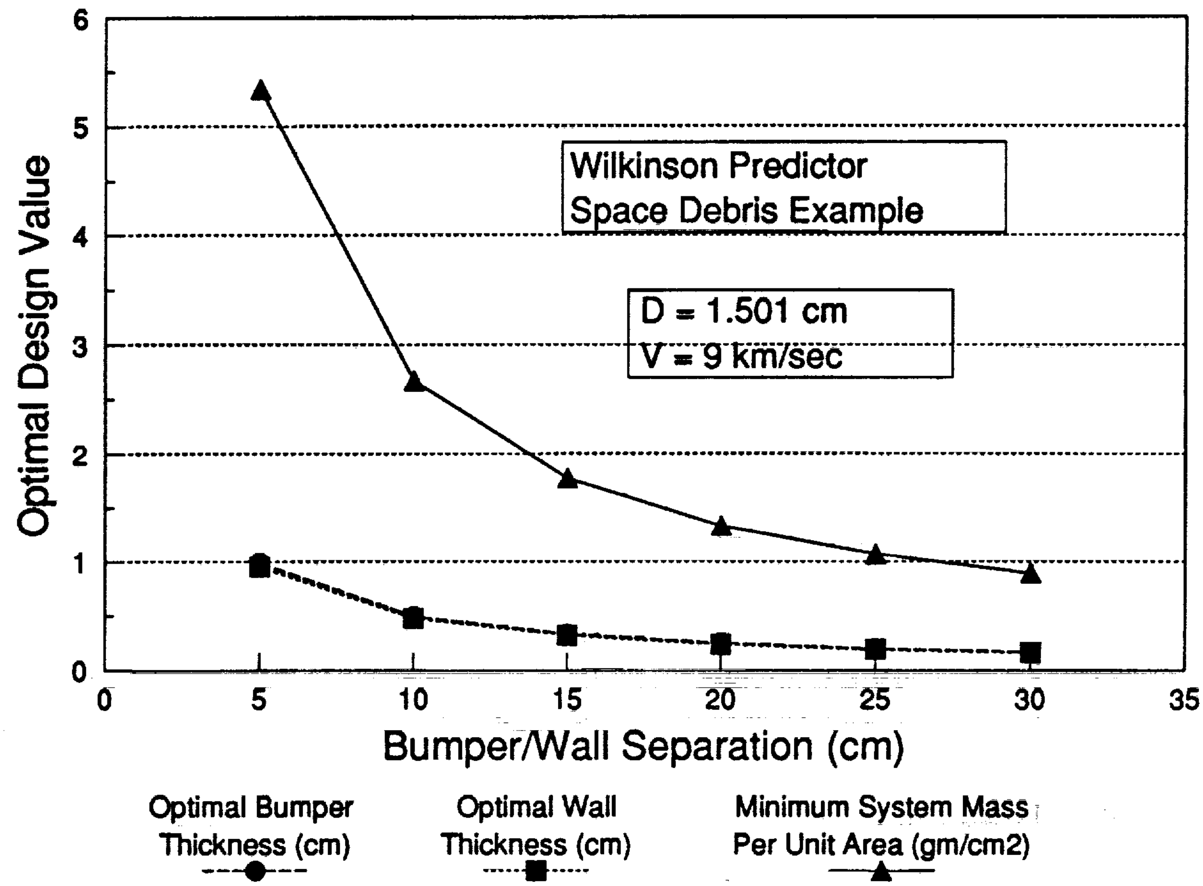



**Figure 2.3-2. Optimal Design Value vs Bumper/Wall Separation for Wilkinson Predictor**

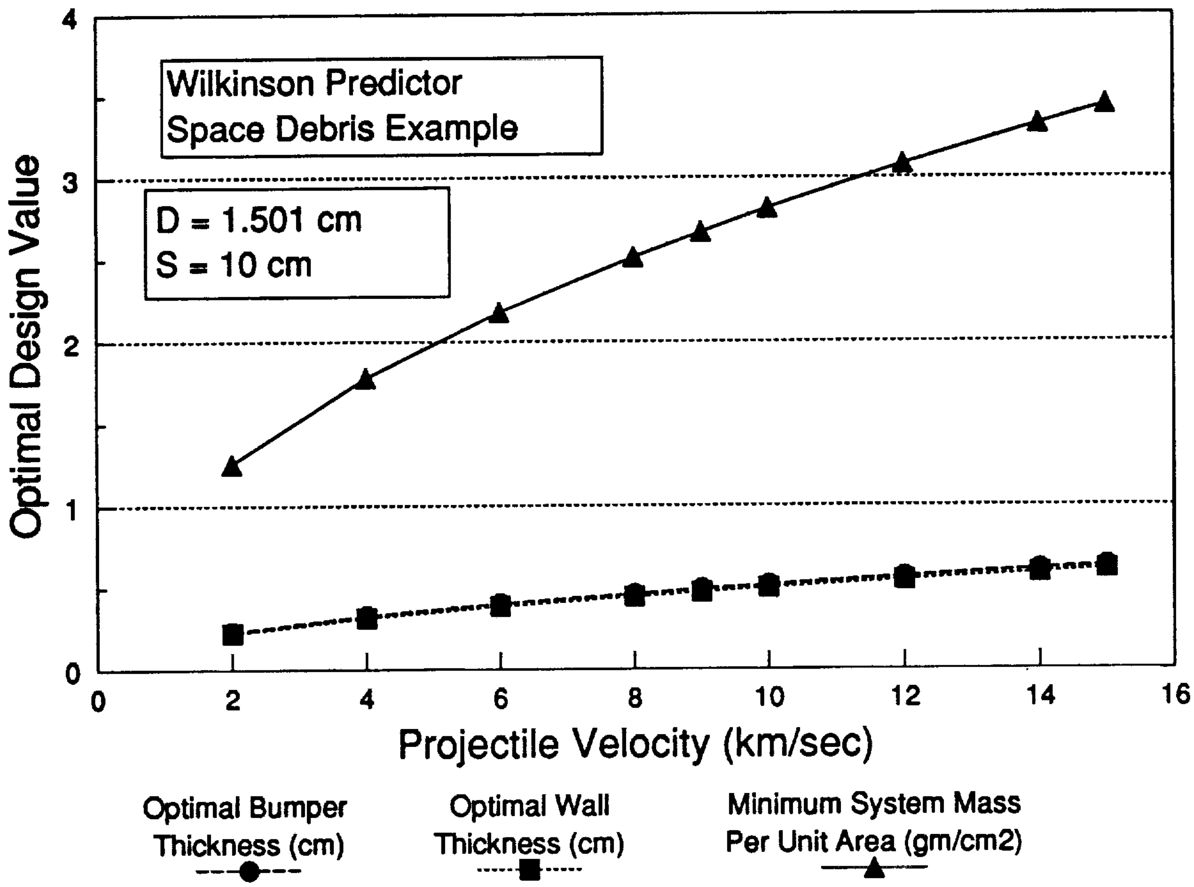



**Figure 2.3-3. Optimal Design Value vs Projectile Velocity for Wilkinson Predictor**

The normal impact predictor for the Burch model is given in functional form as

$$t_2 = \frac{\left(\frac{F_1 D}{N_1}\right)^{1.71}\left(\frac{C}{V}\right)^{2.29}}{S^{0.71}} \qquad [49]$$

where

$$F_1 = 2.42(t_1/D)^{-0.33} + 4.26(t_1/D)^{0.33} - 4.18 \qquad [50]$$

Equation [50] may be approximated by

$$\overline{K} = F_1^{1.71} = 2.8(t_1/D)^{0.57} + 1.58(t_1/D)^{-0.57} \quad [51]$$

Then W is given in posynomial form as

$$W = \rho_1 t_1 + \rho_2 \overline{CK} \quad [52]$$

where

$$\overline{C} = \frac{\left(\frac{D}{N_1}\right)^{1.71}\left(\frac{C}{V}\right)^{2.29}}{S^{0.71}} \quad [53]$$

The dual Geometric Programming problem is to maximize

$$\nu(\delta) = (\rho_1/\delta_1)^{\delta_1}\left(\frac{2.8\rho_2\overline{C}D^{-0.57}}{\delta_2}\right)^{\delta_2}\left(\frac{1.58\rho_2\overline{C}D^{0.57}}{\delta_3}\right)^{\delta_3} \quad [54]$$

subject to

$$\delta_1 + 0.57\delta_2 - 0.57\delta_3 = 0 \quad [55]$$

$$\sum_{i=1}^{3} \delta_i = 1 \quad [56]$$

Equations [55] and [56] may be partially solved to give

$$\delta_2 = 2.33(1 - 1.57\delta_3) \quad [57]$$

$$\delta_1 = 1.33(2\delta_3 - 1) \quad [58]$$

Since the dual variables must all be positive, we have

$$0.5 < \delta_3 < 0.64 \quad [59]$$

Thus, the one degree of difficulty algorithm is given by:

1. Vary $\delta_3$ from 0.5 to 0.64 to find the max $v(\delta)$.

2. Using the corresponding $\delta_3$, solve for $\delta_1$ and $\delta_2$.

3. $t_{1_0} = \delta_1 v_0(\delta)/\rho_1$.

4. $t_{2_0} = \dfrac{v_0(\delta) - \rho_1 t_{1_0}}{\rho_2}$

**Figures 2.3-4, 5, and 6** show the optimal design values of minimum system mass per unit area, and optimal bumper and wall thicknesses vs projectile diameter, bumper/wall separation, and projectile velocity, respectively, for the Burch predictor. **Figure 2.3-4** reflects a constant projectile density as given in equation [13]. In **Figure 2.3-6**, the impact angle remains constant at 0 degrees (normal).

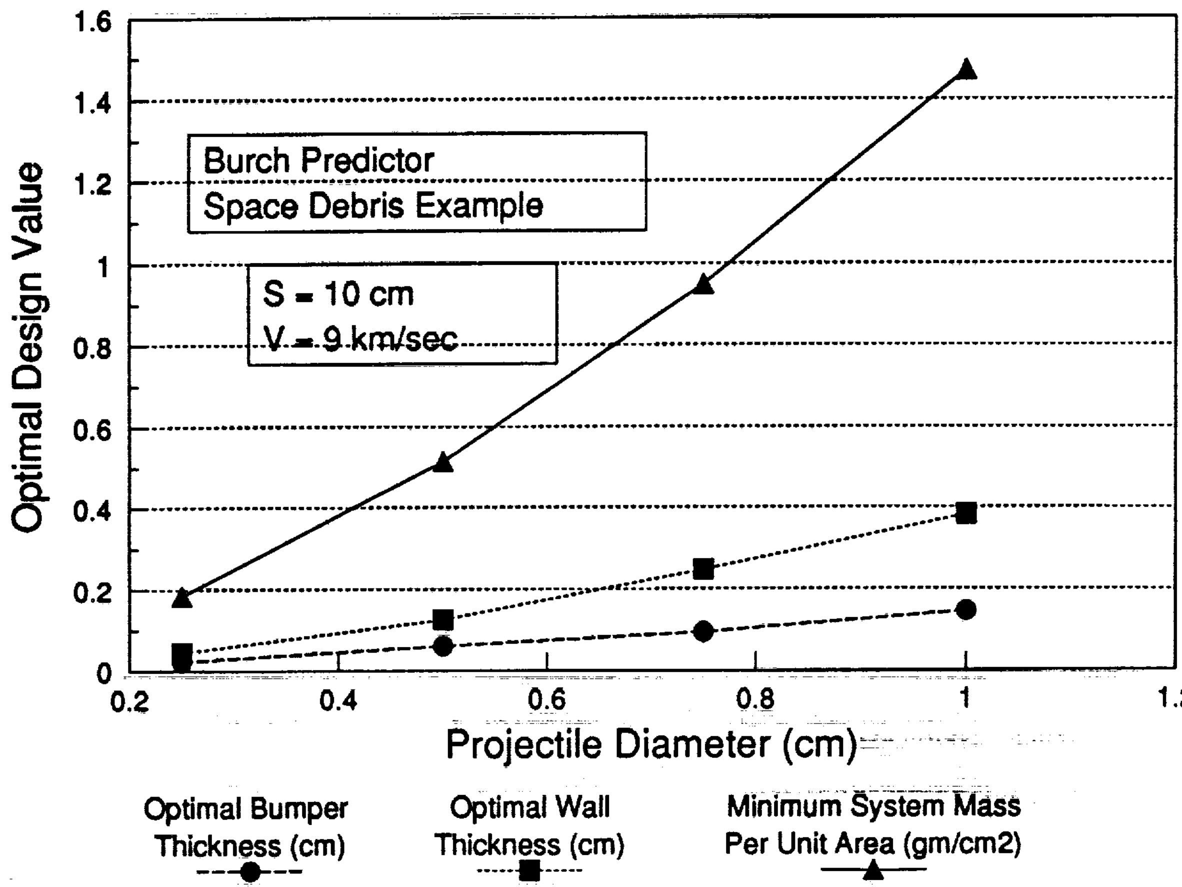



**Figure 2.3-4. Optimal Design Value vs Projectile Diameter for Burch Predictor**

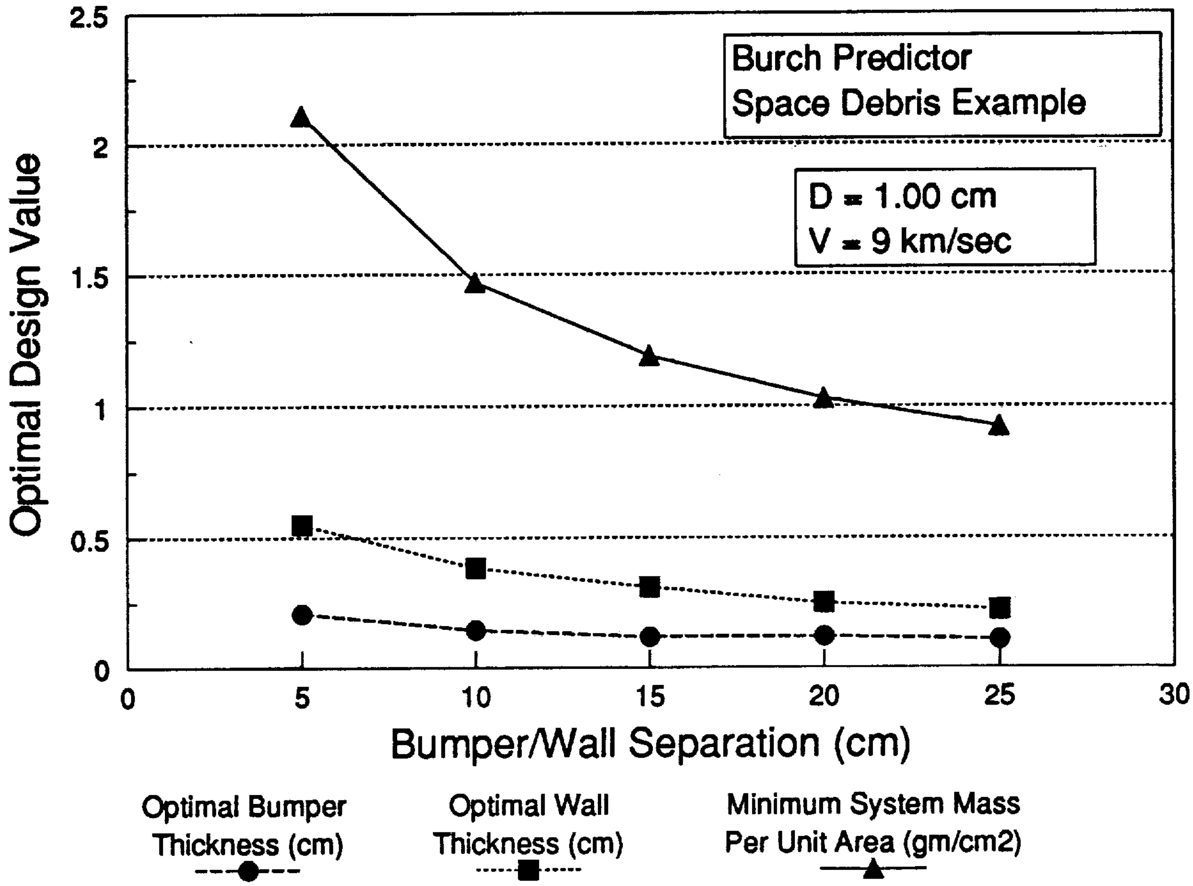



**Figure 2.3-5. Optimal Design Value vs Bumper/Wall Separation for Burch Predictor**

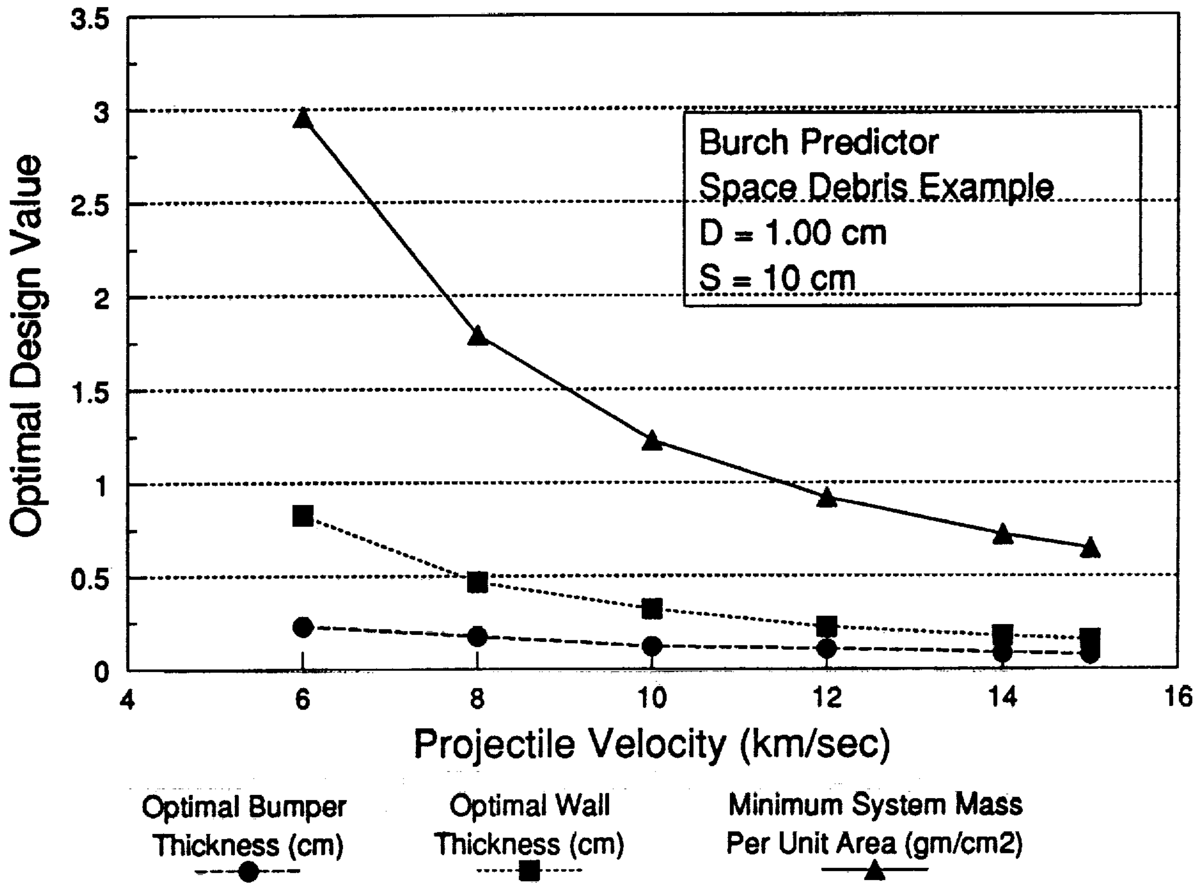



**Figure 2.3-6. Optimal Design Value vs Projectile Velocity for Burch Predictor**

The Nysmith equation was developed for meteoroid impacts and may be written

$$t_2 = \frac{5.08 V^{0.278} D^{2.92}}{t_1^{0.528} S^{1.39}} \qquad [60]$$

with inequality constraints

$$\frac{t_1}{D} \leq 0.5 \qquad [61]$$

and

$$\frac{t_2}{D} \leq 1.0 \qquad [62]$$

Substituting equation [60] into [38] results in

$$W = t_1 + \frac{5.08V^{0.278}D^{2.92}}{t_1^{0.528}S^{1.39}} \qquad [63]$$

The problem constraints may be rewritten

$$t_1 \leq \frac{D}{2} \qquad [64]$$

$$t_1 \geq \frac{21.72V^{0.527}D^{3.636}}{S^{2.633}} \qquad [65]$$

The first step in this analysis is to determine when the problem is feasible. This corresponds to the question: When is the constraint set defined by [64] and [65] nonempty? Clearly, this is the case if

$$\frac{D}{2} \geq \frac{21.72V^{0.527}D^{3.636}}{S^{2.633}} \qquad [66]$$

or

$$D \leq \frac{0.239S}{V^{0.2}} \qquad [67]$$

A more usable form is given by

$$S \geq 4.184DV^{0.2} \qquad [68]$$

The conditions of existence of a local (and thus global) optimal solution to the problem will now be established.

If

$$D \leq 0.23SV^{-0.2} \qquad [69]$$

then the optimal solution to the problem exists and is given by

$$t_{1_0} = \frac{1.907 V^{0.182} D^{1.91}}{S^{0.91}} \quad [70]$$

$$t_{2_0} = \frac{3.613 V^{0.182} D^{1.91}}{S^{0.91}} \quad [71]$$

$$W_0 = \frac{5.520 V^{0.182} D^{1.91}}{S^{0.91}} \quad [72]$$

Note that the ratio of optimal bumper thickness to total thickness is 0.345. The corresponding ratio for the wall is 0.655. Thus, provided the values of the systemic parameters satisfy [69], these ratios are constant.

Finally, notice that we provide optimality conditions for most of the feasibility region. In fact, it is now only necessary to determine the existence of optimal solutions in the interval

$$0.23 S V^{-0.2} \le D \le 0.24 S V^{-0.2} \quad [73]$$

**Figures 2.3-7, 8, and 9** show the optimal design values of minimum system mass per unit area, and optimal bumper and wall thicknesses vs projectile diameter, bumper/wall separation, and projectile velocity, respectively, for the Nysmith predictor. **Figure 2.3-7** reflects a constant meteoroid density. In **Figure 2.3-9**, the impact angle remains constant at 0 degrees (normal).

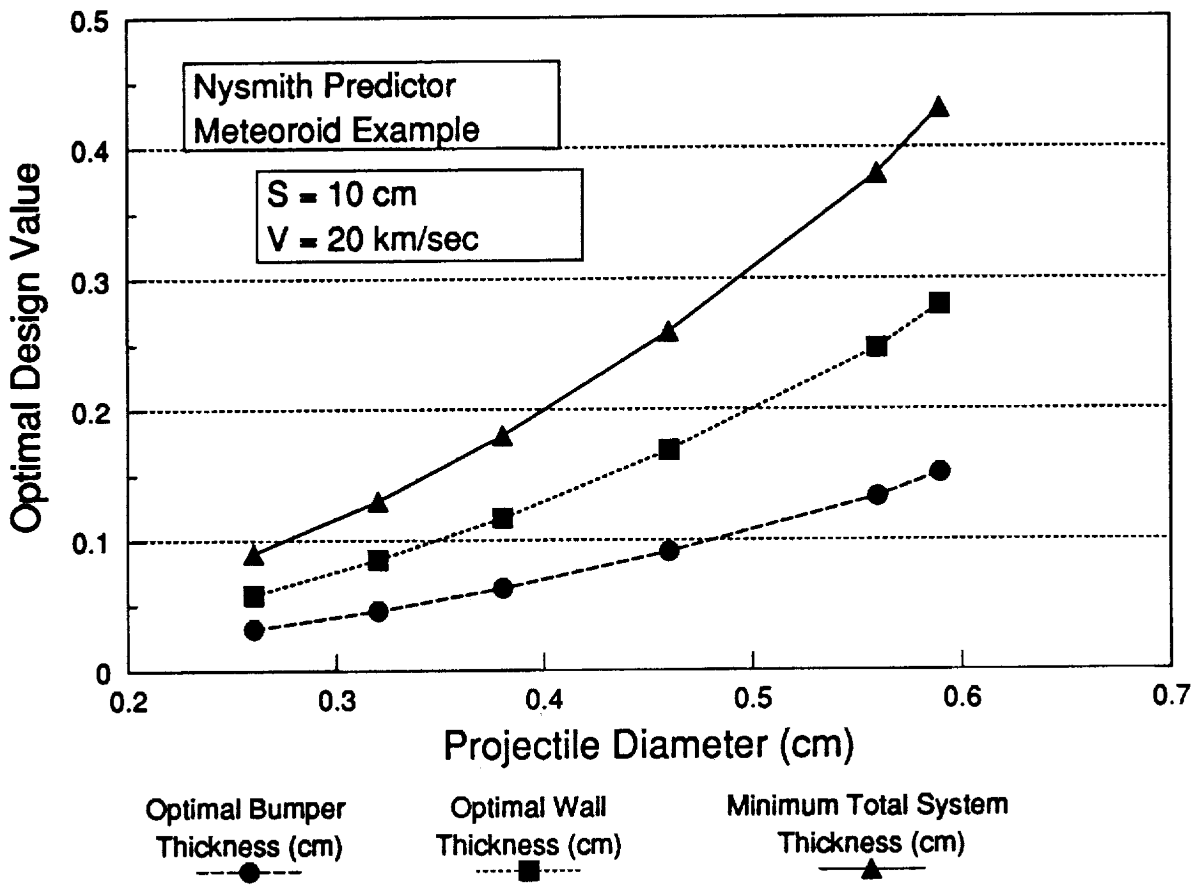



**Figure 2.3-7. Optimal Design Value vs Projectile Diameter for Nysmith Predictor**

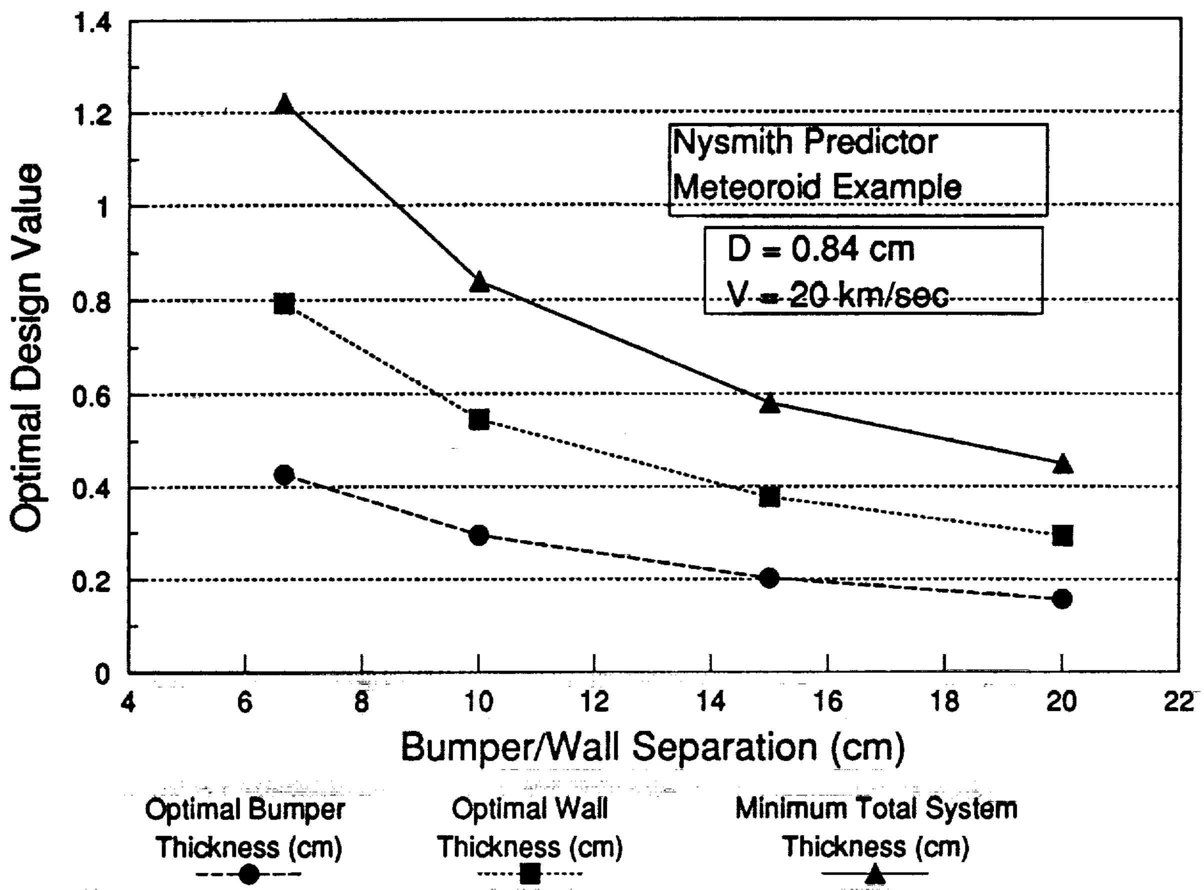



**Figure 2.3-8. Optimal Design Value vs Bumper/Wall Separation for Nysmith Predictor**

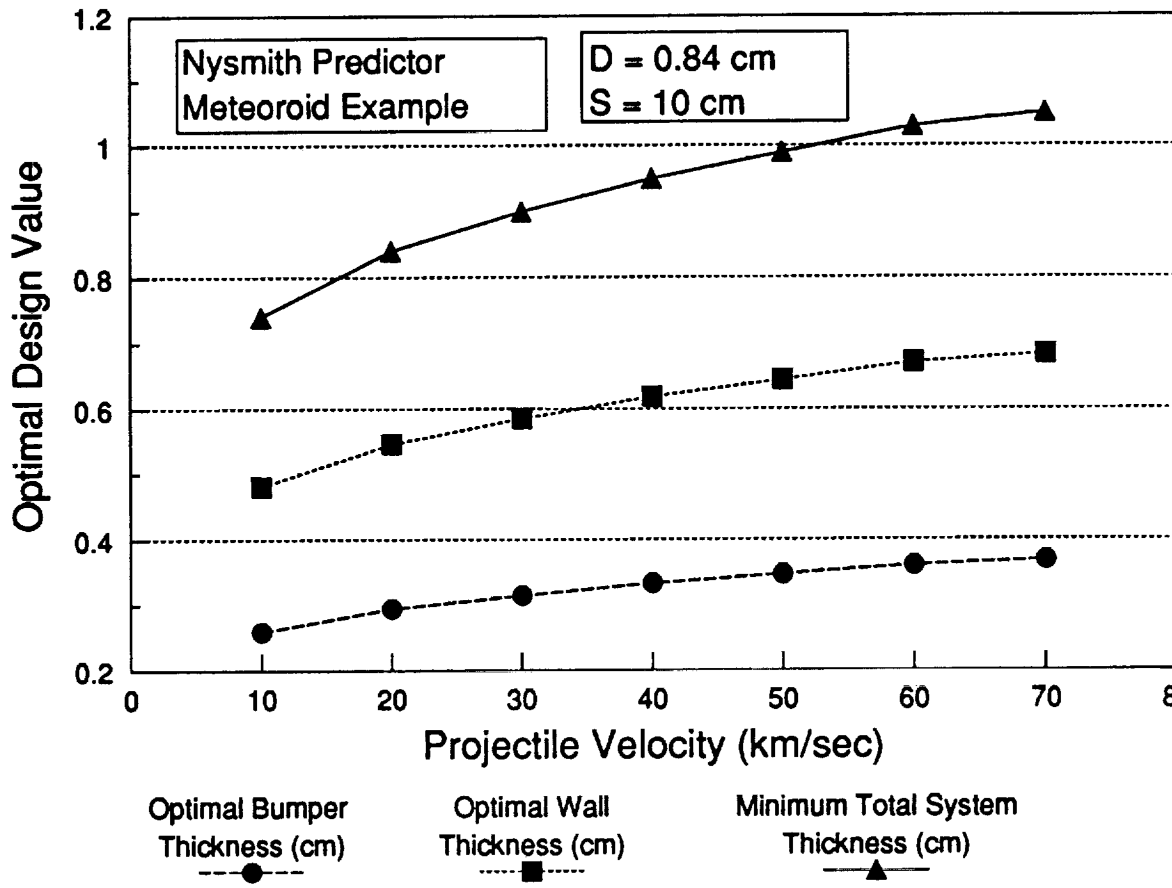



**Figure 2.3-9. Optimal Design Value vs Projectile Velocity for Nysmith Predictor**

We now consider the combination of impact predictors corresponding to ballistic, projectile shatter, and projectile melt/vaporization regions. The optimization problem is first formulated and then solved for these three impact regions. These optimal solutions are then integrated into an overall optimal solution. The predictor equations chosen are based on previous work performed by Boeing. The ballistic, projectile shatter, and projectile melt/vaporization predictors are given by the PEN4, Burch, and Wilkinson models respectively.

The PEN4 model in functional form is given by the following set of equations:

$$t_2 = 1.67\left(\frac{c_1\rho_p}{2S_{y_2}}\right)^{0.31}\left(\frac{0.281D\rho_p}{\rho_2}\right)^{1/3}\cos(\theta) \quad [74]$$

$$c_1 = \frac{a-b}{c+d} \quad [75]$$

$$a = 1.33V^2R_p^2\rho_p^2 \quad [76]$$

$$b = 8S_{y_1}t_1e^{-3.125(10^{-4})V}/\cos(\theta) \quad [77]$$

$$c = 1.33R_p^2\rho_p \quad [78]$$

$$d = R_pt_1\rho_1/\cos(\theta) \quad [79]$$

This set of equations is valid for

$$V \le V_f + 4000 \quad [80]$$

where

$$V_f = 4100 \text{ if } t_1/D \le 0.4 \quad [81]$$

$$V_f = 4986(t_1/D)^{0.21} \text{ if } t_1/D > 0.4 \quad [82]$$

When equations [74]-[79] are substituted into equation [38], a one-dimensional search is performed on $t_1$ with initial point

$$t_1 = 0.16625V^2R_p^2\rho_p\cos(\theta)\frac{e^{3.125(10^{-4})V}}{S_{y_1}} \quad [83]$$

corresponding to $t_2 = 0$. When a local optimal solution is determined, condition [80] is checked to determine if the ballistic region is appropriate for consideration.

The Burch model is actually two separate predictors, one for normal impacts, and one for oblique impacts. The normal impact predictor is given in functional form in equations [49] through [53]. The oblique Burch predictor is formulated in terms of flight path and normal path penetration as

$$t_2 = D\left(\frac{F_1 + 0.63F_2}{N_F}\right)^{12/7} (C/V)^{16/7} (D/S)^{5/7} \qquad [84]$$

where $F_1$ is as defined in [50] and

$$F_2 = 0.5 - 1.87(t_1/D) + (5t_1/D - 1.6)\chi^3 + (1.7 - 12t_1/D)\chi \qquad [85]$$

$$\chi = \tan(\theta) - 0.5 \qquad [86]$$

The weight minimization problem may then be formulated as

$$W = \rho_1 t_1 + \rho_2 t_2 \qquad [87]$$

subject to

$$N_N \le 0.85 \qquad [88]$$

where

$$N_N = F_3(D/t_2)(C/V)^{4/3} \qquad [89]$$

$$F_3 = 0.32(t_1/D)^{5/6} + 0.48(t_1/D)^{1/3}\sin^3(\theta) \qquad [90]$$

and $t_2$ is given by [84]. This problem is solved using an exterior penalty function technique with objective function

$$\phi(t_1) = W + \delta K(N_N - 0.85)^2 \qquad [91]$$

where

$$\delta = 1 \ \text{ if } N_N - 0.85 \ge 0 \qquad [92]$$

$$\delta = 0 \text{ if } N_N - 0.85 < 0 \tag{93}$$

A random search with a 99% confidence interval of 0.01 inches is performed, and K is increased until

$$\delta K (N_N - 0.85)^2 \leq \varepsilon \tag{94}$$

The random search interval for $t_1$ is specified by using the single plate equation

$$t_1 = K_1 m^{0.352} \rho_p^{1/6} V^{0.875} \tag{95}$$

$$K_1 = \frac{0.816}{e^{1/18} \rho_1^{1/2}} \tag{96}$$

The interval is then given by $[0,t_1]$.

Due to the discontinuities existing between the three impact predictors, an integrating algorithm must be developed. This algorithm is included for fixed velocities.

1. Compute optimal design for PEN4 predictor, $\left(t_{1_{0_P}}, t_{2_{0_P}}\right)$.
2. Check against PEN4 constraint [80].
3. If satisfied, the optimal design is $(t_{1_0}, t_{2_0}) = \left(t_{1_{0_P}}, t_{2_{0_P}}\right)$.
4. Otherwise, compute optimal designs for Burch and Wilkinson predictors, $\left(t_{1_{0_B}}, t_{2_{0_B}}\right)$ and $\left(t_{1_{0_W}}, t_{2_{0_W}}\right)$ respectively.
5. Compute Wilkinson wall induced by optimal Burch bumper, $t_{2_W}\left(t_{1_{0_B}}\right)$.
6. Compute Burch wall induced by optimal Wilkinson bumper, $t_{2_B}\left(t_{1_{0_W}}\right)$.
7. Find $(t_{1_0}, t_{2_0}) = \min_{\rho_1 t_1 + \rho_2 t_2}\left[\left\{t_{1_{0_B}}, \max\left(t_{2_{0_B}}, t_{2_W}\left(t_{1_{0_B}}\right)\right)\right\}, \left\{t_{1_{0_W}}, \max\left(t_{2_{0_W}}, t_{2_B}\left(t_{1_{0_W}}\right)\right)\right\}\right]$.

Once the optimal bumper and wall thicknesses are determined for each velocity, the integrated optimal bumper and wall thicknesses are found from

$$t_{i_0} = \int_0^{V_{max}} t_{i_0}(V, \theta(V)) f_n(V) dV \text{ for i=1,2.} \qquad [97]$$

<u>Real Time/Reactive Design</u>

Real time and reactive protective structures design refers to the concept of performing design in orbit through the use of smart structures, smart materials, or the combination of passive and active design techniques. The real time design approaches may be accomplished through particle sensing either before or during impact. Impact particle mass, velocity, angle, and location prediction is performed to provide the necessary algorithmic information to the structure/material controller. The material/structure is then configured to defeat the specific impact scenario anticipated. Real time/reactive protective structures design provides the most flexible and safest design alternative available, but also stresses technology the most.

## 2.4 Aluminum Alloy Bumper Materials

A comparison of aluminum alloy bumper materials is shown in **Table 2.4-1**. As shown, the minimum weight alloy is 2011-T8. Note the wide variation in CMC weights for different aluminum alloy bumper materials.

**Figure 2.4-1** shows the distribution of optimal bumper and wall thicknesses by hyper-velocity impact region for the 2011-T8 aluminum bumper material and 2219-T87 aluminum wall material. Note that the optimal bumper thickness is most heavily influenced by the projectile melt/vaporization region, while the optimal wall thickness is most heavily influenced by the projectile shatter region.

**Figure 2.4-2** shows the percentage area under the velocity probability distribution for the 2011-T8 aluminum bumper. Nearly 2/3 of the likelihood of impacts is above 8 km/sec, where testing is not generally attainable.

**Figure 2.4-3** shows the optimal 2011-T8 bumper thickness as a function of projectile diameter. This relationship is quite linear. Shown in **Figure 2.4-4** is the optimal 2219-T87 wall thickness as a function of projectile diameter. This relationship is slightly convex. **Figure 2.4-5** gives the minimum module weight (normalized to the baseline case) as a function of projectile diameter for the 2011-T8 bumper case.

**Table 2.4-1. Comparison of Aluminum Alloy Bumper Materials**

| ALUMINUM ALLOY BUMPER TYPE | OPTIMAL BUMPER THICKNESS (CM) | OPTIMAL WALL THICKNESS (CM) | MINIMUM CMC WEIGHT (KG) |
|---|---|---|---|
| 2219-T87 | 0.46 | 0.65 | 5715 |
| 1100-H18 | 0.50 | 0.64 | 5839 |
| 2011-T8 | 0.46 | 0.64 | 5665 |
| 2014-T6 | 0.44 | 0.71 | 5910 |
| 2024-T81 | 0.44 | 0.72 | 5929 |
| 5005-H18 | 0.49 | 0.64 | 5760 |
| 5050-H38 | 0.49 | 0.64 | 5768 |
| 5052-H38 | 0.49 | 0.65 | 5748 |
| 5056-H38 | 0.49 | 0.66 | 5762 |
| 5083-O | 0.53 | 0.65 | 5978 |
| 5086-0 | 0.55 | 0.65 | 6059 |
| 5154-H38 | 0.49 | 0.65 | 5769 |
| 5357-H38 | 0.48 | 0.64 | 5737 |
| 5456-O | 0.52 | 0.65 | 5942 |
| 6061-T6 | 0.48 | 0.64 | 5695 |
| 6063-T6 | 0.48 | 0.64 | 5737 |
| 6101-T6 | 0.49 | 0.64 | 5760 |
| 6151-T6 | 0.48 | 0.65 | 5719 |
| 7075-T6 | 0.43 | 0.71 | 5858 |

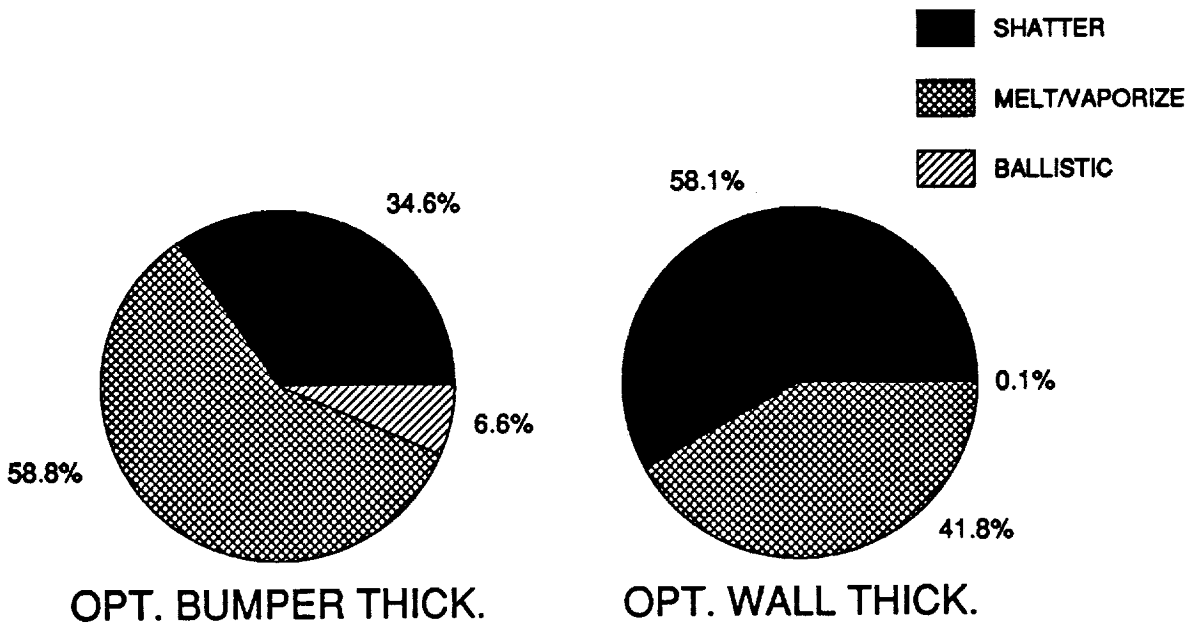



**Figure 2.4-1. Optimal Design Distribution By Impact Region (2011-T8 Bumper)**

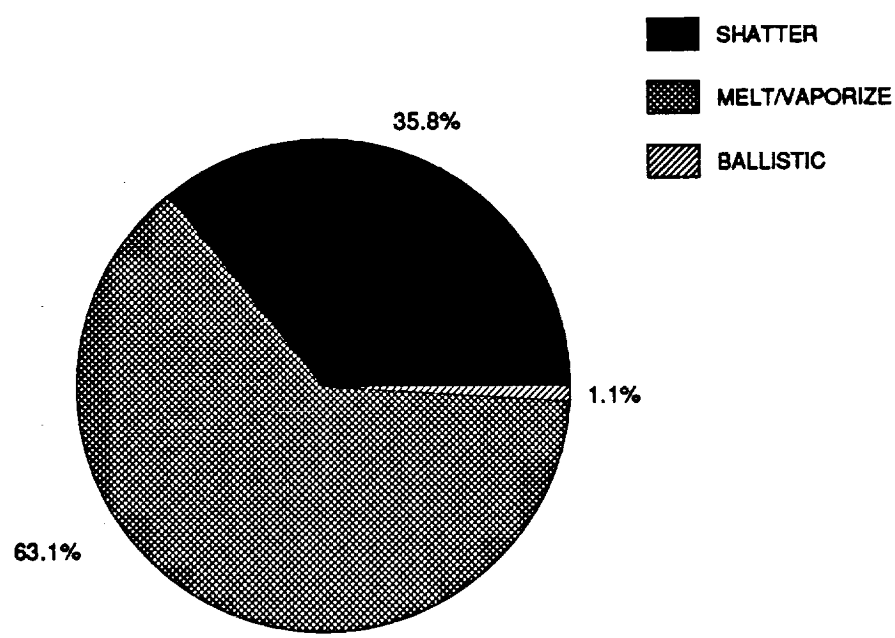



**Figure 2.4-2. Impact Velocity Distribution By Region (2011-T8 Bumper)**

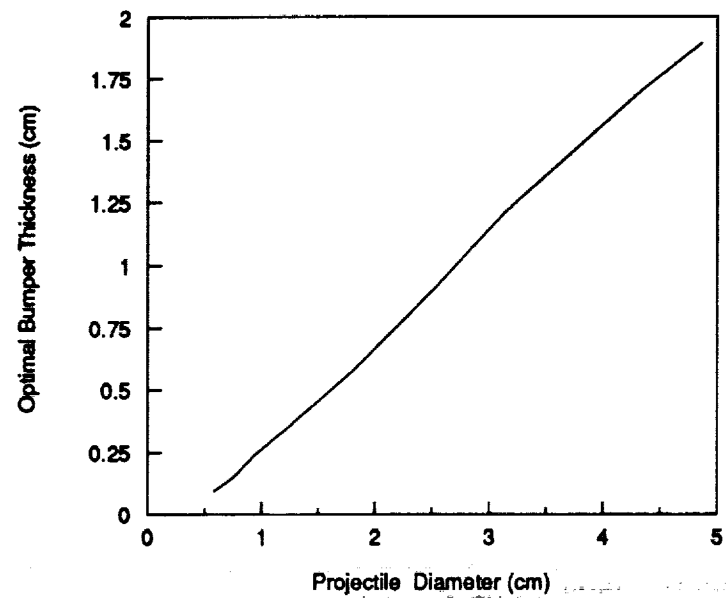



**Figure 2.4-3. Optimal Bumper Thickness vs Projectile Diameter (2011-T8 Bumper)**

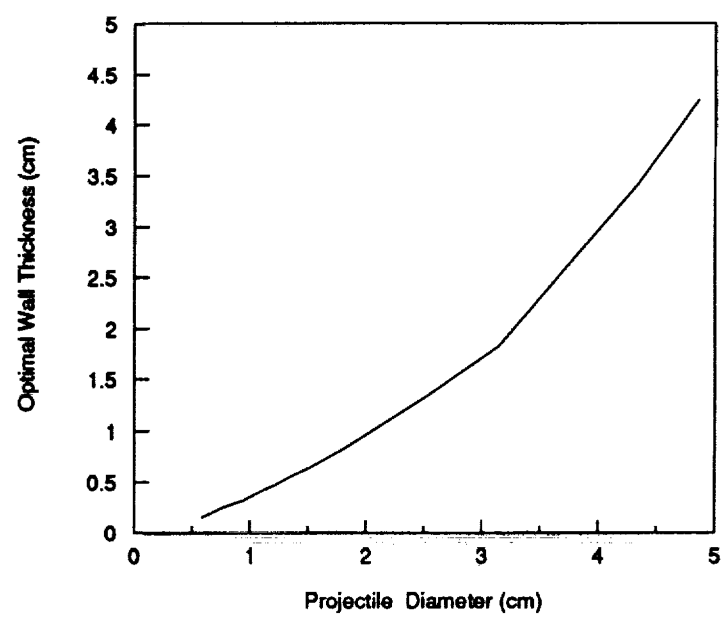



**Figure 2.4-4. Optimal Wall Thickness vs Projectile Diameter (2011-T8 Bumper)**

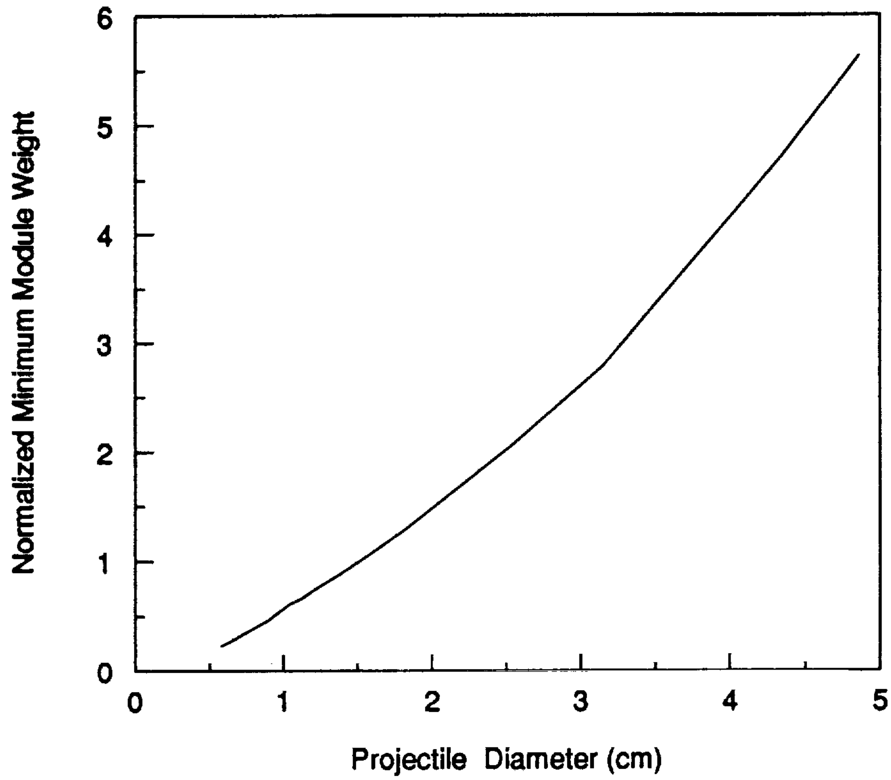



**Figure 2.4-5. Minimum Module Weight vs Projectile Diameter (2011-T8 Bumper)**

## 2.5 Bumper/Wall Separation

**Figure 2.5-1** shows the decreasing relationship between minimum CMC weight and bumper/wall separation. The CMC weight shown is normalized to the baseline minimum weight of 5665 kg given in **Table 2.4-1.** Note that increasing the bumper/wall separation from 10 to 15 cm results in a 25% decrease in CMC weight. The optimal bumper/wall separation of roughly 200-250 cm which minimizes the normalized minimum CMC weight is shown in **Figure 2.5-2.** Finally, the optimal bumper and wall thicknesses as functions of bumper/wall separation are given in **Figure 2.5-3.** This depicts a fairly constant optimal ratio between bumper and wall thickness.

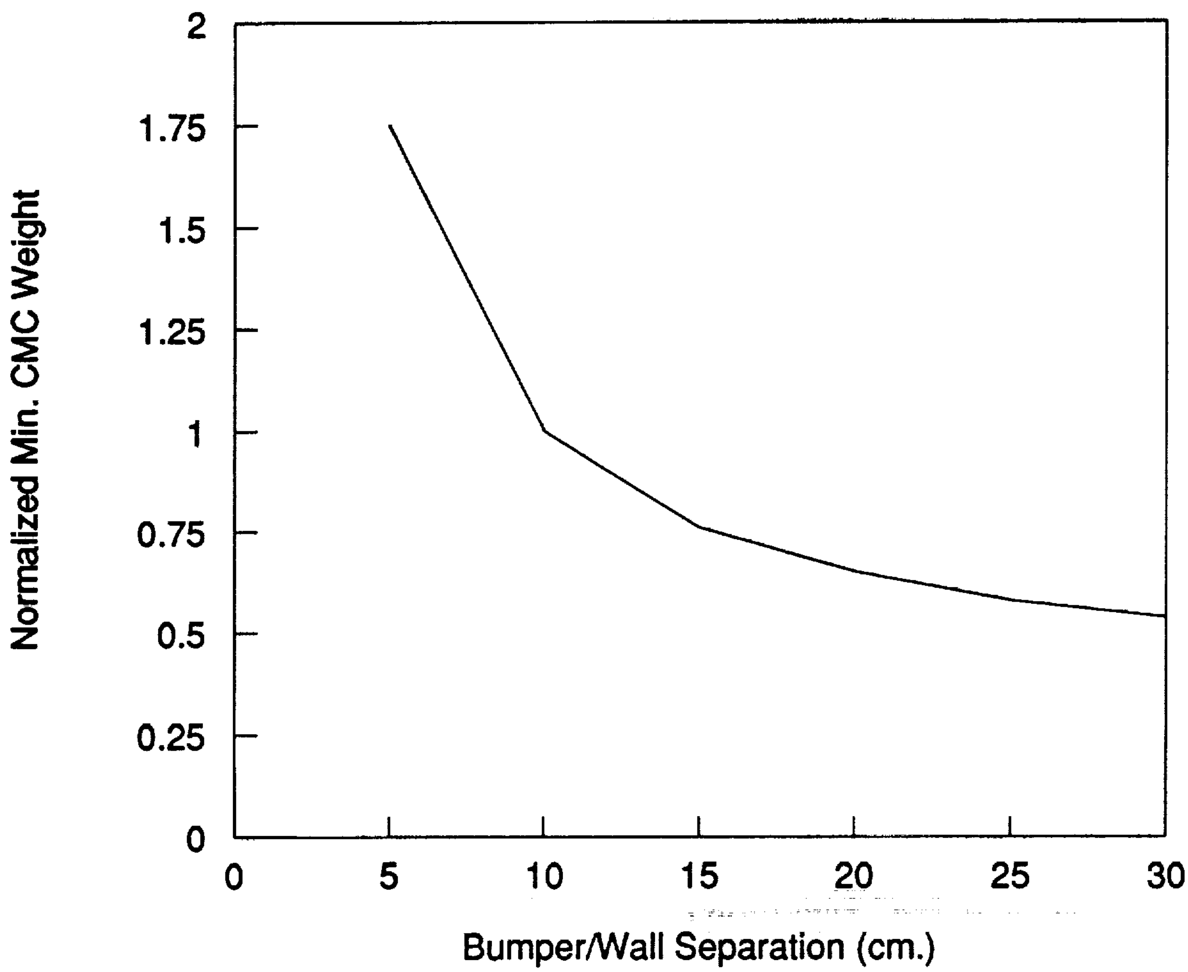



**Figure 2.5-1. Minimum CMC Weight vs Bumper/Wall Separation (Aluminum Alloys)**

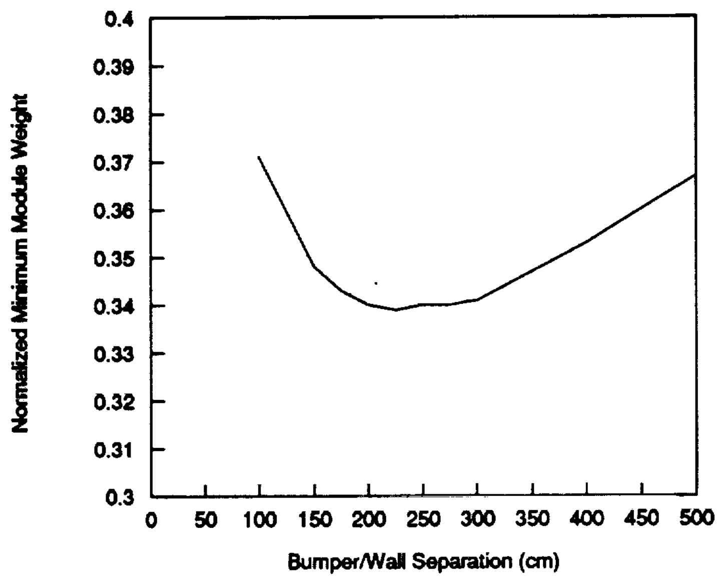



**Figure 2.5-2. Minimum Module Weight vs Bumper/Wall Separation (2011-T8 Bumper)**

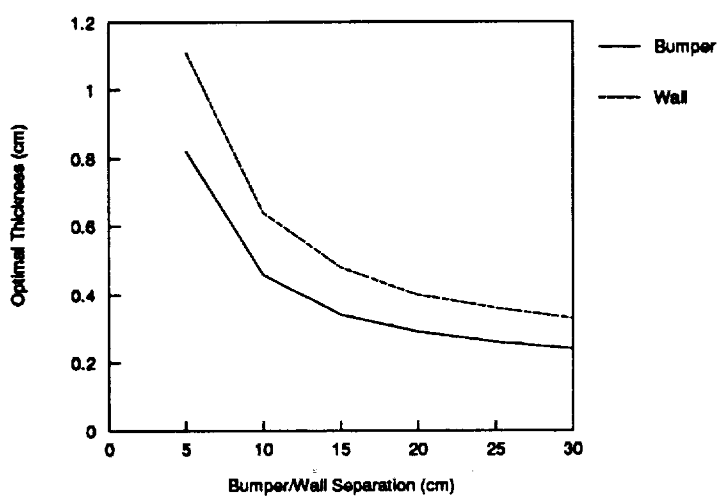



**Figure 2.5-3. Optimal CMC Thicknesses vs Bumper/Wall Separation (2011-T8 Bumper)**

## 2.6 Space Station Altitude

**Figures 2.6-1** and **2.6-2** show the relationships between Space Station altitude and projectile diameter and minimum CMC weight, respectively. Note the high sensitivity of design weight to altitude between 200 and 1000 km. The optimal bumper and wall thicknesses as functions of Space Station altitude are given in **Figure 2.6-3.**

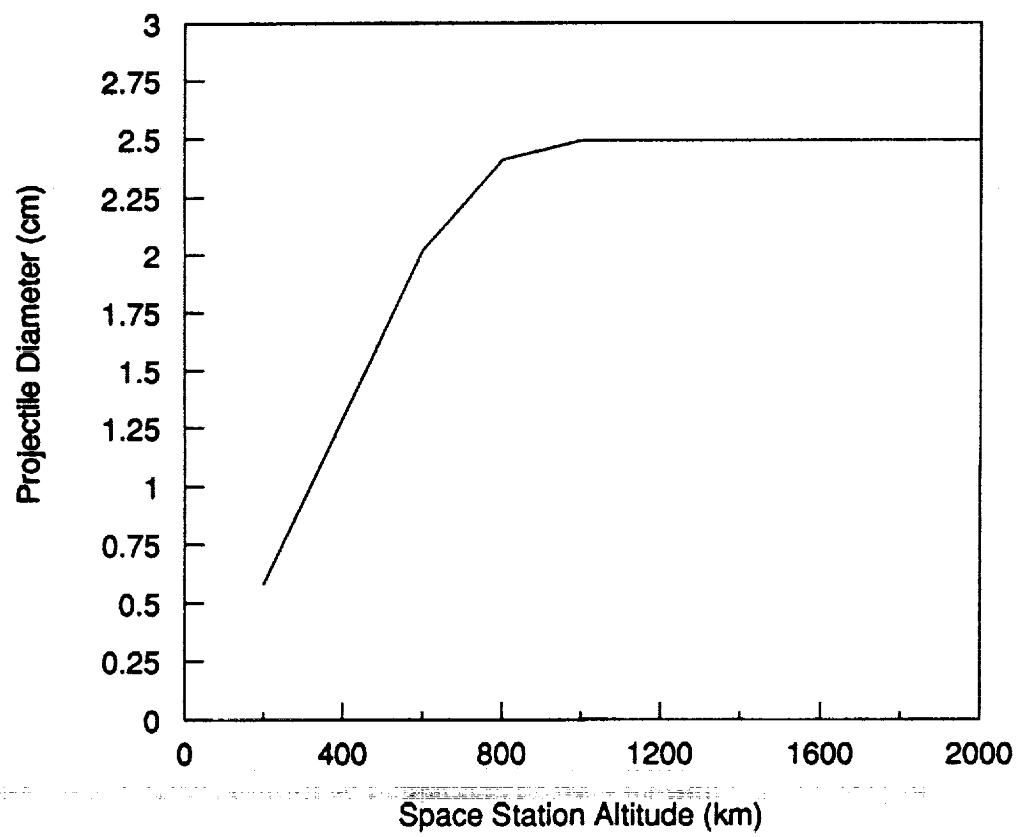



**Figure 2.6-1. Projectile Diameter vs Space Station Altitude**

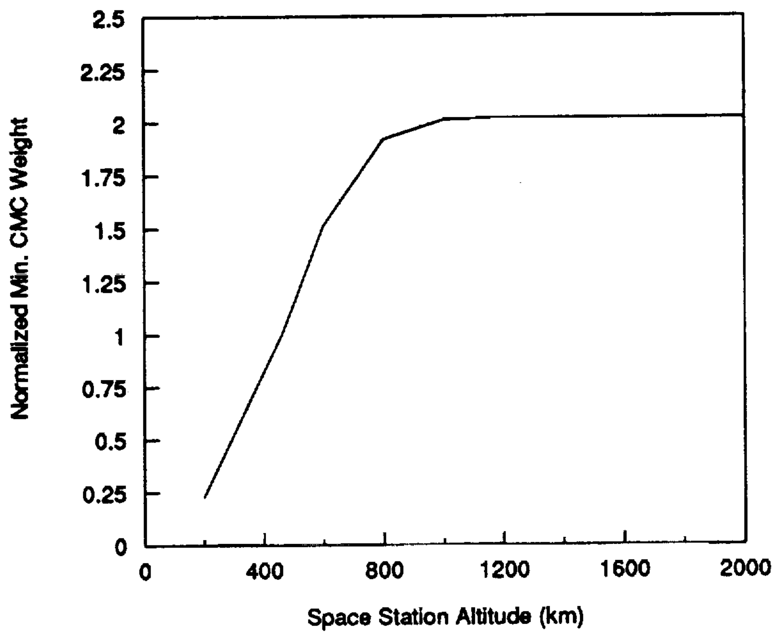



**Figure 2.6-2. Minimum CMC Weight vs Space Station Altitude (Aluminum Alloys)**

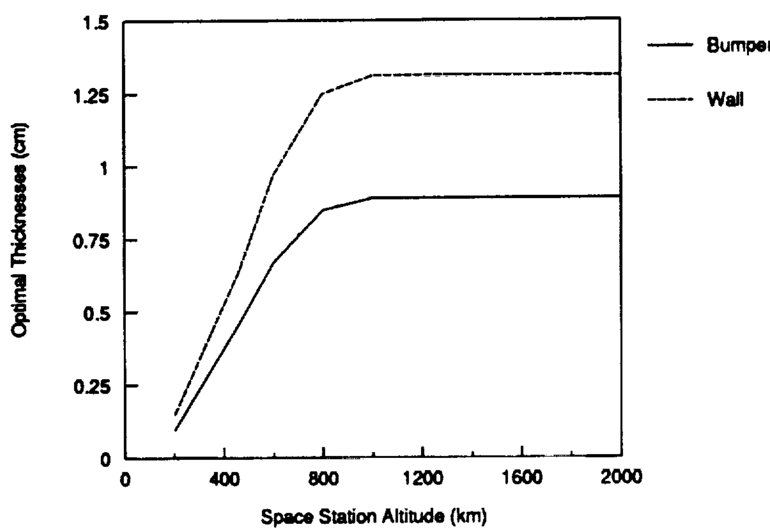



**Figure 2.6-3. Optimal CMC Thicknesses vs Space Station Altitude (2011-T8 Bumper)**

## 2.7 Risk Considerations

### Particle Velocity

**Figure 2.7-1** shows the normalized debris velocity probability distribution for the Space Station at 28.5 degrees inclination. Note the wide distribution of potential impact velocities from 0 to roughly 15 km/sec. Recall, also, the widely differing structural responses, and thus, optimal designs, over this velocity range. **Figure 2.7-2** shows the cumulative normalized velocity probability distribution for the Space Station.

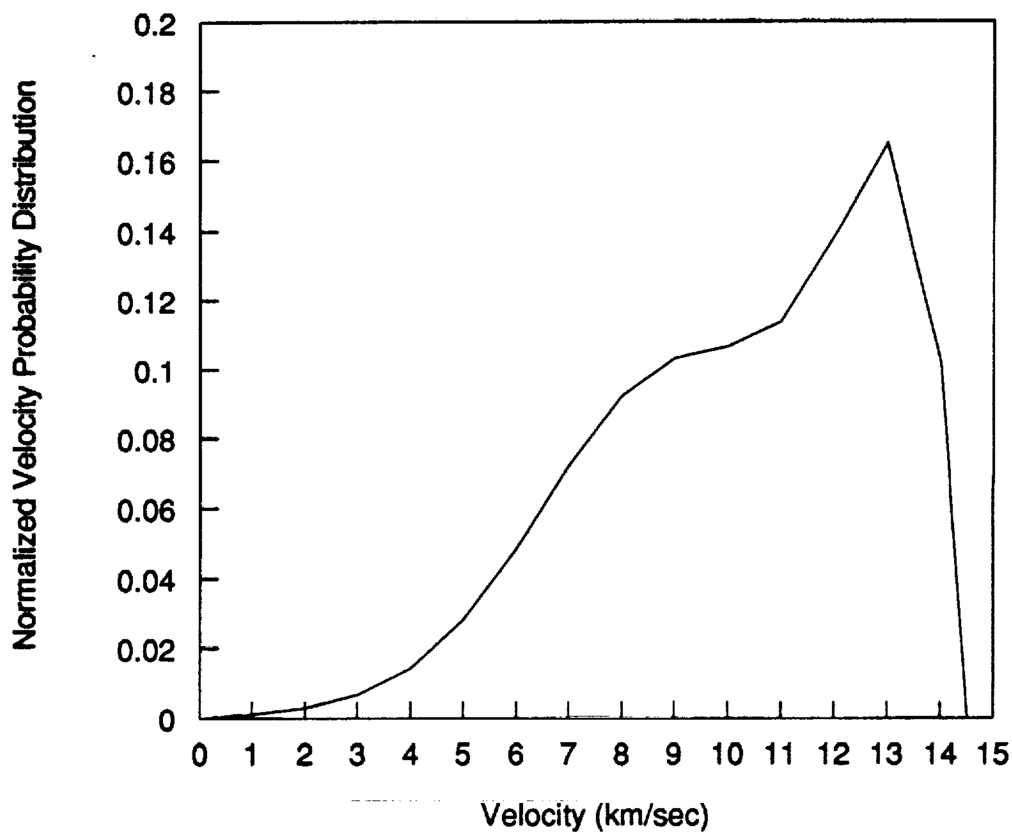



**Figure 2.7-1. Normalized Velocity Probability Distribution For 28.5 Degrees Inclination**

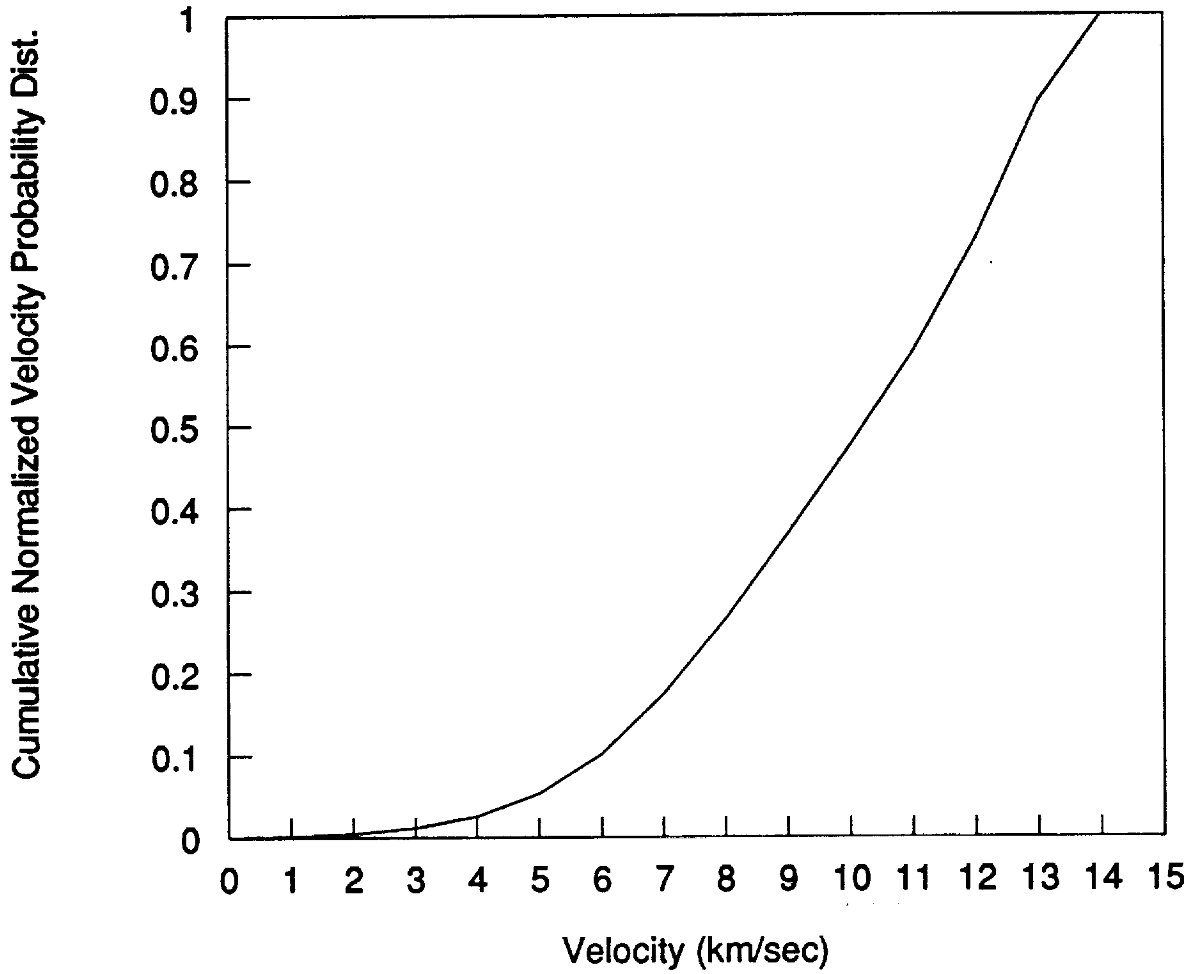



**Figure 2.7-2. Cumulative Normalized Velocity Probability Distribution For 28.5 Degrees Inclination**

Particle Impact Angle

The relationship between particle impact angle and velocity as prescribed by [17] is shown in **Figure 2.7-3.** Uncertainty bands are included as dashed lines. **Figure 2.7-4** shows the normalized angular probability distribution for the Space Station. Again, the optimal protective structures designs vary greatly over this range. **Figure 2.7-5** shows the cumulative normalized angular probability distribution for the Space Station.

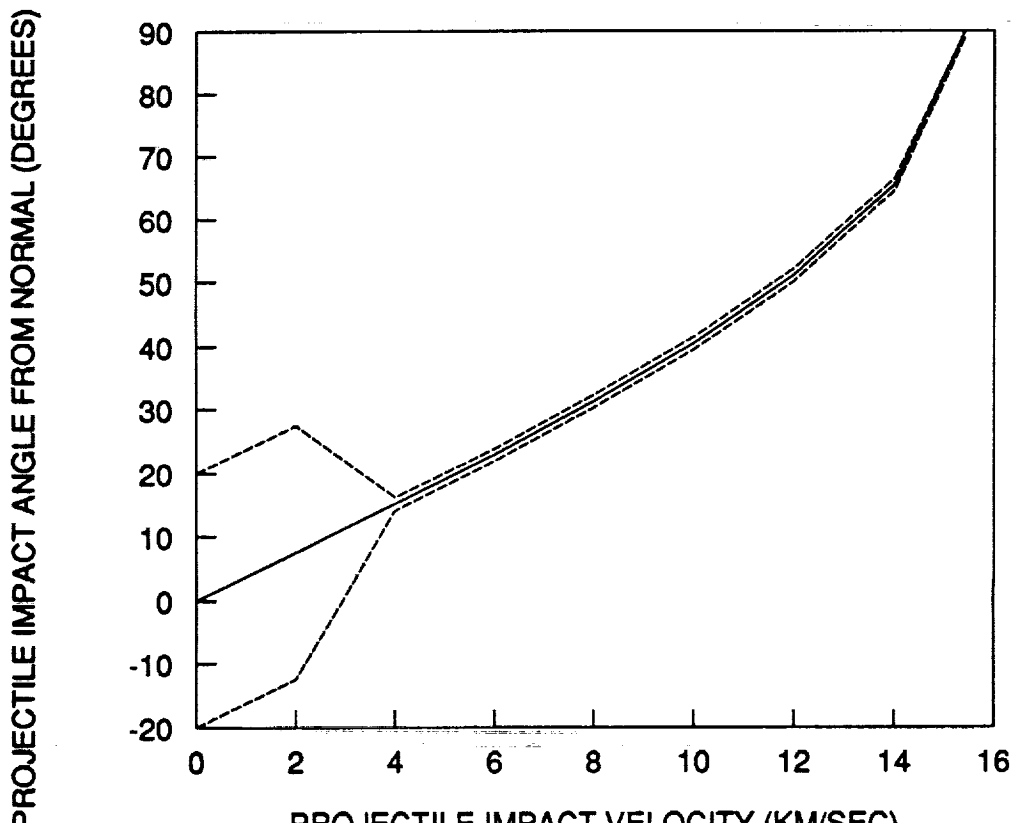



**Figure 2.7-3. Projectile Impact Angle From Normal of Surface Oriented Parallel to CMC Velocity Vector vs Impact Velocity**

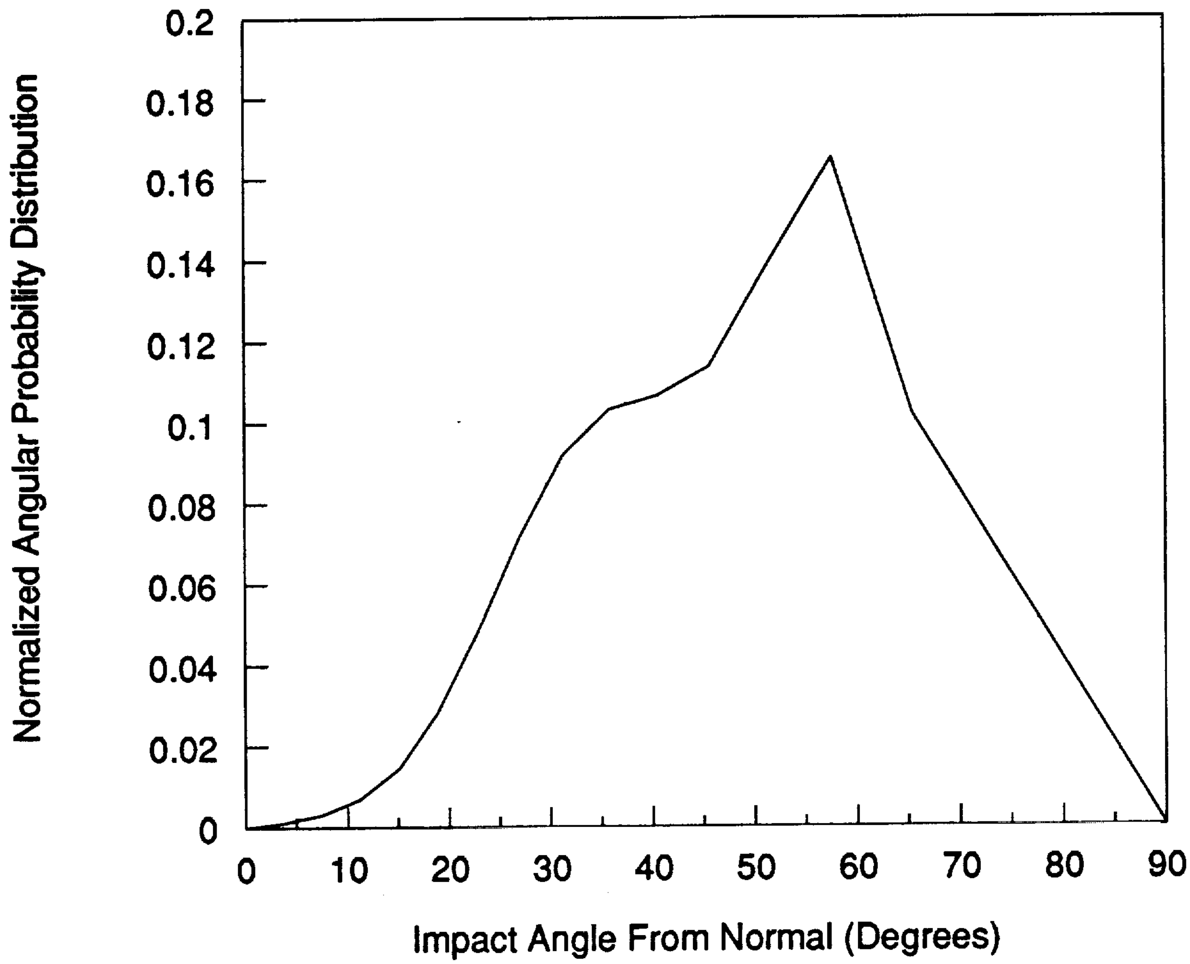



**Figure 2.7-4. Normalized Angular Probability Distribution For 28.5 Degrees Inclination For A Surface Oriented Parallel to CMC Velocity Vector**

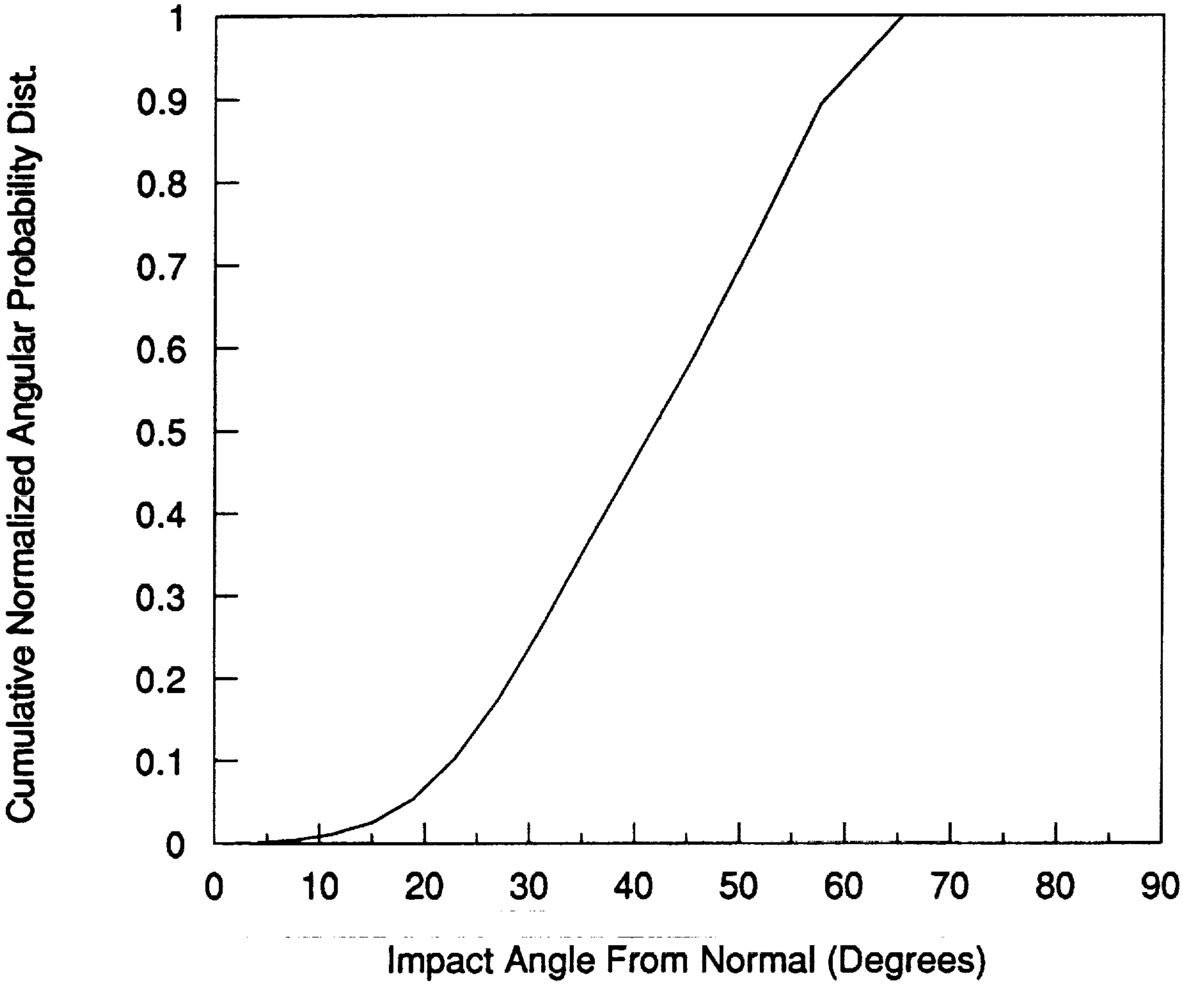



**Figure 2.7-5. Cumulative Normalized Angular Probability Distribution For 28.5 Degrees Inclination For A Surface Oriented Parallel to CMC Velocity Vector**

Particle Arrival Time

The particle arrival times are generally assumed to be Poisson distributed. Thus, the particle interarrival times are exponentially distributed. However, the mean times of arrivals change over time, and therefore, particle arrival times follow a nonstationary Poisson process. The obvious risk associated with the particle arrival times is not knowing when impacts will occur. Sensor data could reduce this risk.

Mission Risk

**Figures 2.7-6 and 2.7-7** show the relationships between total CMC mission risk and projectile diameter and minimum CMC weight, respectively. The weight shown is normalized to the baselined weight of 5665 kg. CMC mission risk is defined as one minus the total CMC probability of no penetration. Note that an increase form 0.03 to 0.05 in mission risk results in a 30% protective structures design weight reduction. The optimal bumper and wall thicknesses as functions of mission risk are given in **Figure 2.7-8**.

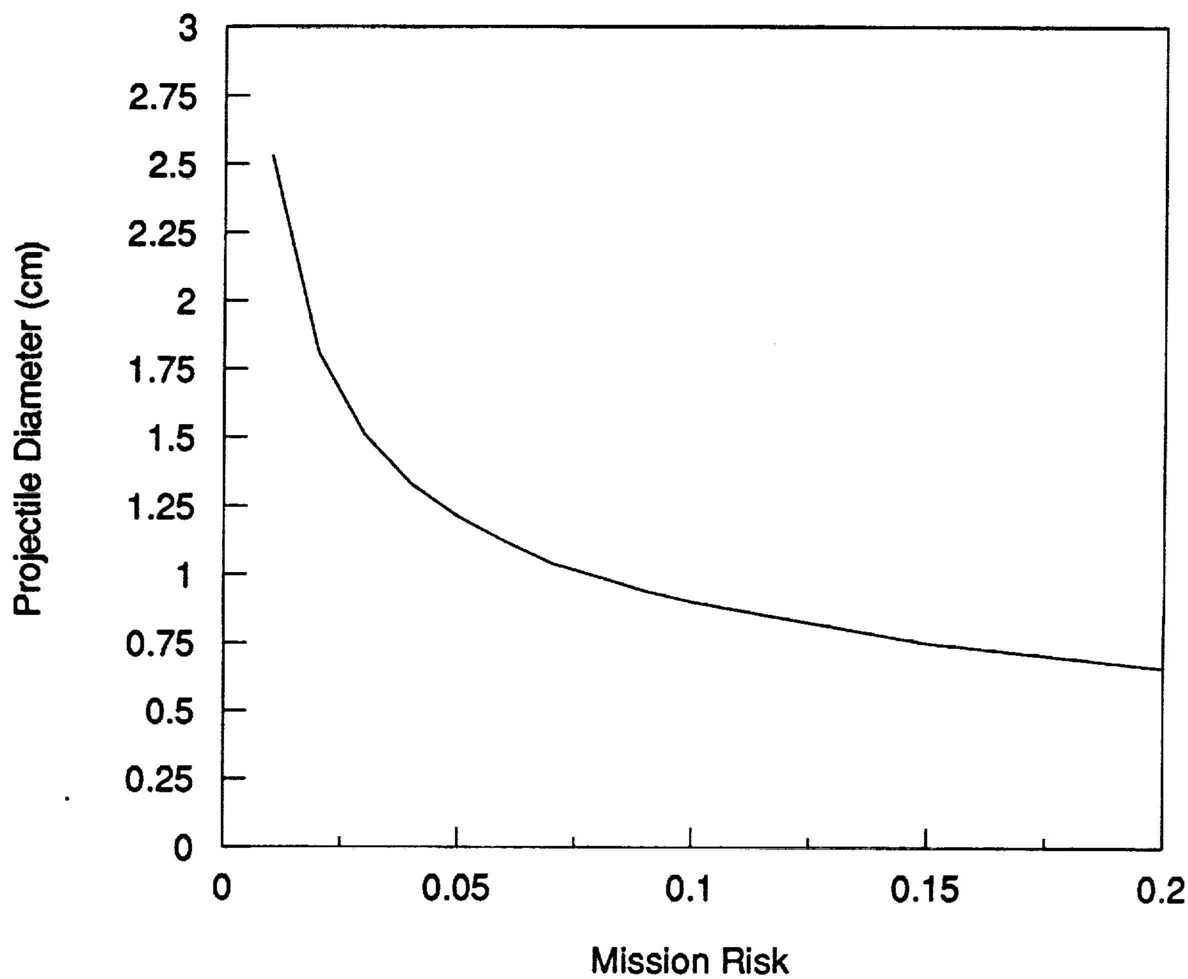



**Figure 2.7-6. Projectile Diameter vs Mission Risk**

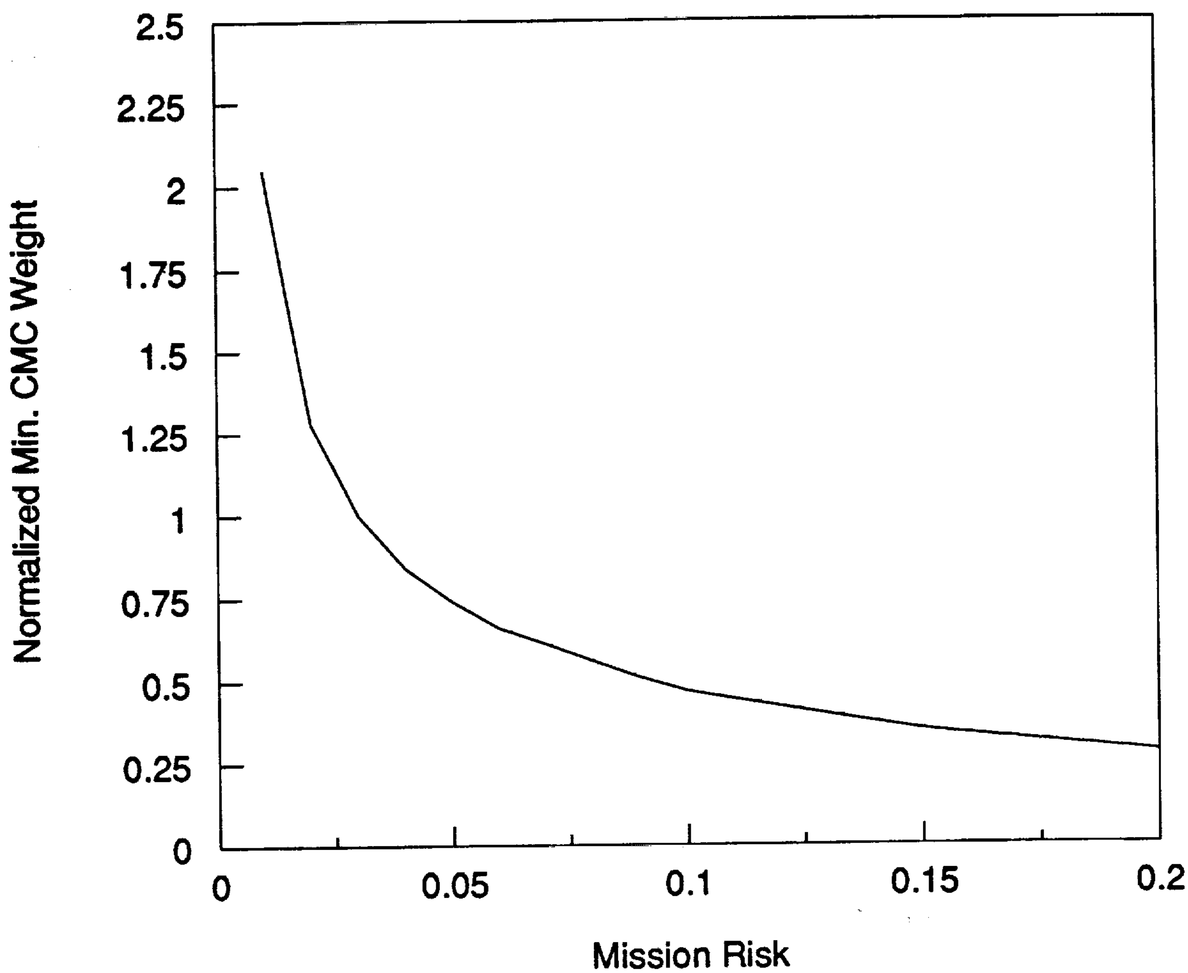



**Figure 2.7-7. Minimum CMC Weight vs Mission Risk (Aluminum Alloys)**

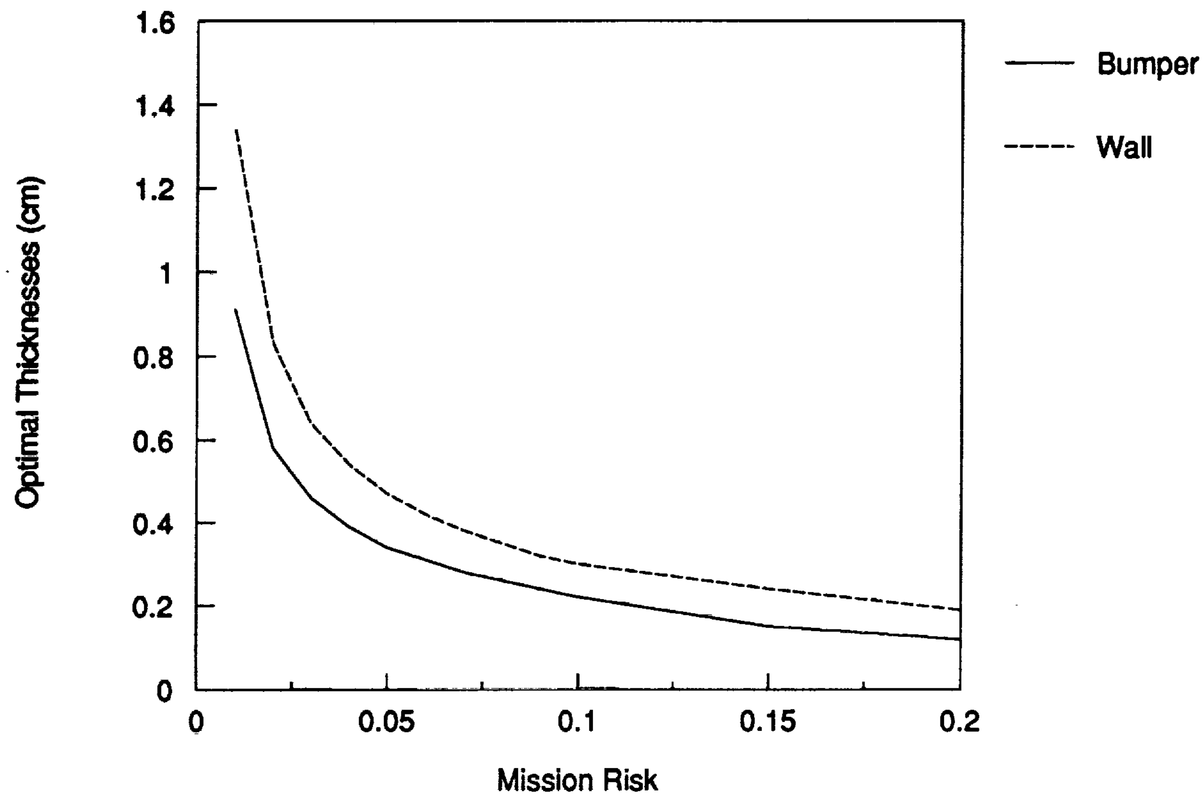



**Figure 2.7-8. Optimal CMC Thicknesses vs Mission Risk (2011-T8 Bumper)**

<u>Mission Duration</u>

**Figures 2.7-9 and 2.7-10** show the relationships between Space Station beginning year of operation and projectile diameter and minimum CMC weight, respectively. Note the convex shape between 1995 and 2000 followed by a concave representation through 2005. This is due to a benign solar flux effect in the latter years. A schedule delay of 5 years results in a 50% increase in protective structures design weight. The optimal bumper and wall thicknesses as functions of first year of operation are given in **Figure 2.7-11.**

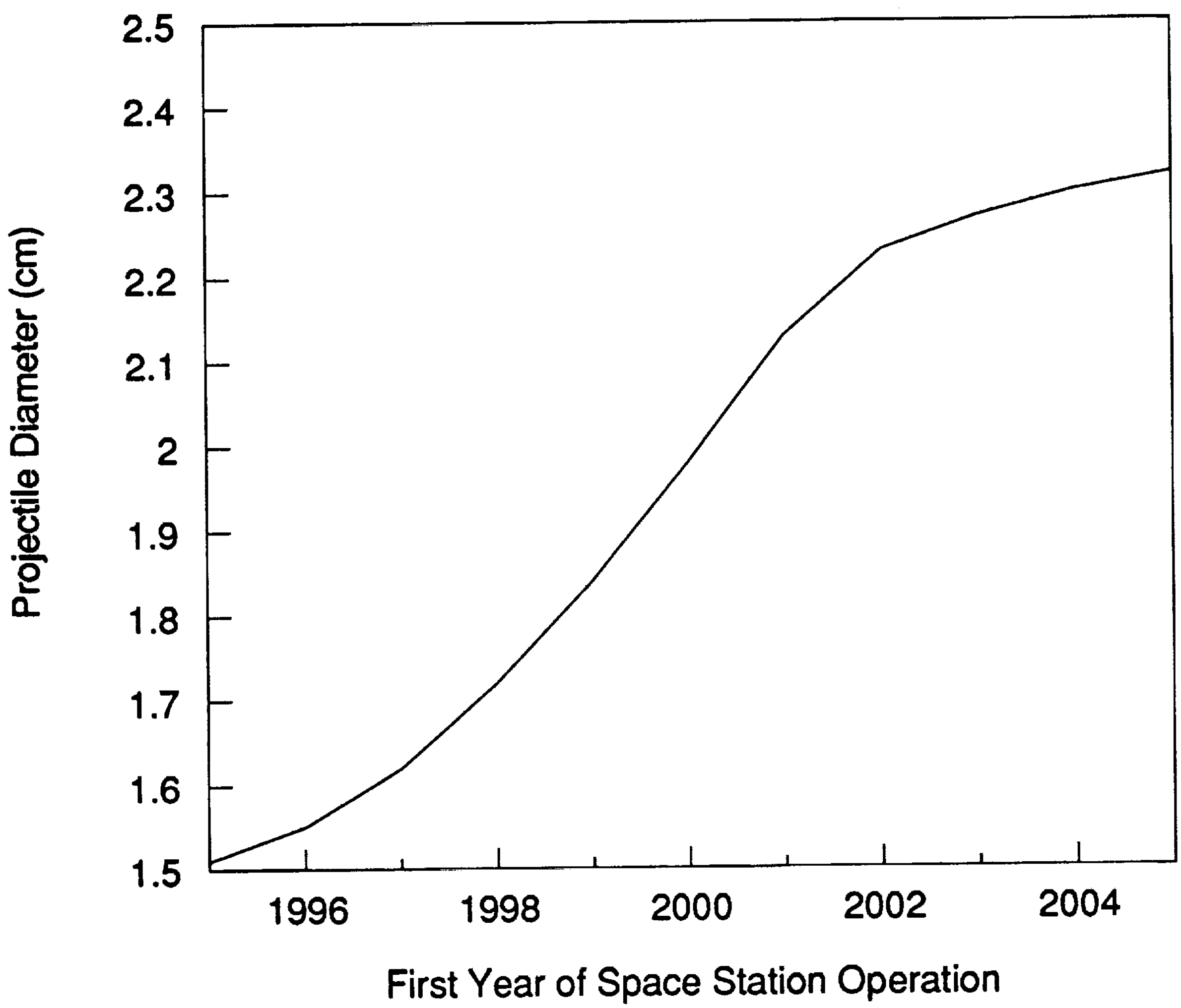



**Figure 2.7-9. Projectile Diameter vs First Year of Space Station Operation**

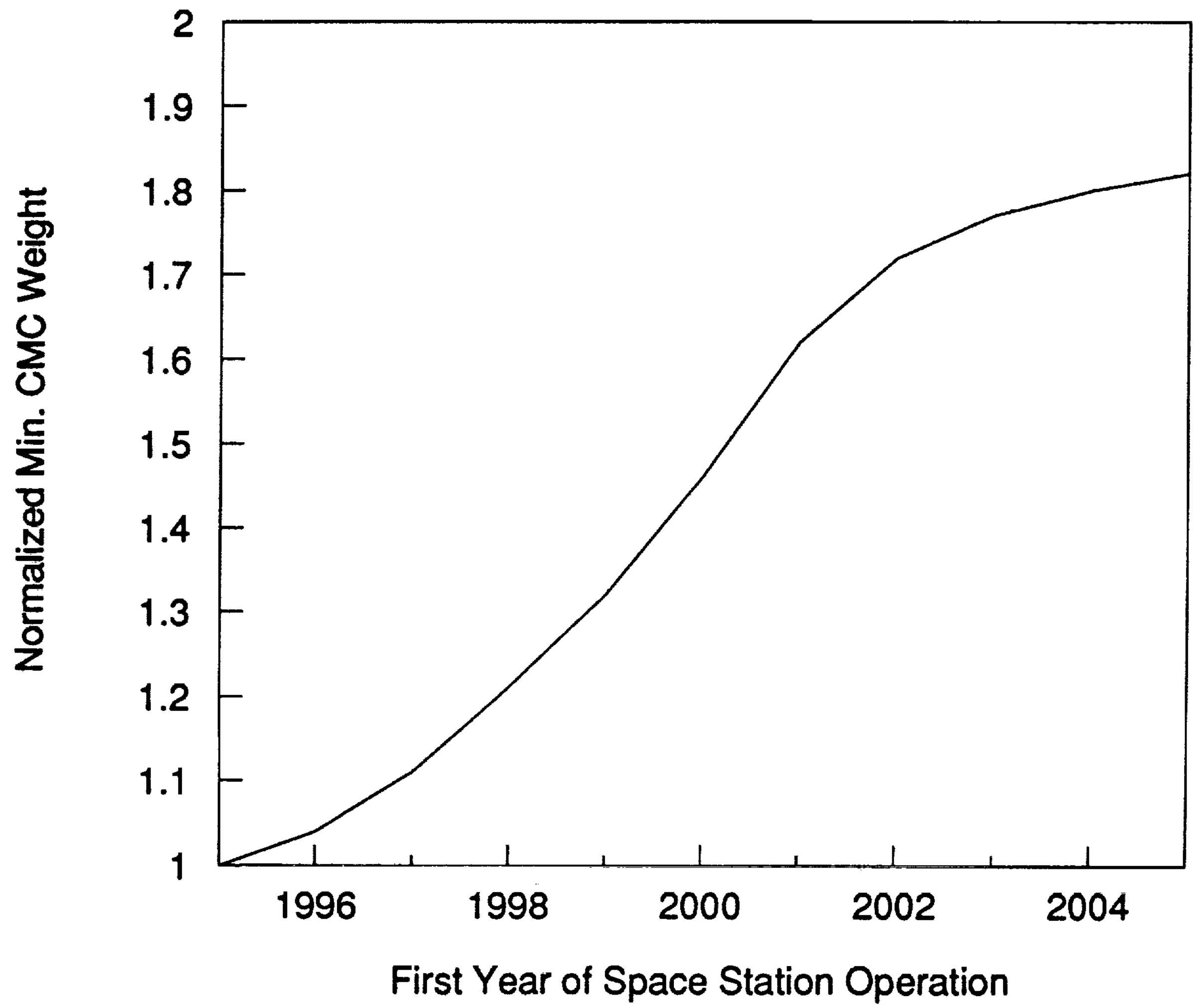



**Figure 2.7-10. Minimum CMC Weight vs First Year of Operation (Aluminum Alloys)**

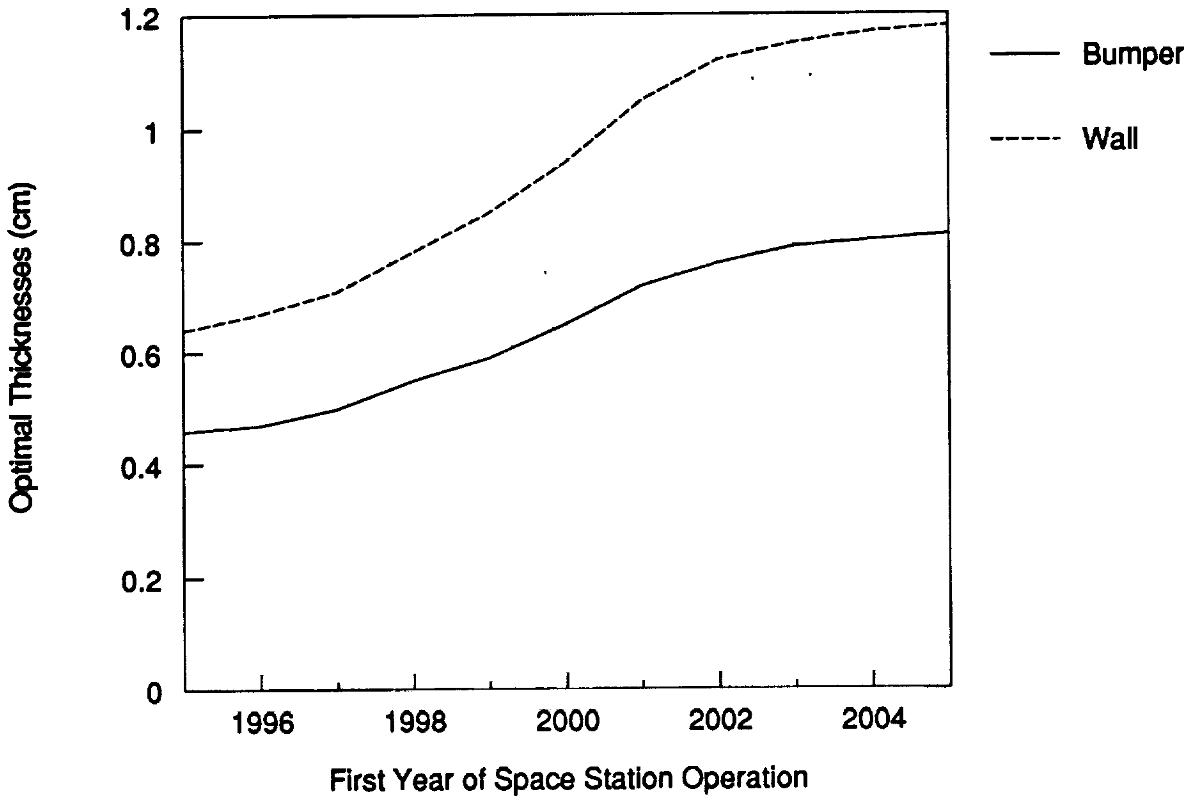



**Figure 2.7-11. Optimal CMC Thicknesses vs First Year of Operation (2011-T8 Bumper)**

**Figures 2.7-12 and 2.7-13** show the relationships between Space Station mission duration and projectile diameter and minimum CMC weight, respectively. These trades are for constant beginning years of operation of 1995. Note the shape reversal occurring at about 15 years. This is due to a solar flux effect for that particular period. A 10 year increase in mission duration more than doubles protective structures design weight. The optimal bumper and wall thicknesses as functions of Space Station mission duration are given in **Figure 2.7-14**.

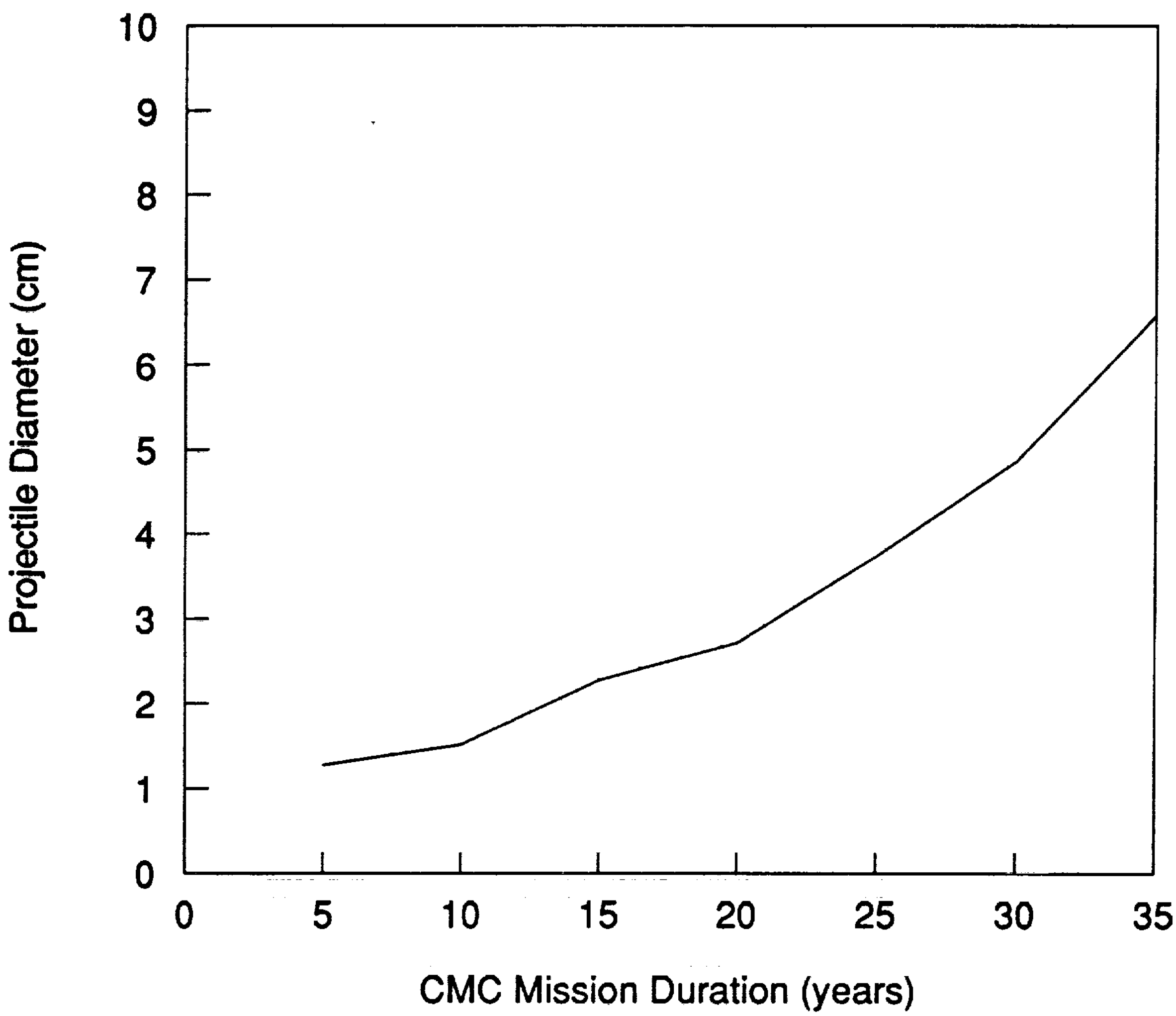



**Figure 2.7-12. Projectile Diameter vs Space Station Mission Duration**

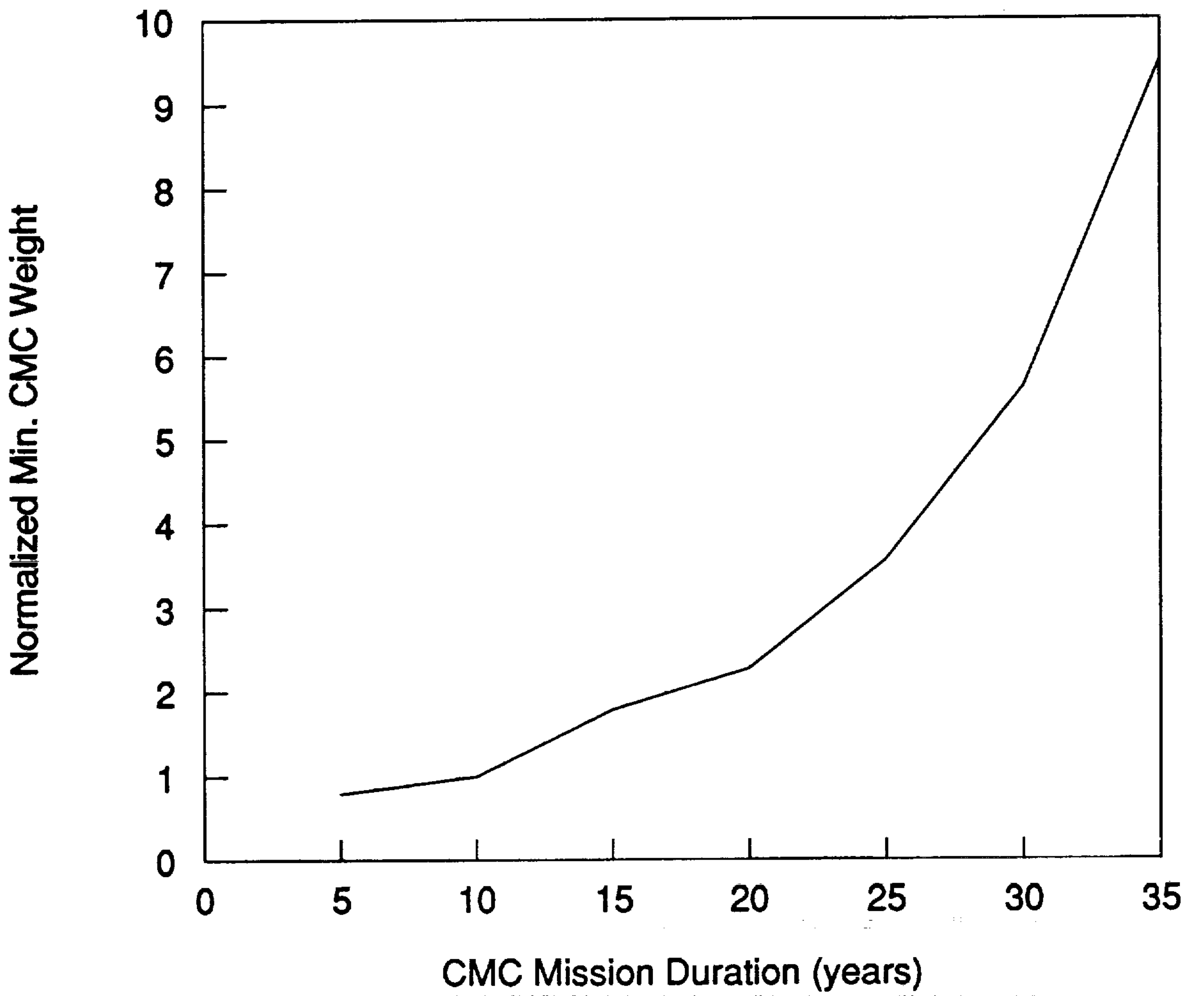



**Figure 2.7-13. Minimum CMC Weight vs Mission Duration (Aluminum Alloys)**

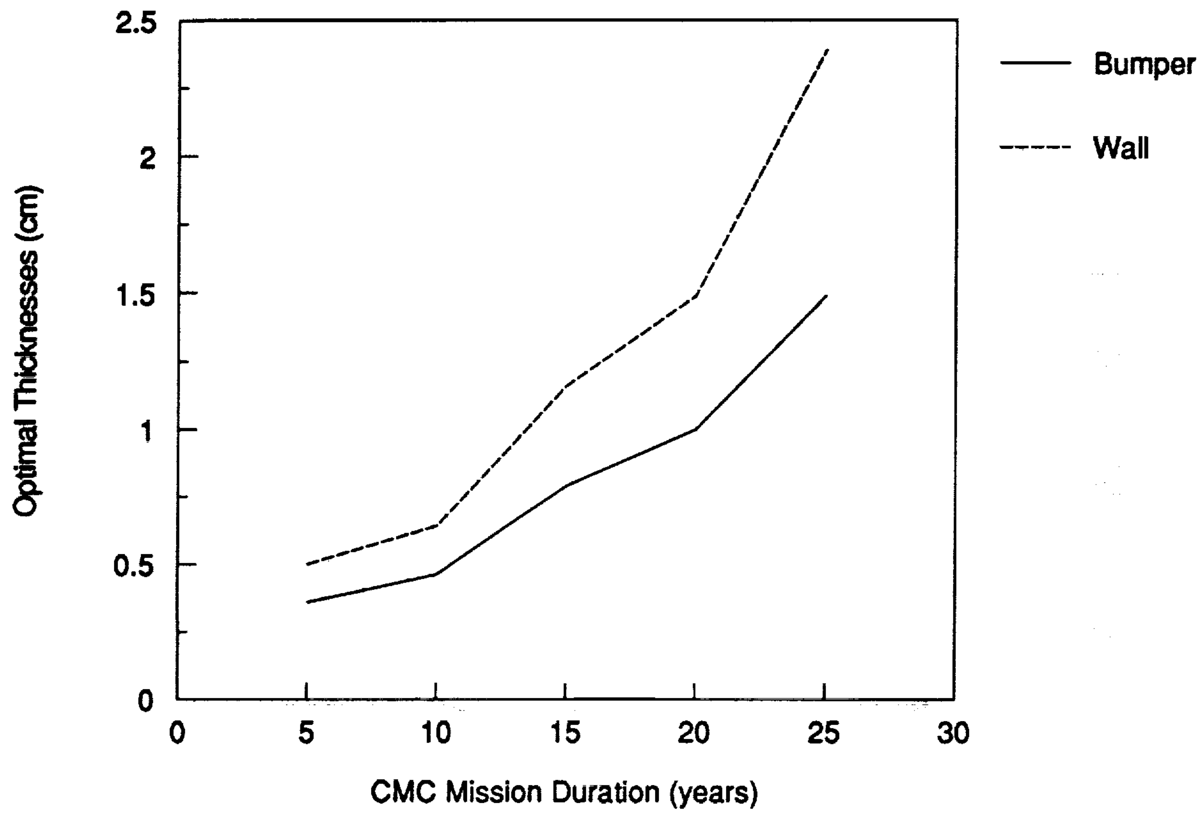



**Figure 2.7-14. Optimal CMC Thicknesses vs Mission Duration (2011-T8 Bumper)**

## 2.8 Uncertainty Considerations

Particle Diameter/Space Debris Growth Rate

**Figures 2.8-1 and 2.8-2** show the relationships between space debris growth rate and projectile diameter and minimum CMC weight, respectively. Note that the design implications are more severe than that indicated by the growth in projectile diameter. This is due to the fact that the structural response of the protective structures is a nonlinear function of projectile diameter growth. Additionally, note that an increase in space debris growth rate from 5% to 8% results in a 50% increase in minimum protective structures design weight. The optimal bumper and wall thicknesses as functions of space debris growth rate are given in **Figure 2.8-3**.

Note that the optimal ratio between bumper and wall is fairly constant up to about 6% debris growth rate, and then decreases as the wall thickness becomes a greater influence on protective structures design.

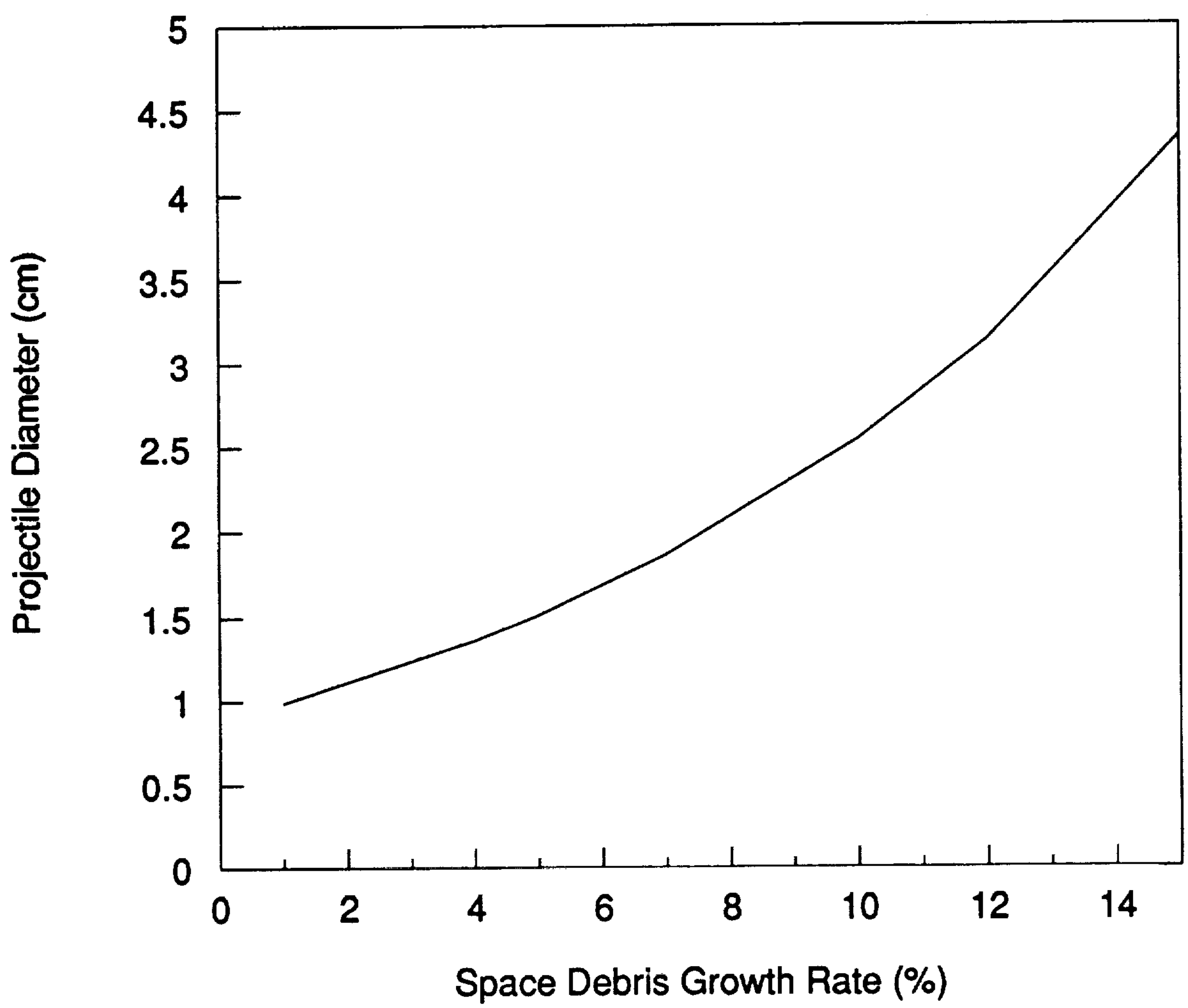



**Figure 2.8-1. Projectile Diameter vs Space Debris Growth Rate**

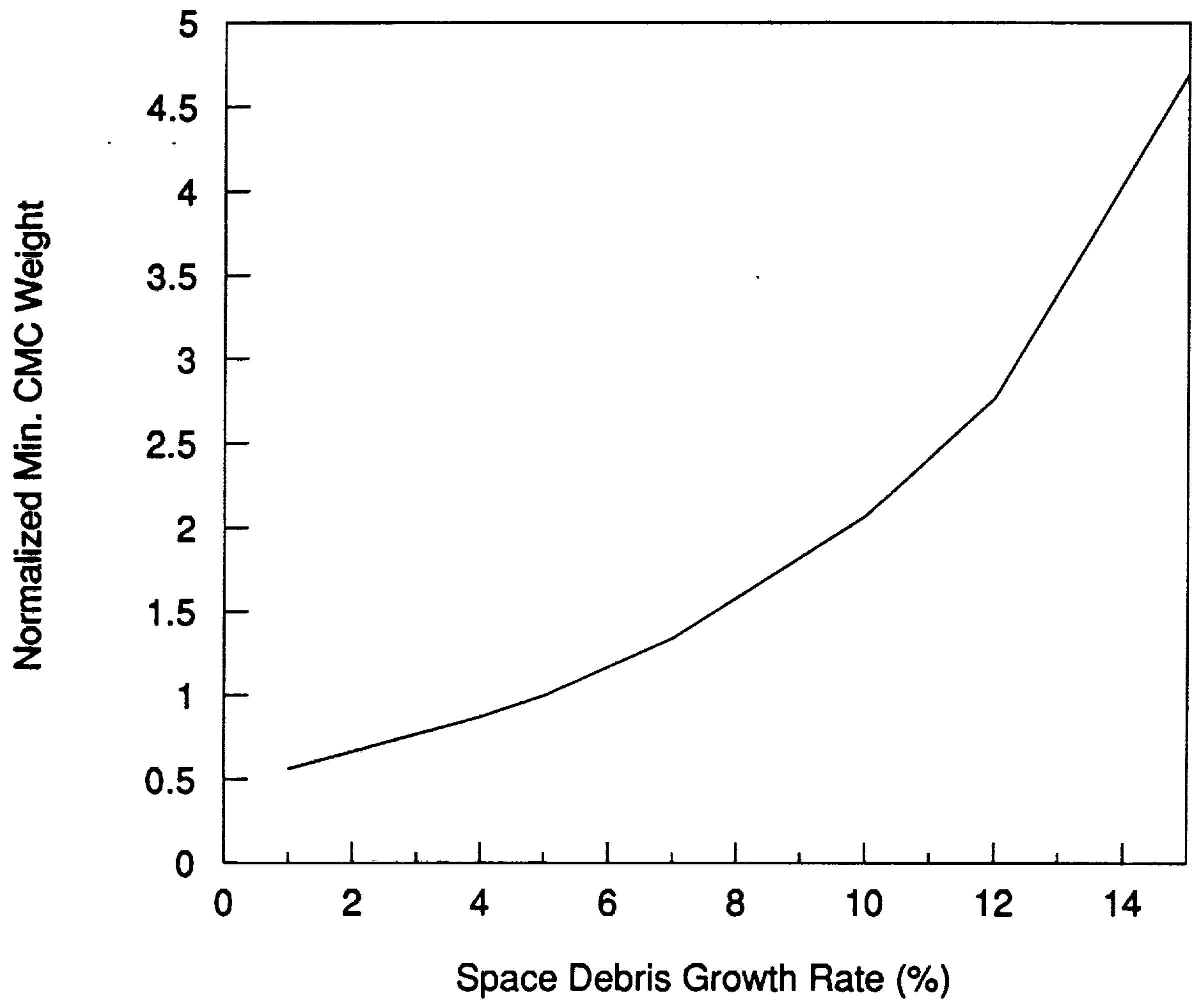



**Figure 2.8-2. Minimum CMC Weight vs Debris Growth Rate (Aluminum Alloys)**

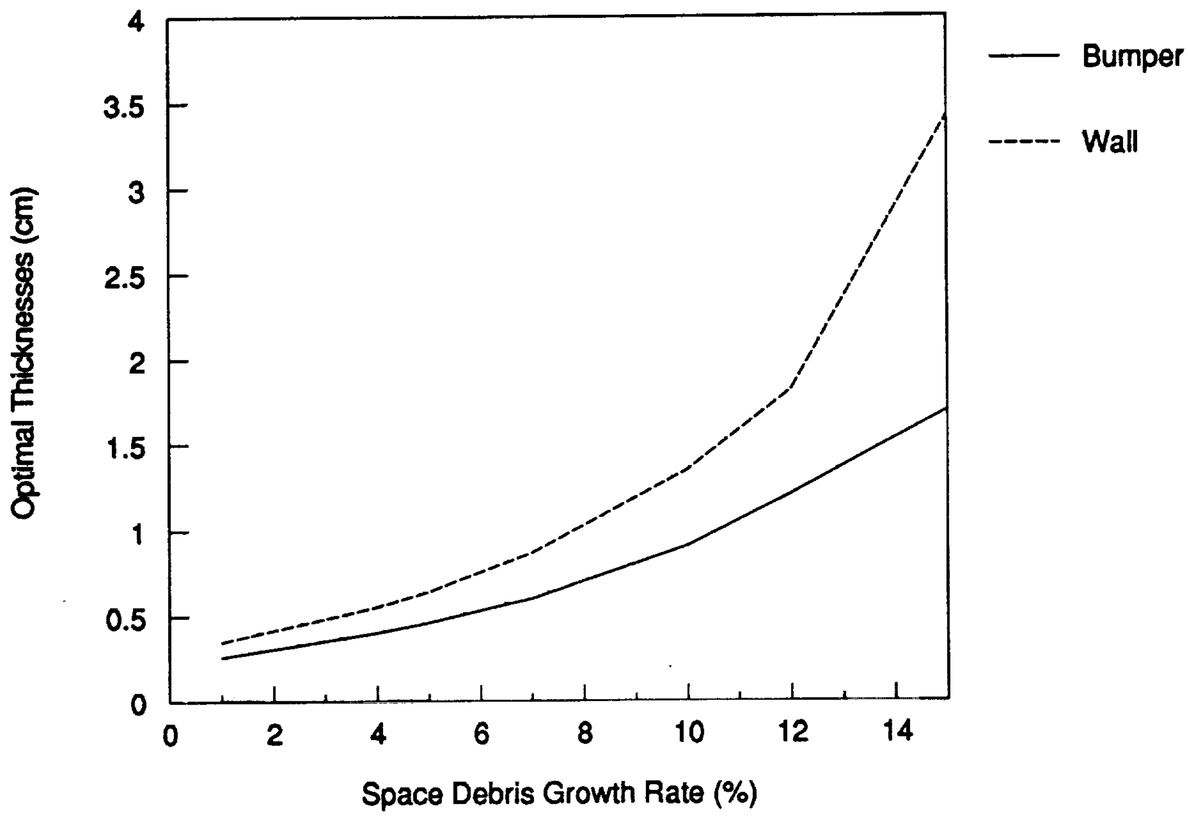



**Figure 2.8-3. Optimal CMC Thicknesses vs Debris Growth Rate (2011-T8 Bumper)**

Particle Shape/Density

The distribution of particle shapes for space debris in orbit is unknown. The potential variation in protective structures design effectiveness due to changes in particle shapes has been shown by hydrocode and impact test data to be relatively large.

The particle density is generally unknown as well. It is modelled as a decreasing function of projectile diameter as shown in **Figure 2.8-4**.

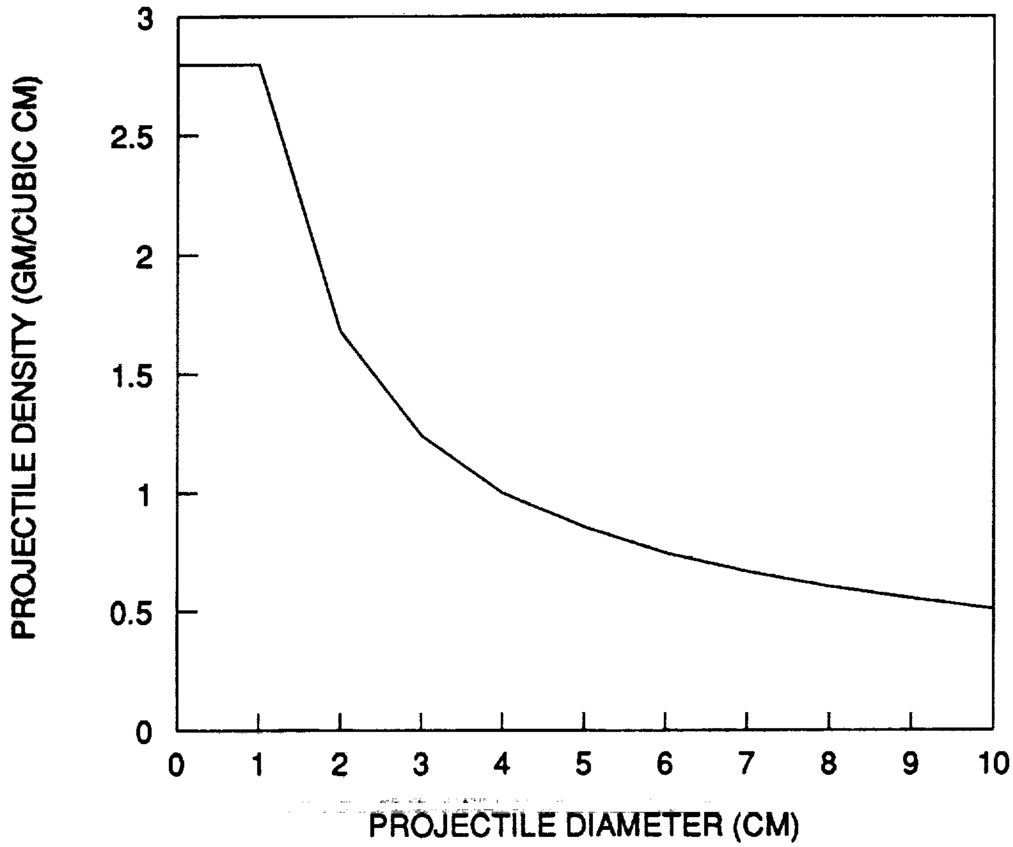



**Figure 2.8-4. Space Debris Particle Density vs Diameter**

<u>Uncertainties in Risk Parameters</u>

Although distributions exist for the risk parameters, these are subject to uncertainties in their accuracy and development. For instance, the distribution of projectile velocities is subject to uncertainties. Uncertainties in mission risk may be measured by establishing confidence intervals about the expected mission risk.

## 2.9 Second Order Parametric Analyses

This section includes numerous design trade parametrics to aid the designer in decision-making and design consequences of environment-related issues. The four independent variables shown are bumper/wall separation, space debris growth rate, CMC mission duration, and CMC mission risk.

<u>Bumper/Wall Separation</u>

**Figures 2.9-1** through **2.9-3** show the effects of bumper/wall separation on minimum CMC weight for various space debris growth rates, CMC mission durations, and CMC mission risks, respectively. Note, for instance, that the protective structures designer can maintain equivalent weight if the space debris growth rate is actually 7% by increasing the bumper/wall separation from 10 to 15 cm.

<u>Space Debris Growth Rate</u>

**Figures 2.9-4** through **2.9-6** show the effects of space debris growth rate on minimum CMC weight for various bumper/wall separations, CMC mission durations, and CMC mission risks, respectively. Note, for instance, that the protective structures designer can maintain equivalent weight if the space debris growth rate is actually 9% by increasing the mission risk from 3% to 5%.

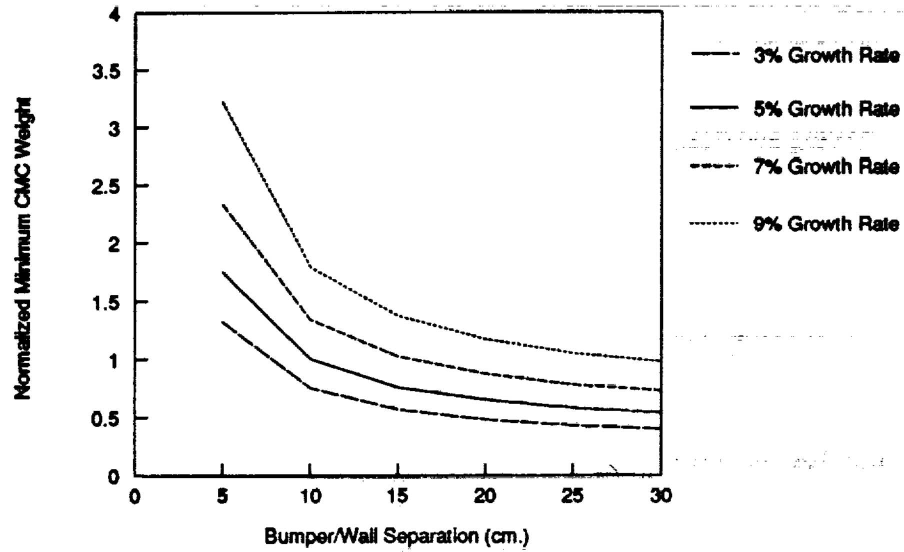



**Figure 2.9-1. Minimum Core Module Weight vs Bumper/Wall Separation for Various Space Debris Growth Rates (2011-T8 Aluminum)**

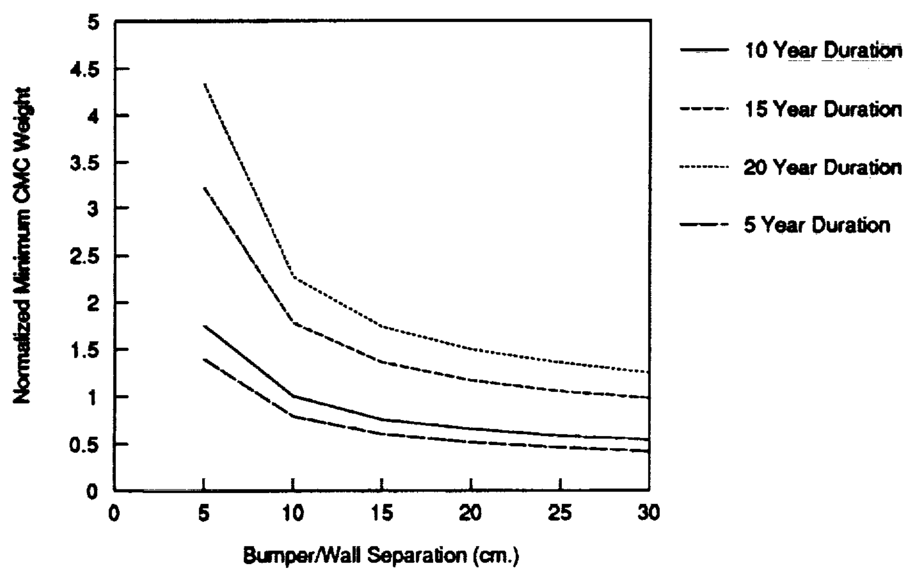



**Figure 2.9-2. Minimum Core Module Weight vs Bumper/Wall Separation for Various CMC Durations (2011-T8 Aluminum)**

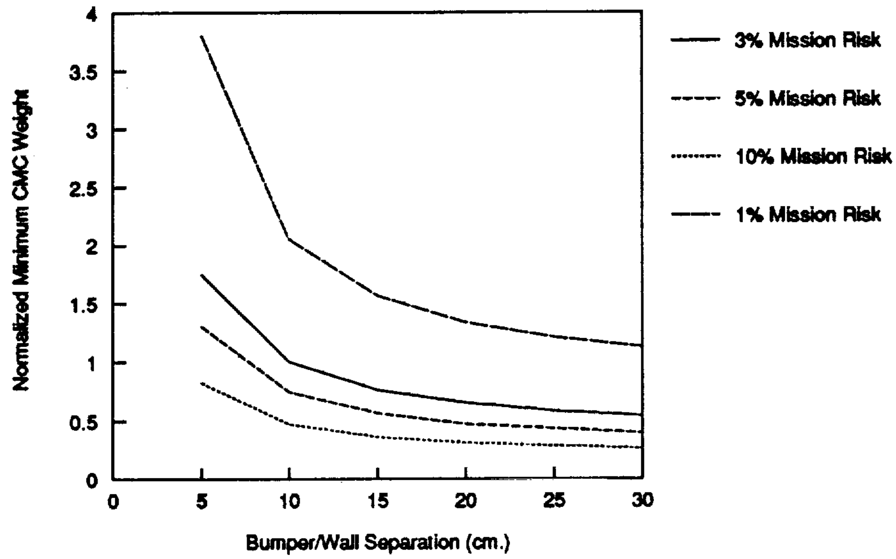



**Figure 2.9-3. Minimum Core Module Weight vs Bumper/Wall Separation for Various CMC Mission Risks (2011-T8 Aluminum)**

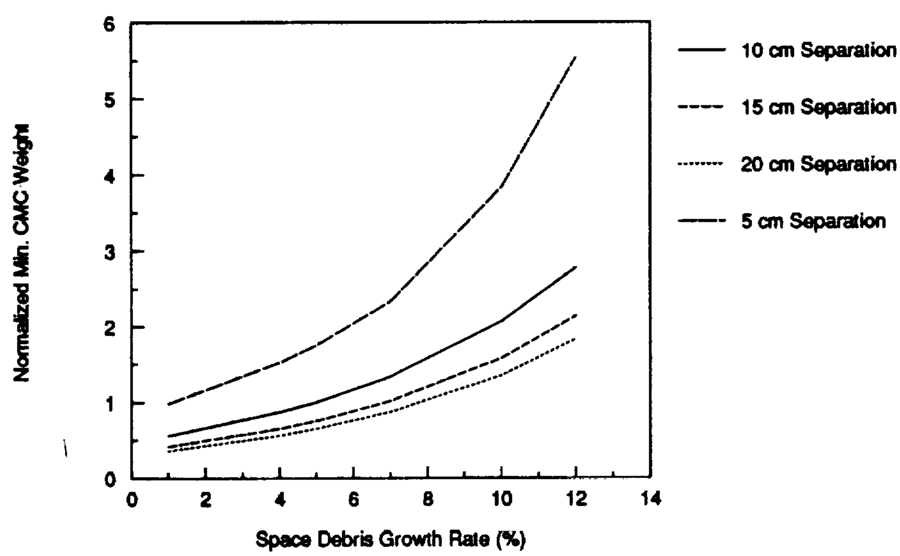



**Figure 2.9-4. Minimum Core Module Weight vs Space Debris Growth Rate for Various Bumper/Wall Separations (2011-T8 Aluminum)**

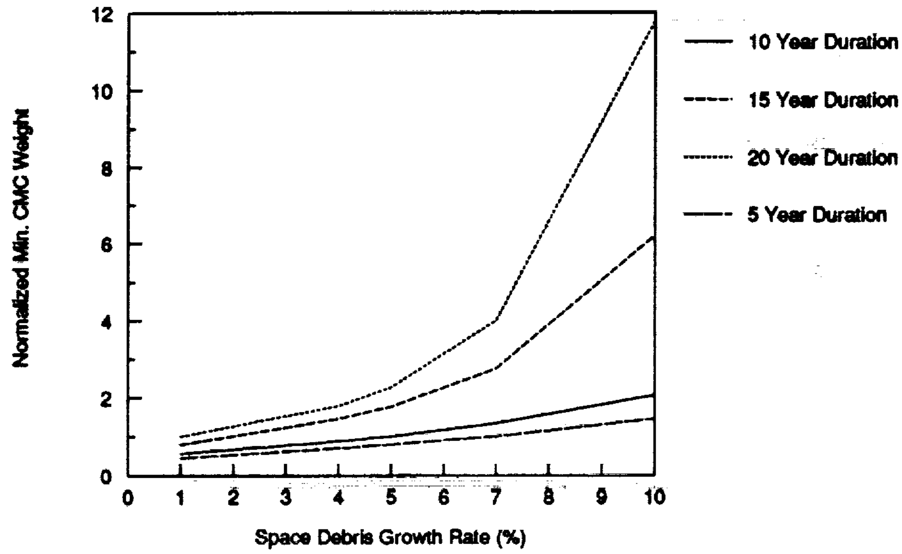



**Figure 2.9-5. Minimum Core Module Weight vs Space Debris Growth Rate for Various CMC Mission Durations (2011-T8 Aluminum)**

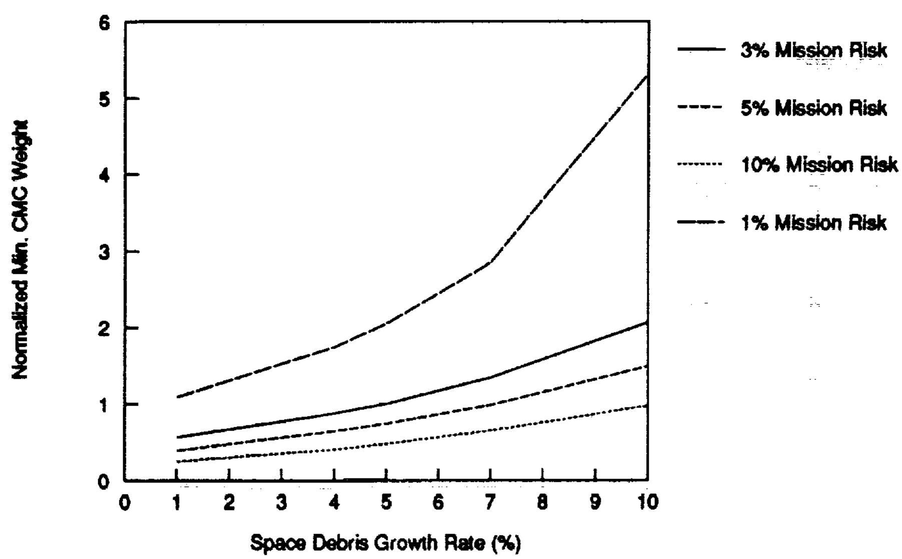



**Figure 2.9-6. Minimum Core Module Weight vs Space Debris Growth Rate for Various CMC Mission Risks (2011-T8 Aluminum)**

## CMC Mission Duration

**Figures 2.9-7** through **2.9-9** show the effects of CMC mission duration on minimum CMC weight for various bumper/wall separations, space debris growth rates, and CMC mission risks, respectively. Note, for instance, that if mission duration increases from 10 to 15 years, the protective structures designer can maintain equivalent weight by increasing the bumper/wall separation from 10 to 20 cm.

## CMC Mission Risk

**Figures 2.9-10** through **2.9-12** show the effects of CMC mission risk on minimum CMC weight for various bumper/wall separations, space debris growth rates, and CMC mission durations, respectively. Note, for instance, that if mission risk increases from 3% to 10%, the protective structures designer can afford to reduce the bumper/wall separation from 10 to 5 cm while maintaining weight.

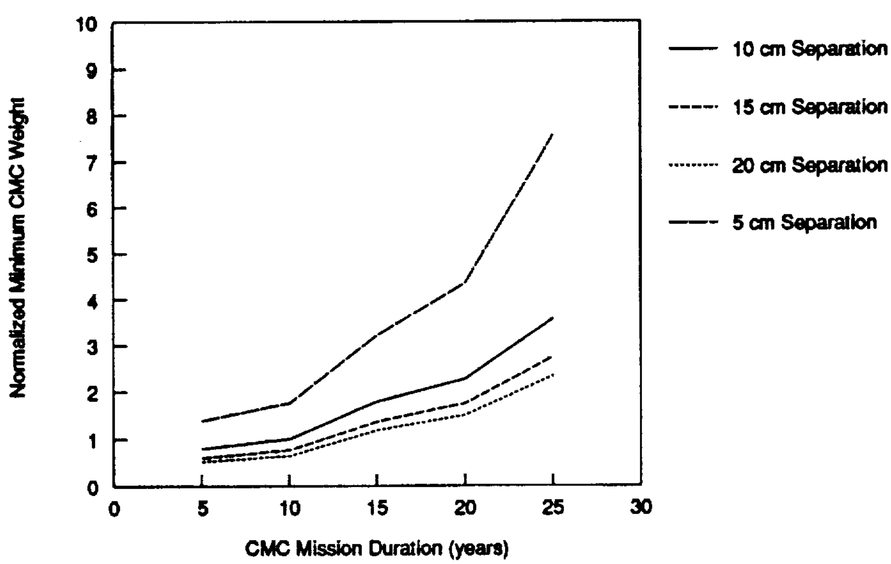



**Figure 2.9-7. Minimum Core Module Weight vs CMC Mission Duration for Various CMC Bumper/Wall Separations (2011-T8 Aluminum)**

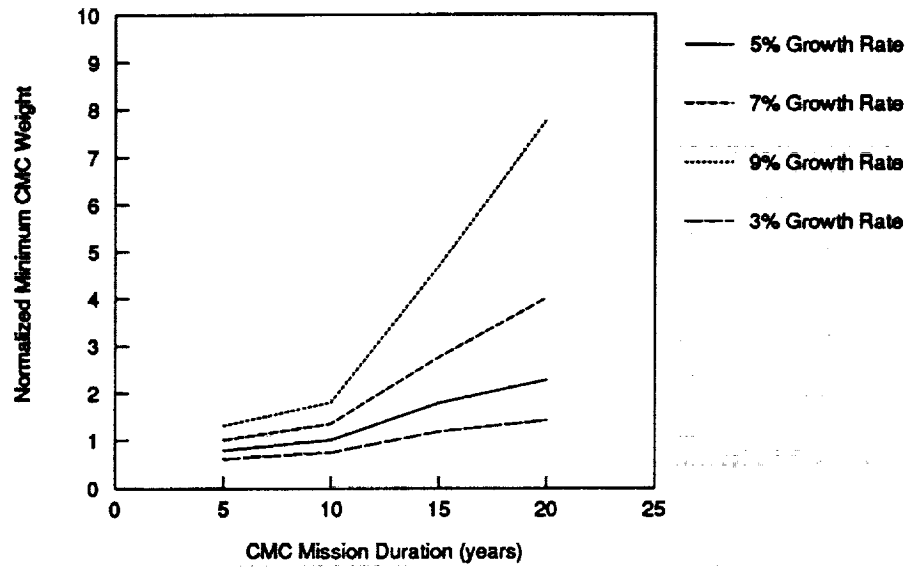



**Figure 2.9-8. Minimum Core Module Weight vs CMC Mission Duration for Various Space Debris Growth Rates (2011-T8 Aluminum)**

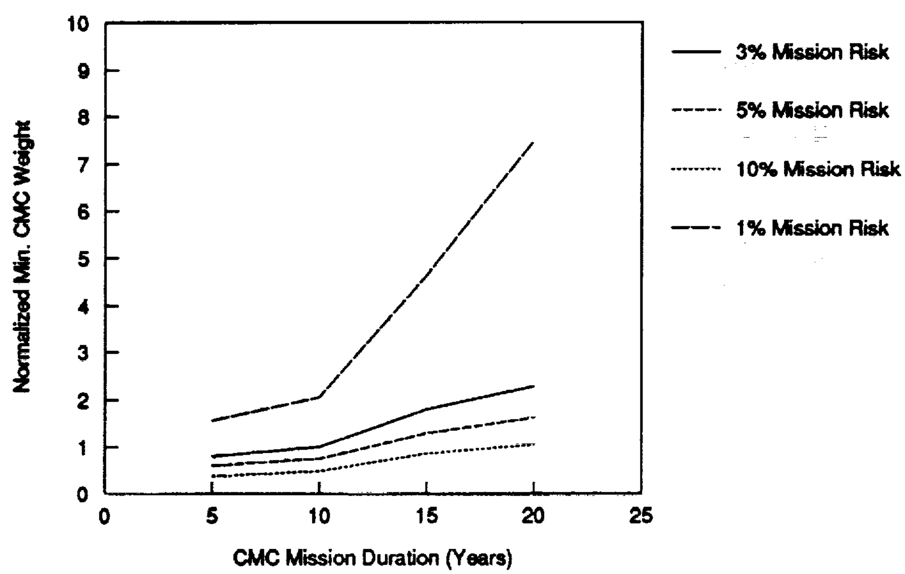



**Figure 2.9-9. Minimum Core Module Weight vs CMC Mission Duration for Various CMC Mission Risks (2011-T8 Aluminum)**

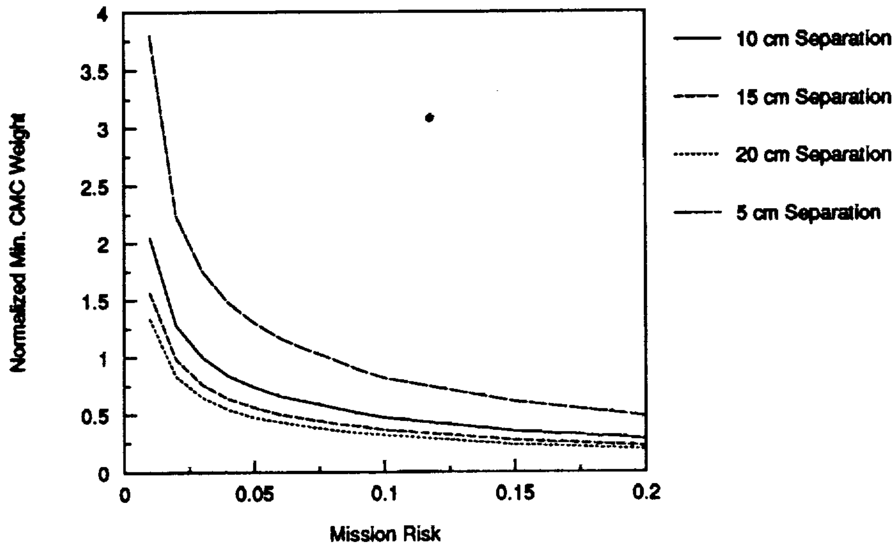



**Figure 2.9-10. Minimum Core Module Weight vs CMC Mission Risk for Various Bumper/Wall Separations (2011-T8 Aluminum)**

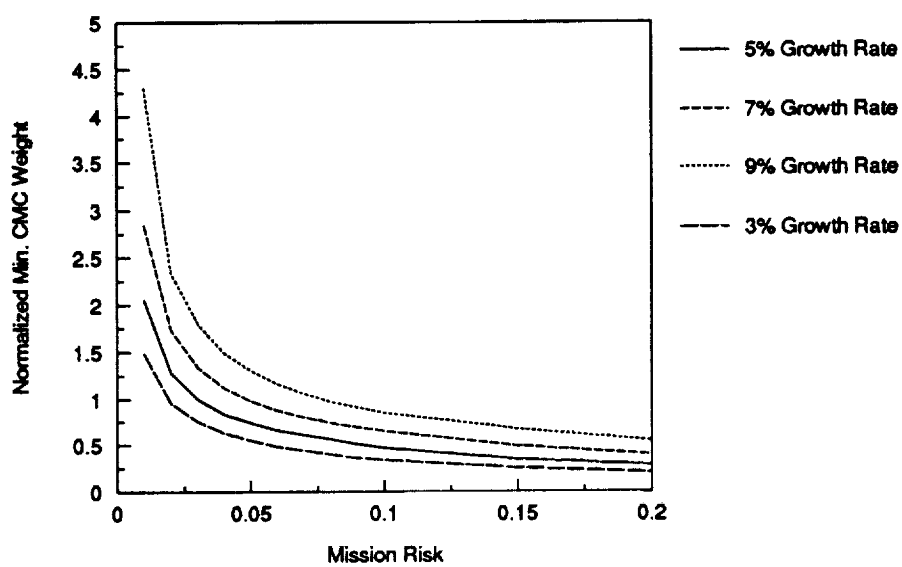



**Figure 2.9-11. Minimum Core Module Weight vs CMC Mission Risk for Various Space Debris Growth Rates (2011-T8 Aluminum)**

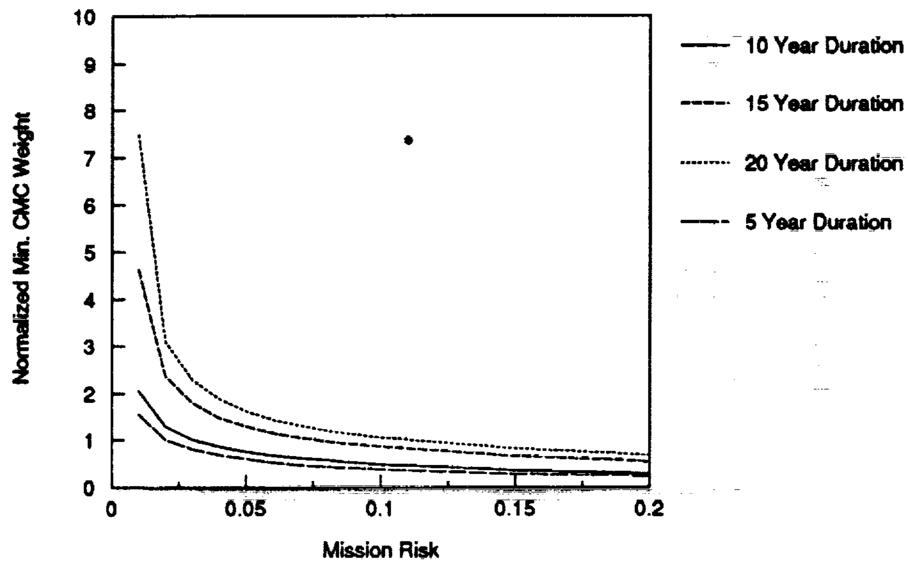



**Figure 2.9-12. Minimum Core Module Weight vs CMC Mission Risk for Various CMC Mission Durations (2011-T8 Aluminum)**

## 2.10 Conclusions and Recommendations For Section 2

**Conclusions**

| |
|---|
| 1. Global analytic nonlinear design optimization can be performed for the projectile melt/vaporization region (Wilkinson), for normal impacts in the projectile shatter region (Burch), and for the Nysmith predictor using Geometric Programming. |
| 2. For the predictors investigated, the optimal ratio of bumper mass per unit area to total mass per unit area may vary with mission, environment, projectile mass, and velocity regime. |
| 3. There is a large incentive for increasing the bumper/wall separation from 10 to 15 cm for all predictors investigated. |
| 4. All predictors reflect increasing design sensitivity to projectile diameter and decreasing design sensitivity to bumper/wall separation. |
| 5. The Wilkinson and Nysmith predictors reflect increasing design sensitivity to projectile velocity, while the Burch relationship is decreasing. |
| 6. For the combined predictors, 2011-T8 is the preferable aluminum alloy bumper choice for the baseline parameters. |
| 7. For the combined predictors, increasing the bumper/wall separation from 10 to 15 cm reduces the minimum module weight by 25%. |
| 8. Minimum CMC weight is very sensitive to space debris growth rate above 7% and Space Station altitude below 1000 km for the combined predictors. |
| 9. CMC protective structures design depends greatly on mission duration for the combined predictors. |
| 10. For the combined predictors, increasing the CMC mission risk from 3% to 5% reduces the minimum module weight by about 30%. |

**Recommendations**

| |
|---|
| 1. Alternate metallic bumper materials should be investigated. |
| 2. Uncertainty analyses should be performed relative to the space debris environment parameters. |
| 3. A combined meteoroid/space debris optimization algorithm should be implemented. |
| 4. Advanced materials should be investigated. |

# 3 PROTECTIVE STRUCTURES DESIGN OPTIMIZATION CODE (PSDOC) OVERVIEW

PSDOC (Protective Structures Design Optimization Code) was developed under NASA-MSFC Contract NAS8-37378 "Optimization Techniques Applied to Passive Measures for In-Orbit Spacecraft Survivability". The purpose of PSDOC is to provide a user-friendly PC environment for a number of design and analytical tools including IMPACT10V, developed by SAIC. Specific analysis areas for spacecraft protective structures design optimization include selection of environment, spacecraft characteristics and mission, and hypervelocity impact predictor models. The significant features of PSDOC are a menu-driven scenario and input capability, post-processing features, and file management system.

The application of SAIC's Flexible Model - Graphical User Interface to PSDOC was but one utilization of this software. The graphical user interface (GUI) environment used for PSDOC was developed for assisting technical personnel in gaining access to computer based models without a thorough knowledge of the code itself. Other applications are easily fitted to existing models by SAIC engineers and software scientists. Attachments of this GUI software to existing models or "Retrofitting" allows for newer coding techniques and hardware technology advancements to be immediately available to older, validated models without affecting the code's reliability. Once this initial connection has been made and checked with the original version, additional input and output alterations to the model are often desired and can be handled by SAIC staff under the direction of our customers.

The PSDOC environment (retrofit to IMPACT10) was developed in coordination with Sherman Avans and Jennifer Robinson of NASA-MSFC and Robert Mog, Andy Laidig, and Kevin

Leonard of SAIC. The PSDOC user's manual delivered to NASA-MSFC in Aug. 1990 presents an overview of the windowing techniques and operating instructions for use of the PSDOC environment. This manual contains all the necessary information for efficient use of PSDOC.

# 4 MONTE CARLO SIMULATION ANALYSIS TOOL

## 4.1 Monte Carlo Simulation Purpose

The purpose of this simulation is to provide a statistical tool to address and quantify protective structures design risks, uncertainties, and options, and to address system-level issues relevant to designer decision-making and possible implications. The system of initial interest is the structural configuration of WP01, including the Core Module Configuration. "Grow-to" systems include module internal configurations and external structures (trusses, solar arrays, etc.) as specified in the redesign.

Initial investigations of interest include statistical analyses of primary impacts, penetrations, and vulnerable areas. "Grow-to" investigations include interior effects, secondary ricochet effects, and SSF element interrelations.

Risk considerations include environment particle velocity, impact angle, and component probability of impact. Uncertainty considerations include SSF IOC/FOC, particle diameter, mass-density, shape, and uncertainties in particle velocity and impact angle distributions.

## 4.2 Monte Carlo Simulation Development Approach

The tool development approach is to define the current SSF mission parameters and design configuration, and interpret the geometry mathematically using FASTGEN. The mission parameters drive requirements specification, including environment definitions. These considerations, combined with appropriate random number modules and the FASTGEN results, produce the necessary shotline time histories and intersecting body calculations. Survivability assessments follow and employ deterministic models for hypervelocity penetration prediction. Statistical assessments follow to supply answers to the questions of interest.

The top-level version of the Monte Carlo simulation tool will be executed on IBM-compatible PC's. The current version of this tool runs on a VAX. It is anticipated that the detailed version of this tool will operate on a CRAY. I/O requirements are discussed in Section 4.4.

Verification and validation of the tool will be performed using a combination of PSDOC and BUMPER. If the program execution times are considerable, a design of experiments approach will be used to specify production run matrices.

## 4.3 Particle Time-Arrival Process for Monte Carlo Simulation Development

Several algorithms have been developed for the particle time-arrival process. The standard assumption in this area is that arrival times are Poisson distributed. This means that the inter-arrival times are exponentially distributed, and sorting of arrival times is not required. Mean data is derived from the environment flux and appropriate spacecraft areas. This algorithm leads to a terminating simulation defined by the mission profile.

Realistically, however, the meteoroid and debris environments are both nonstationary Poisson processes, at best, since the mean arrival rates vary in time over the mission profile. An approximation algorithm has been developed which alters the mean arrival rate to represent the time period under consideration. However, this algorithm is not exact, since a period of high arrival rates could be neglected using a low arrival rate corresponding to the previous period, or vice versa. Thus, a more exact (continuous) algorithm should be developed. The approximating algorithm for the space debris environment is given as:

1. $Input T_i, T_f, d_{max}, d_{min}, \Delta d.$ $Set \quad t = T_i.$

2. Using equations [1]-[7], develop a cumulative flux-diameter distribution:

$$P(x \le d) = \frac{\sum_{x=0}^{\left[\frac{d+1-d_{min}}{\Delta d}\right]_{l.i.}+1} F\left(d_{min} + \sum_{x=0}^{\left[\frac{d+1-d_{min}}{\Delta d}\right]_{l.i.}+1} \Delta dx, h, i, t, s\right)}{\sum_{x=0}^{\left[\frac{d_{max}+1-d_{min}}{\Delta d}\right]_{l.i.}+1} F(d_{min} + \sum_{x=0}^{\left[\frac{d_{max}+1-d_{min}}{\Delta d}\right]_{l.i.}+1} \Delta dx, h, i, t, s)}$$

$$\forall d = d_{min} + \Delta dx, x \in \left[0, \frac{d_{max} - d_{min}}{\Delta d}\right]$$

3. Draw $U_1 \sim U[0,1]$.

4. Find $F \ni P(x \le d) = U_1$.

5. Set $\beta = 1/F$.

6. Draw $U_2 \sim U[0,1]$.

7. Set $\Delta t = -\beta \ln(U_2)$.

8. Update simulation time: $t = t + \Delta t$.

9. Is $t \ge T_f$?

No, then go to 2 to create next event.

Yes, then quit and gather statistics.

If independent mean and variance data for arrival rates are available, a uniform arrival process may be used as an alternative to Poisson arrivals. To compare this approach with the Poisson process, the variance may be set equal to the square of the mean. An algorithm has been developed for independent mean and variance data.

Augmentation/repair times may be modelled using a number of distributions, if this modelling is of interest. If mean data only for time to repair is available, an exponential service model may be used. If independent mean and variance data are available, the gamma, weibull, lognormal, or beta distributions may be appropriate.

## 4.4 Simulation Status

To date, the following items have been completed:

A. Enveloping Geometries Established For:
   1. Sphere: Enter Radius
   2. Cylinder: Enter Radius, Length
   3. Box: Enter length, Width, Height

B. Nonstationary Poisson Arrival Process Algorithm For Space Debris
   1. First Random Variate Establishes Point On Cumulative Flux-Diameter Curve At Current Mission Time
   2. Absolute Flux Is Inverted To Give Mean Interarrival Time
   3. Second Random Variate Establishes Time Between Arrivals Using Exponential Distribution
   4. Cumulative Flux-Diameter Curve Is Updated For New Mission Time

C. Impact Characteristics For Space Debris
   1. Impact Velocity
   2. Impact Angle
   3. Particle Density/Mass

D. Look-Up Tables
   1. Solar Flux (Monthly)
   2. Inclination Factor
   3. Flux-Diameter Curves

E. Impact History Data Including Event Time, Diameter, Density, Mass, Velocity, Angle

F. Fixed Time Data Including Absolute Flux, Normalized Flux, Cumulative Normalized Flux Distributions As Functions of Diameter

The following items remain to be completed:

I. Impact Location/Orientation
II. Meteoroid Environment
III. Geometry/Shotline Integration
IV. Graphical Outputs
  A. Time/Interarrival Time Histories and Distributions
  B. Particle Diameter/Mass, Velocity Distributions
  C. Particle Location Display/Contours
  D. Statistics Modules
  E. 3-D Plots, e.g. Diameter vs. Velocity vs. Time

# 5 DEVELOPMENT OF ADVANCED SHIELDING CONCEPTS

## 5.1 Introduction

The development of advanced shielding concepts presented in this section is a preliminary theoretical modification of the Wilkinson and ballistic PEN4 predictors to multiple bumper situations. The intent is to perform this preliminary analysis, and then correlate the results with existing test data to improve the models.

## 5.2 Extension to Multiple Bumpers for Wilkinson Predictor

A number of different approaches have been attempted to modify the Wilkinson predictor mathematically for multiple bumper systems. The one successful approach (physically) is given as follows:

1. Modify the Wilkinson form in a product sense as:

$$t_n = \frac{0.364 D^3 \rho_p V \cos(\theta)}{L_n \left( \prod_{i=1}^{n-1} S_i^2 \right) \rho_n} \quad \text{for} \quad \frac{D \rho_p}{\prod_{i=1}^{n-1} \rho_i t_i} \le 1, \tag{98}$$

$$t_n = \frac{0.364 D^4 \rho_p^2 V \cos(\theta)}{L_n \left( \prod_{i=1}^{n-1} S_i^2 \right) \left( \prod_{i=1}^{n-1} \rho_i t_i \right) \rho_n} \quad \text{for} \quad \frac{D \rho_p}{\prod_{i=1}^{n-1} \rho_i t_i} > 1. \tag{99}$$

If our goal is to minimize system mass per unit area subject to the total separation between first bumper and last wall equal to some desired value, we may write this as

$$\min W = \sum_{i=1}^{n-1} m_i + \frac{0.364 D^4 \rho_p^2 V \cos\theta}{L_n \left( \prod_{i=1}^{n-1} S_i^2 \right) \left( \prod_{i=1}^{n-1} m_i \right)} \tag{100}$$

$$s.t. \sum_{i=1}^{n-1} S_i = S_{ToT} \tag{101}$$

$$where \quad m_i = \rho_i t_i \tag{102}$$

$S_{ToT}$ is the total separation between the first bumper and the wall, and n-1 is the total number of bumpers (n is the total number of plates).

Under condition [99], the dual Geometric Programming objective function is given by

$$\max v(\delta) = \prod_{i=1}^{n-1} (1/\delta_i)^{\delta_i} (K/\delta_n)^{\delta_n} \mu_1^{\mu_1} \prod_{j=1}^{n-1} \left( \frac{1}{S_{ToT}\delta_j'} \right)^{\delta_j'} \qquad [103]$$

$$K = \frac{0.364 D^4 \rho_p^2 V \cos(\theta)}{L_n} \qquad [104]$$

$$\sum_{i=1}^{n} \delta_i = 1 \qquad [105]$$

$$\delta_i - \delta_n = 0, \qquad i = 1,2,\ldots,n-1 \qquad [106]$$

$$-2\delta_n + \delta_j' = 0, \qquad j = 1,2,\ldots,n-1 \qquad [107]$$

$$\mu_1 = \sum_{j=1}^{n-1} \delta_j' \qquad [108]$$

Note that the degree of difficulty is 0, with 2n-2 independent variables corresponding to the n-1 bumper areal densities and the n-1 separations.

Equations [106] and [107] together imply

$$\delta_i = \delta_n = 1/n, \qquad i = 1,2,\ldots,n-1 \qquad [109]$$

$$\delta_j' = 2\delta_n = 2/n, \qquad j = 1,2,\ldots,n-1 \qquad [110]$$

The minimum weight and globally areal densities are given by

$$W_0 = n \left[ \frac{0.364 D^4 \rho_p^2 V \cos(\theta)}{L_n} \right]^{1/n} \left( \frac{n-1}{S_{ToT}} \right)^{\frac{2n-2}{n}} \qquad [111]$$

$$m_{i_0} = \left[ \frac{0.364 D^4 \rho_p^2 V \cos(\theta)}{L_n} \right]^{1/n} \left( \frac{n-1}{S_{ToT}} \right)^{\frac{2n-2}{n}}, \qquad i = 1,2,\ldots,n \qquad [112]$$

The optimal individual separations are given by

$$S_{j_0} = \frac{S_{ToT}}{n-1}, \qquad j = 1,2,\ldots,n-1 \qquad [113]$$

The optimal separations are equal and uniformly distributed over the total available separation. Thus, the globally optimal algorithm for the multi-bumper Wilkinson Predictor is

1. Determine $\prod_{i=1}^{n-1} m_{i_0}$ from equation [112].

2. Compute $\dfrac{D\rho_p}{\prod_{i=1}^{n-1} m_{i_0}}$.

3. If $\dfrac{D\rho_p}{\prod_{i=1}^{n-1} m_{i_0}} > 1$, then quit.

4. If $\dfrac{D\rho_p}{\prod_{i=1}^{n-1} m_{i_0}} \leq 1$, the optimal design is $\left(m_{1_0}, m_{2_0}, \ldots, m_{n_0} / \left(\dfrac{D\rho_p}{\prod_{i=1}^{n-1} m_{i_0}}\right)\right)$.

## 5.3 Results

Several results using the development of Section 5.2 are given in this section. The baseline assumptions are a particle density of 2.8 gm/cm$^3$, velocity of 9 km/sec, diameter of 1 cm, impacting normally into a configuration with a total bumper/wall separation of 10 cm.

**Figures 5.3-1 and 5.3-2** show how the optimal protective structures design configuration varies with number of bumpers for projectile diameters of 1 and 3 cm, respectively. Note that for a 1 cm particle diameter, the optimal number of bumpers is 2, while for 3 cm, it is 3 bumpers. Also, note the significant penalty for choosing the wrong number of bumpers in these cases, as well as the lack of symmetry of these penalties about the optimal number of bumpers.

**Figure 5.3-3** shows the optimal protective structures design configuration including optimal number of bumpers as a function of particle diameter. Increasing particle diameter results in an increasing optimal number of bumpers to defeat the particle. Note the optimal transition regions between 1 and 2 bumpers (corresponding to particle diameters between 0.75 and 1 cm) and 2 and 3 bumpers (corresponding to particle diameters between 2.25 and 2.5 cm). Also, note the very linear minimum system areal density, showing the stabilizing effect of increasing the number of bumpers in the configuration.

**Figure 5.3-4** shows the optimal protective structures design configuration including optimal number of bumpers as a function of particle velocity. The most striking feature of this trade is the relative insensitivity to velocity for a dual bumper system.

**Figure 5.3-5** shows the optimal protective structures design configuration as a function of total bumper/wall separation. As in previous studies, there is a large weight incentive for increasing the total separation. Furthermore, increased separation allows for more bumpers to disrupt the incoming particle.

**Figure 5.3-6** is a replica of **Figure 5.3-5**, except that the optimal individual separations are included.

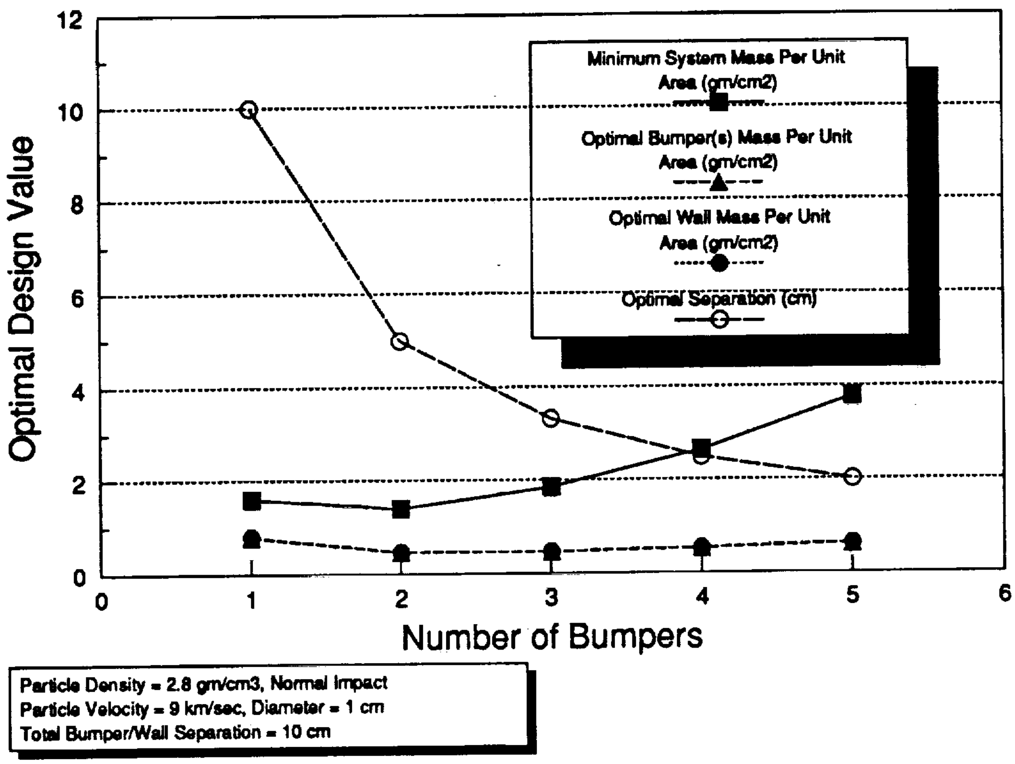
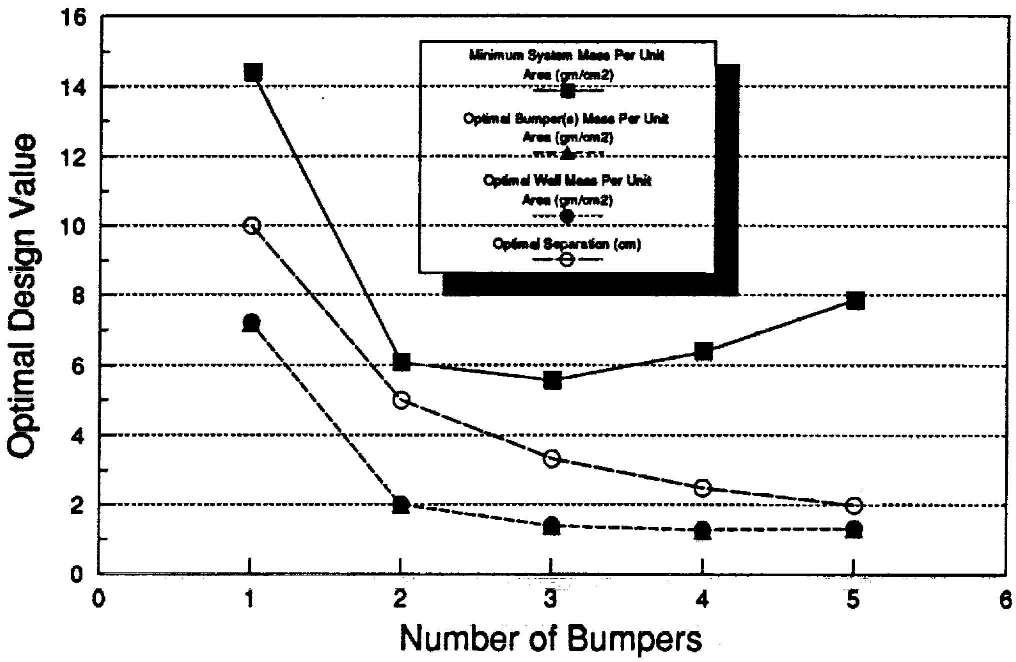



**Figure 5.3-1. Determining Optimal Number of Bumpers for Modified Wilkinson**




Particle Density = 2.8 gm/cm3, Normal Impact
Particle Velocity = 9 km/sec, Diameter = 3 cm
Total Bumper/Wall Separation = 10 cm

**Figure 5.3-2. Determining Optimal Number of Bumpers for Modified Wilkinson**

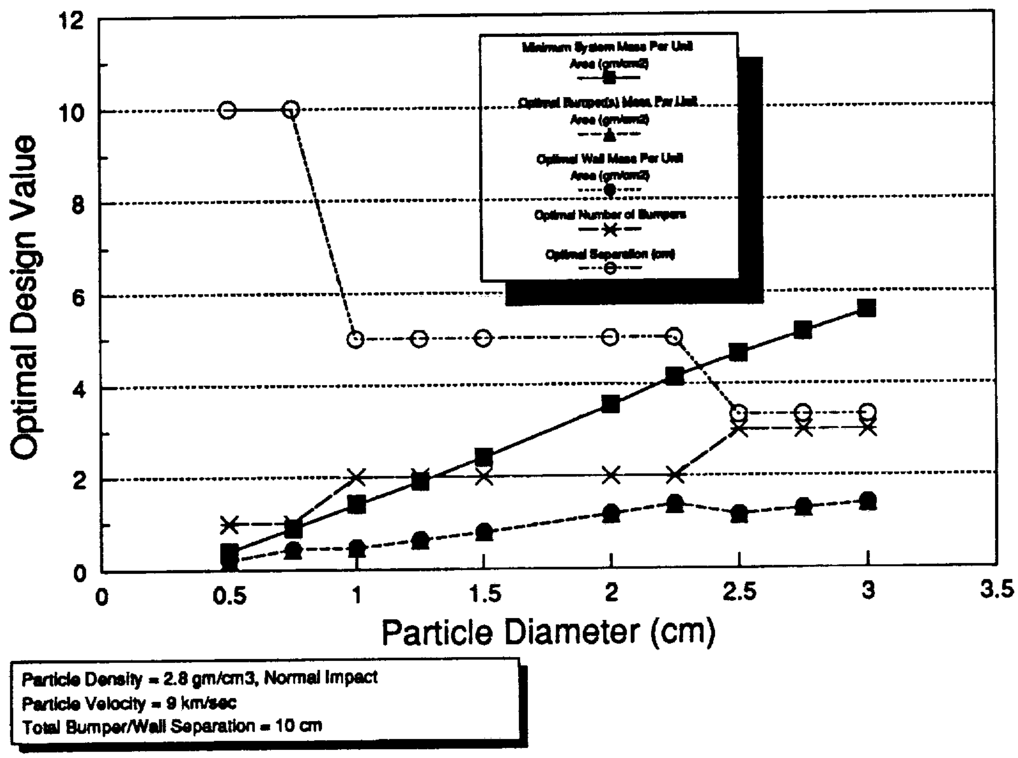



**Figure 5.3-3. Optimal Protective Structures Design Values vs. Particle Diameter (Modified Wilkinson)**

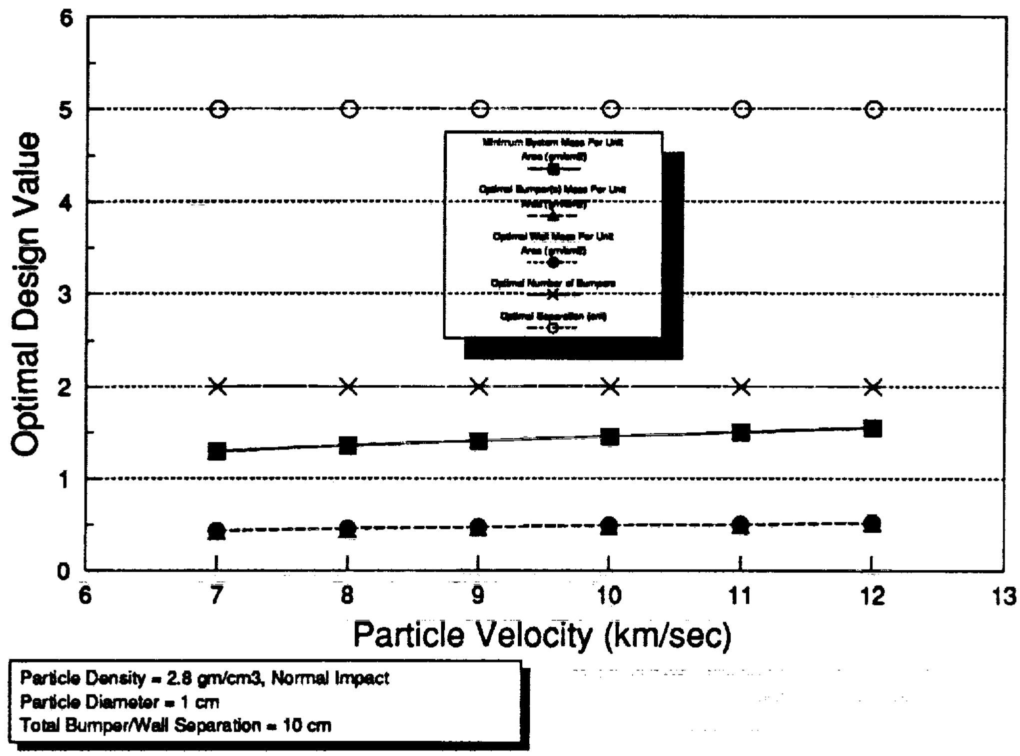



**Figure 5.3-4. Optimal Protective Structures Design Values vs. Particle Velocity (Modified Wilkinson)**

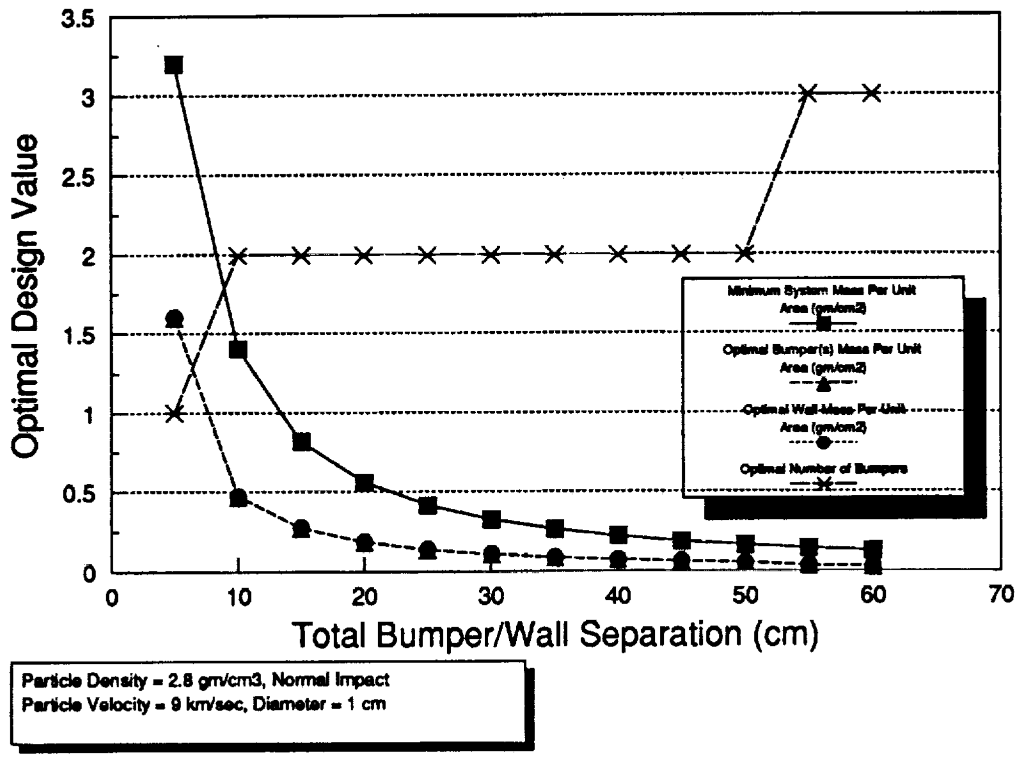



**Figure 5.3-5. Optimal Protective Structures Design Values vs. Total Bumper/Wall Separation (Modified Wilkinson)**

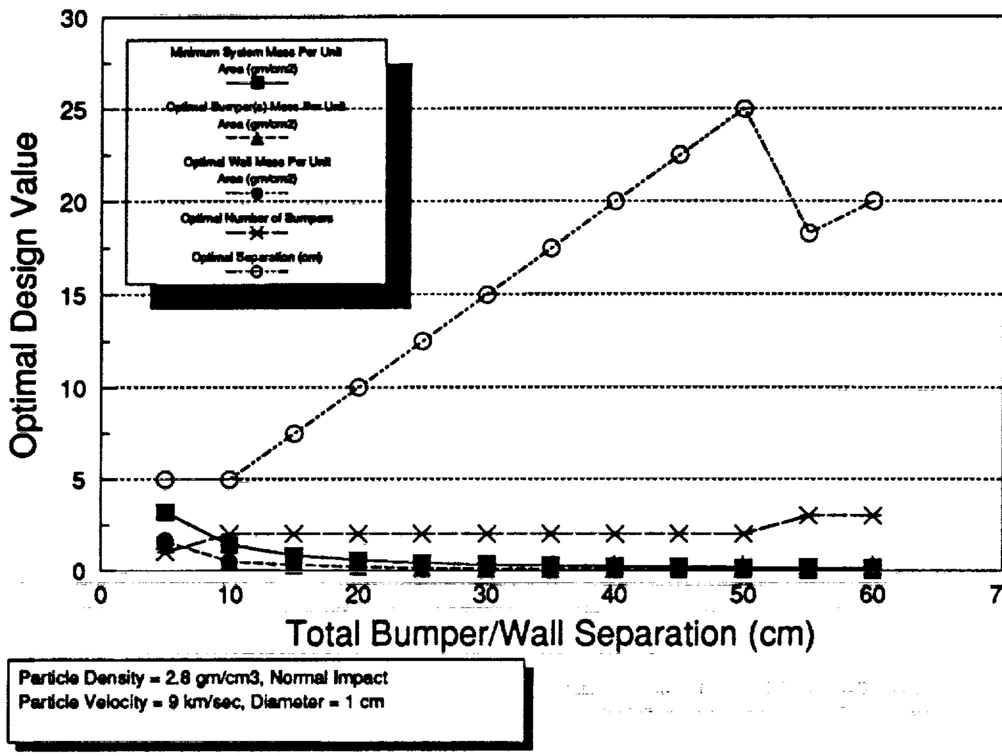



**Figure 5.3-6. Optimal Protective Structures Design Values vs. Total Bumper/Wall Separation (Modified Wilkinson)**

## 5.4 Extension to Multiple Bumpers for Ballistic PEN4 Predictor

The multiple bumper recursion equations are given by:

$$V_f = 4100, \quad \frac{T_1}{D} \leq 0.4 \tag{114}$$

$$V_f = 4986\left(\frac{T_1}{D}\right)^{0.21}, \quad \frac{T_1}{D} > 0.4 \tag{115}$$

$$V_{50_j} = \left[\left(\frac{0.6T_j}{\left(\frac{0.281D\rho_p}{\rho_j}\right)^{1/3}\cos(\theta)}\right)^{1/0.31}\frac{2S_{y_j}}{\rho_p}\right]^{1/2} \tag{116}$$

The first bumper is penetrated if

$$V > V_{50_{j=1}} \tag{117}$$

The residual velocity (from the first bumper) is

$$V_{R_1} = \left[ \frac{1.33V^2R_p^2\rho_p - \left(8S_{y_1}T_1e^{-0.0003125V}\right)/\cos(\theta)}{1.33R_p^2\rho_p + R_pT_1\rho_1/\cos(\theta)} \right]^{1/2} \tag{118}$$

The second bumper is penetrated if

$$V_{R_1} > V_{50_{j=2}} \tag{119}$$

The residual velocity (from the second bumper) is

$$V_{R_2} = \left[ \frac{1.33V_{R_1}^2R_p^2\rho_p - \left(8S_{y_2}T_2e^{-0.0003125V_{R_1}}\right)/\cos(\theta)}{1.33R_p^2\rho_p + R_pT_2\rho_2/\cos(\theta)} \right]^{1/2} \tag{120}$$

The third bumper is penetrated if

$$V_{R_2} > V_{50_{j=3}} \tag{121}$$

.

.

.

The residual velocity (from the (n-1)st bumper) is

$$V_{R_{n-1}} = \left[ \frac{1.33V_{R_{n-2}}^2R_p^2\rho_p - \left(8S_{y_{n-1}}T_{n-1}e^{-0.0003125V_{R_{n-2}}}\right)/\cos(\theta)}{1.33R_p^2\rho_p + R_pT_{n-1}\rho_{n-1}/\cos(\theta)} \right]^{1/2} \tag{122}$$

The nth bumper is penetrated if

$$V_{R_{n-1}} > V_{50_{j=n}} \tag{123}$$

Given 6061-T6 aluminum bumper materials (yield strength of 35 ksi, density of 2.71 gm/cm$^3$, total thickness of 0.16 cm), 2219-T87 aluminum wall (yield strength of 51 ksi, density of 2.81 gm/cm$^3$, thickness of 0.3175 cm), a projectile density of 2.81 gm/cm$^3$, and a projectile impact angle of 0 degrees (normal), **Figure 5.4-1** shows the ballistic limit curves for single, double, and triple bumper configurations. Note the relatively minor sensitivity to number of bumpers over this limited range.

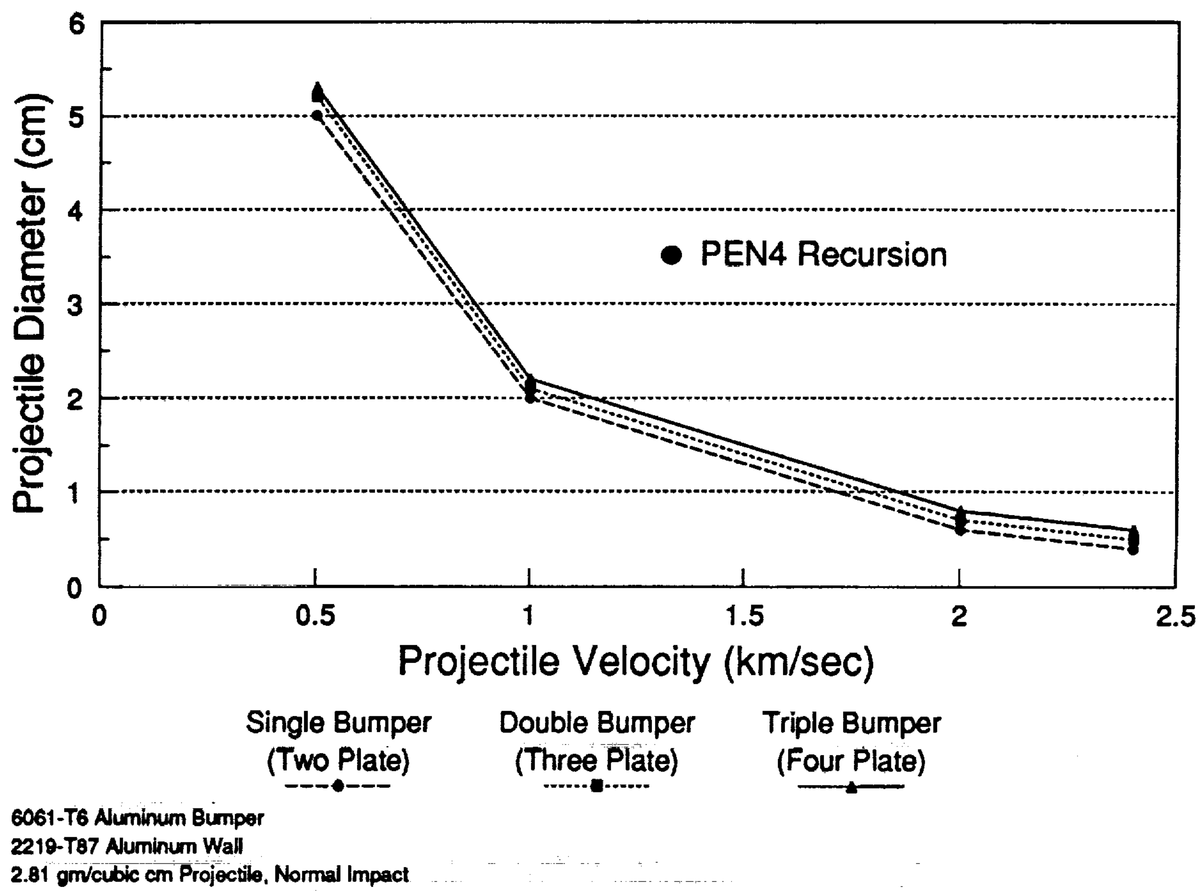



**Figure 5.4-1 Critical Diameter vs. Projectile Velocity for Multiple Bumper Systems Using Ballistic PEN4 Recursion**

## 5.5 Advanced Shielding Concepts Status

To date, these multibumper concepts have been shown for a theoretical modification of the Wilkinson predictor, as well as for the ballistic PEN4 predictor. It is recommended that these concepts be extended to the Burch predictor, and that the Wilkinson extension be correlated with hydrocode data and the Burch extension with impact test data.

# 6 DISCRETE PROTECTIVE STRUCTURES DESIGN OPTIMIZATION

## 6.1 Introduction

### Background

Within the field of nonlinear programming lies a technique called geometric programming. Geometric programming is a purely algebraic method that provides global, and often analytic, solutions to certain problems previously discussed in Section 2.3. These problems are called posynomial programs, and they are generally nonconvex, nonlinearly constrained formulations. The field of geometric programming has been extended to programs which are not posynomials;[43,44,117-122] however, the global features of the solution are not retained in this extension. Thus, the term posynomial programming, sometimes called prototype geometric programming, refers only to programs composed entirely of posynomials.

In general, discrete nonlinear optimization techniques are even less capable than continuous ones of providing global and analytic solutions.[123] In particular, many current discrete nonlinear techniques employ branch and bound derivatives, which generally do not result in global optimization properties, except for convex programming problems. Discrete posynomial problems which can be transformed to prototype geometric programs, on the other hand, result in global optimization upon transfer to the dual. The general transformation can then be applied to engineering design problems with independent variables restricted to standard or discrete availabilities.

### Subtask Goal

This subtask addressing the development and application of discrete nonlinear optimization techniques is not required in the Statement of Work, but is a natural extension of the traditional continuous optimization problem. A full treatment of this subtask is given in Discrete Posynomial Programming With Applications To Spacecraft Protective Structures Design

__Optimization__, by R. A. Mog. The goal of this subtask is to develop a theory for solving nonlinear programming problems that may be stated in standard posynomial form under the guidelines of prototype geometric programming, but with discrete constraints on the primal independent variables. The main development thrust is in the direction of dual program solution methods, although primal solution techniques are also developed. Dual method solution approaches will depend on problem degree of difficulty, but for problems with nontrivial degrees of difficulty, partial invariance and direct search techniques are investigated for their utility. Because signomial (polynomial with undetermined coefficients and real exponents) programming methods do not result in global optimization, extensions of discrete techniques to signomial and reversed inequality constraint problems are considered secondary to this effort.

Another goal of this subtask is to demonstrate applications of the developed discrete posynomial programming methodologies. These applications include challenges in the field of spacecraft protective structures design optimization and emphasize missions that are relevant to Space Station Freedom and space debris/meteoroid environments. Specific hypervelocity impact predictor models include those of Nysmith, Wilkinson, and Burch.

__Subtask Approach__

After a brief review of posynomial programming in Section 6.2, two primal methods for solving the discrete posynomial program are introduced in Section 6.3. The methods are numerical in nature and easy to apply to practical problems. However, no global or analytic information is guaranteed in their application. Therefore, in Section 6.4, a dual method is developed which provides the global optimal solution to the discrete posynomial program. Finally, three case studies which illustrate the capabilities of the primal and dual methods in this field are presented in Section 6.5.

## 6.2 Review of Posynomial Programming

In a search performed at the Redstone Scientific Information Center (RSIC) to determine documents with the keywords "Geometric Programming" in either the title or abstract, a total of 92 listings were found. Of the 92 listings, approximately 34 were theoretical and 58 applied. Most of the theoretical listings dealt with algorithmic improvements, code comparisons, tutorial papers explaining the method, and theses on specific areas of geometric programming developments. Of the 58 applied listings, almost all involved structural design applications.[1,3,17,37,58,70,79,92,96,107-112,126,133,145,151,152,155] Other applied areas included economic,[67] communications,[69] and traffic flow problems.[59] Perhaps most surprising is that 27 of the 92 listings were written after 1980. Since geometric programming was formalized in 1967, this points to a possible resurgence in the method's use.

The most interesting conclusion from the article survey is that the relatively large number of application papers conflicts with the dismissals of many textbook authors concerning the utility of geometric programming. With this many listings, it is clear that some scientists are finding great uses for GP.

Based on the relatively large number of applied geometric programming listings in the article survey, it is apparent that GP possesses a fairly high utility, particularly in the area of structural design. Because GP is the only nonlinear programming (NLP) technique which offers the guarantee of a globally optimal solution for certain nonconvex problems, it should be considered more widely in practice. Additionally, for zero degree of difficulty problems, GP can provide an analytic optimal solution for the objective function and independent variables. This attribute provides greater insight for the system designer than that obtainable by other NLP techniques. Finally, the values of the dual variables may provide very crucial design information alone in terms of the physical parameters of the problem at hand.

## 6.3 Discrete Posynomial Programming Using Primal Methods

### Introduction

As explained earlier, geometric programming includes both posynomial and signomial programs and dual and primal approaches. In this section, a primal method for solving discrete posynomial programs is developed. The technique employed is an exterior penalty function method[10,53] supported by two alternate search techniques: a random/exhaustive search[53] and a Hooke and Jeeves pattern search.[10,53] The primal methodology computer code is called POLYPRIME.FOR and is given in Appendix A.

### Penalty Function Development

Penalty function techniques are widely used numerical optimization methods which convert constrained optimization problems into unconstrained ones with appropriate penalties for not satisfying the constraints.[10,53] Two general classes of penalty functions exist: exterior and barrier functions. Exterior penalty function methods generally begin with points outside the feasible solution space and progressively drive the solutions into the feasible region. Barrier function methods require feasible initial points in setting up blockades along the constraint surfaces.

For the problem of solving the primal formulation of discrete posynomial programs, an exterior penalty function technique is chosen to relieve the analyst of the burden of specifying a feasible initial point. This requirement could be particularly difficult when combinations of continuous and discrete constraints are involved. The primal problem is specified as

$$\min f = \sum_{i=1}^{n} c_i \prod_{j=1}^{k} x_j^{a_{ij}} \qquad [124]$$

$$s.t. \qquad g_l = \sum_{i=1}^{m_l} c_{il} \prod_{j=1}^{k} x_j^{a_{ijl}} \le K_l, \qquad l = 1,2,\ldots,p \qquad [125]$$

and the additional discrete constraints

$$x_j = r_j n_j, \qquad j = 1,2,\ldots,q \le k \tag{126}$$

One choice for an unconstrained penalty function is

$$\min\phi = \sum_{i=1}^{n} c_i \prod_{j=1}^{k} x_j^{a_{ij}} + \sum_{l=1}^{p} \delta_l A_l \left( \sum_{i=1}^{m_l} c_{il} \prod_{j=1}^{k} x_j^{a_{ijl}} - K_l \right)^2 + \sum_{j=1}^{q} \Delta_j \alpha_j \left| \frac{x_j}{r_j} - \left[ \frac{x_j}{r_j} \right]_{n.i.} \right|^{1/2} \tag{127}$$

where

$$\delta_l = 1, \qquad g_l - K_l > 0 \tag{128}$$

$$\delta_l = 0, \qquad g_l - K_l < 0, \qquad l = 1,2,\ldots,p \tag{129}$$

$$\Delta_j = 1, \qquad j = 1,2,\ldots,q. \tag{130}$$

Here, it is assumed, without loss of generality, that the $x_j$'s may be reordered such that the first q of them are those requiring discrete solutions. Also, the discrete penalty term has an exponent of 1/2 to require a stricter measure of convergence, since

$$\left| \frac{x_j}{r_j} - \left[ \frac{x_j}{r_j} \right]_{n.i.} \right| \le 0.5 < 1 \tag{131}$$

In POLYPRIME.FOR, the accelerating factors begin at 1.0 and progressively are multiplied by 10 until convergence is reached, i.e.

$$\sum_{l=1}^{p} \delta_l A_l \left( \sum_{i=1}^{m_l} c_{il} \prod_{j=1}^{k} x_j^{a_{ijl}} - K_l \right)^2 + \sum_{j=1}^{q} \Delta_j \alpha_j \left| \frac{x_j}{r_j} - \left[ \frac{x_j}{r_j} \right]_{n.i.} \right|^{1/2} \le \varepsilon = 0.001 \tag{132}$$

Note that this method handles mixed discrete problems as well as continuous and purely discrete problems. Furthermore, note that although we are strictly concerned with posynomial programming problems, this penalty function approach is equally valid for generalized polynomials or signomials and signomial constraints. Additionally, Type I, II, or III inequality constraints are valid. However, the constraints must be converted to Type I, less than or equal to constraints, for implementation in POLYPRIME.FOR. Similarly, the constraint right hand side values, $K_l$, may take any real value. This is a more generalized form than that allowed for prototype posynomial programming where the $K_l$'s must be equal to 1 for all constraints. Finally, note that if convergence of continuous equality constraints is difficult to achieve using this formulation, it may be easily modified by adding a penalty term for equality constraints separate from inequality constraints. Note, however, that continuous equality constraints combined with discrete variable constraints could easily result in no feasible solution situations.

Now, once the unconstrained penalty objective function is established, a method to minimize it must be found. Two approaches using search techniques are discussed in the next two sections, followed by a comparison of the methods.

### Random/Exhaustive Search Subtechnique

A random search technique[53] is analogous to throwing darts at a dartboard with no adaptation or learning between throws. (There do exist adaptive random search techniques, but these won't be considered here.) Although random search techniques may appear unsophisticated due to their brute force nature, they are particularly useful in establishing optima of highly nonlinear and multimodal functions. Since discrete nonlinear optimization problems tend to add a degree of this type of complexity, it would appear that random search techniques would prove fruitful.

The number of search points for a pure random search is given by Gottfried as

$$m_h = \frac{-1}{F_h^k} \ln(1 - P_h) \qquad [133]$$

The search space is defined by specifying an interval of interest for each of the independent variables. The main drawback for employing a random search technique in a discrete optimization problem using penalty functions is the severity of the convergence criteria. Unless the random draw is extremely fortuitous, the convergence criteria will not be met, particularly for sparsely populated discrete feasible regions. For this reason, an exhaustive search option is automatically called in POLYPRIME.FOR when the number of discrete feasible points (as specified by the search space) is less than the number of search points given by $m_h$ above. On the other hand, if the search region is dense with discrete points, as compared with the pres-pecified number of random search points, then random search proceeds normally.

## Hooke and Jeeves Pattern Search Subtechnique

The Hooke and Jeeves pattern search[10,53] is a more methodical unconstrained search technique, which requires an initial point, but no variable search intervals. The technique begins with exploratory moves to establish a base point. These moves are followed by pattern moves through successive base points. Convergence requirements are more easily met for discrete problems using this method.

## Comparison of Subtechniques

The Hooke and Jeeves pattern search technique is more methodical and generally converges faster than the random/exhaustive search technique. Furthermore, it requires only an initial point rather than an interval of investigation. On the other hand, the Hooke and Jeeves method is generally fairly sensitive to the initial search point and is less likely to find global optima for multimodal functions. Furthermore, although the random/exhaustive search technique may overly restrict the region of interest for a variable, this condition can easily be diagnosed when the optimal solution is found at an interval endpoint.

One excellent approach to combining the two methods' strong points is to use the random search technique in solving the corresponding continuous optimization problem, and then use that solution as the initial search point for the Hooke and Jeeves subtechnique for the discrete problem. Another interesting study would be to compare the two methods from a time-effectiveness standpoint.

## 6.4 Discrete Posynomial Programming Using Dual Methods

### Introduction

The use of penalty function techniques with random/exhaustive and Hooke and Jeeves search subtechniques may provide rapid convergence for discrete posynomial (and signomial) programs. However, numerical instabilities may occur in the penalty function acceleration parameters due to ridges in the penalty function. Additionally, global optimal solutions are not generally achieved, particularly when exhaustive search is not performed. Finally, little analytical information is gained for sensitivity analysis when numerical methods are applied. The restriction to posynomial programs does not lead to any significant advantages over signomial programs using the primal approach. Indeed, the method is perfectly valid for generalized polynomials or signomials and general constraints.

In this section, dual methods are applied to discrete posynomial programs. The main advantages to these approaches are the guarantee of a global optimal solution and the analytic information gained during the process. The main disadvantages are that some derivation is required and that many nondegenerate continuous programs result in discrete programs with high degrees of difficulty.

## General Development and Degree of Difficulty

The prototype discrete primal program is given as

$$\min f = \sum_{i=1}^{n} c_i \prod_{j=1}^{k} x_j^{a_{ij}} \qquad [134]$$

$$s.t. \qquad g_l = \sum_{i=1}^{m_l} c_{il} \prod_{j=1}^{k} x_j^{a_{ijl}} \le 1, \qquad l = 1,2,\ldots,p \qquad [135]$$

$$x_j = r_j n_j, \qquad j = 1,2,\ldots,q \le k \qquad [136]$$

where $c_i$, $x_j$, $c_{il}$, and $r_j$ are positive valued for all i, j, and l, and $n_j$ is a positive integer for all j. Note that this primal program is quite similar to that given in Section 6.3, with the only difference being the replacement of the $K_l$'s with the value 1. Additionally, in the case of the dual methods, the positivity restrictions are strictly adhered to. Recall that the primal method is equally valid for all real coefficients, general right hand side values, and Type I, II, and III constraints.

The first and most obvious fact to note about this problem is that the continuous optimal objective function value (obtained by not considering the discrete constraints) is always a lower bound for the discrete objective function value. This is easily seen by contradiction, since, for any set of $n_j$'s established in a discrete optimal solution, the independent variables may always take the values $n_j r_j$ for any equivalent continuous problem.

The second item to notice is that the discrete equality constraints may always be written as pairs of prototype posynomial constraints, i.e.

$$\frac{x_j}{r_j n_j} \le 1 \wedge \frac{r_j n_j}{x_j} \le 1 \qquad [137]$$

Based on this observation, we may establish a first theorem for discrete posynomial programming.

**Theorem 1.** Suppose $\{n_1, n_2, \ldots, n_q\}$ are positive integers. Then, provided the dual program is consistent and the feasible discrete solution space is nonempty, the mixed discrete posynomial programming solution is globally optimal. Furthermore, the discrete dual program degree of difficulty is 2q greater than that for the continuous dual program.

**Proof:**

By the preceding observation, the discrete primal problem can always be formulated as a prototype posynomial program, which has been shown to be globally optimal (provided feasibility/consistency relationships hold) under the dual program

$$\max v(\delta) = \prod_{i=1}^{n}\left(\frac{c_i}{\delta_i}\right)^{\delta_i}\left(\prod_{j=1}^{q}(r_j n_j)^{\delta_{2j}-\delta_{1j}}\right)\left(\prod_{l=1}^{p}\mu_l^{\mu_l}\left(\prod_{j=1}^{m_l}\left(\frac{c_{jl}}{\delta'_{jl}}\right)^{\delta'_{jl}}\right)\right) \qquad [138]$$

with

$$\sum_{i=1}^{n}\delta_i a_{ih} - (\delta_{2h} - \delta_{1h}) + \sum_{l=1}^{p}\left(\sum_{j=1}^{m_l}\delta'_{jl} a_{jhl}\right) = 0 \qquad h = 1, 2, \ldots, q \qquad [139]$$

$$\sum_{i=1}^{n}\delta_i a_{ih} + \sum_{l=1}^{p}\left(\sum_{j=1}^{m_l}\delta'_{jl} a_{jhl}\right) = 0 \qquad h = q+1, q+2, \ldots, k \qquad [140]$$

$$\sum_{i=1}^{n}\delta_i = 1 \qquad [141]$$

$$\mu_l = \sum_{j=1}^{m_l}\delta'_{jl} \qquad l = 1, 2, \ldots, p \qquad [142]$$

Thus, there are k+p+1 equations in $n+2q+p+m_1+m_2+\ldots+m_p$ unknowns. The discrete dual degree of difficulty is given by

$$DOD = n - k - 1 + 2q + \sum_{l=1}^{p} m_l \qquad [143]$$

which is 2q greater than that for the continuous dual problem given by equation [32]. Thus, the theorem is proved.

Note that this theorem is not particularly useful without some idea of the nature of the desired integer values $n_j$. The following theorem is useful as an expression for the basic dual variables.

**Theorem 2.** For the discrete dual program, the basic dual variables may be written in terms of the nonbasic dual variables, the $n_j$'s, and the nondiscrete primal variables as

$$\delta_m = c_m \left[ \frac{\prod_{j=q+1}^{k} x_j^{a_{mj}}}{\prod_{\substack{i=1 \\ i \neq m}}^{n} \left( \frac{c_i}{\delta_i} \right)^{\delta_i} \prod_{j=1}^{q} (r_j n_j)^{\delta_{2j} - \delta_{1j} - a_{mj}} \prod_{l=1}^{p} \left( \sum_{j=1}^{m_l} \delta'_{jl} \right)^{\left( \sum_{j=1}^{m_l} \delta'_{jl} \right)} \left( \prod_{j=1}^{m_l} \left( \frac{c_{jl}}{\delta'_{jl}} \right)^{\delta'_{jl}} \right)} \right]^{\frac{1}{\sum_{\substack{i=1 \\ i \neq m}}^{n} \delta_i}} \qquad [144]$$

for m=1,2,...,n.

**Proof:**

The global optimal solution is given at the equality of the arithmetic and geometric means, i.e. when

$$c_i \prod_{j=1}^{k} x_j^{a_{ij}} = \delta_i v(\delta), \qquad i = 1, 2, \ldots, n. \qquad [145]$$

This is often referred to as the dual-to-primal transformation. Thus, we may write

$$\left( \frac{c_i}{\delta_i} \right) \prod_{j=1}^{k} x_j^{a_{ij}} = \prod_{i=1}^{n} \left( \frac{c_i}{\delta_i} \right)^{\delta_i} \prod_{j=1}^{q} (r_j n_j)^{\delta_{2j} - \delta_{1j}} \prod_{l=1}^{p} \mu_l^{\mu_l} \left( \prod_{j=1}^{m_l} \left( \frac{c_{jl}}{\delta'_{jl}} \right)^{\delta'_{jl}} \right), \qquad i = 1, 2, \ldots, n \qquad [146]$$

or

$$\prod_{j=1}^{k} x_j^{a_{mj}} = \left( \frac{c_m}{\delta_m} \right)^{\delta_m - 1} \prod_{\substack{i=1 \\ i \neq m}}^{n} \left( \frac{c_i}{\delta_i} \right)^{\delta_i} \prod_{j=1}^{q} (r_j n_j)^{\delta_{2j} - \delta_{1j}} \prod_{l=1}^{p} \mu_l^{\mu_l} \left( \prod_{j=1}^{m_l} \left( \frac{c_{jl}}{\delta'_{jl}} \right)^{\delta'_{jl}} \right), \qquad m = 1, 2, \ldots, n \qquad [147]$$

But

$$x_j = r_j n_j, \qquad j = 1,2,\ldots,q \Rightarrow \tag{148}$$

$$\left(\frac{\delta_m}{c_m}\right)^{\delta_m - 1} = \frac{\prod_{\substack{i=1\\ i\neq m}}^{n}\left(\frac{c_i}{\delta_i}\right)^{\delta_i}\prod_{j=1}^{q}(r_j n_j)^{\delta_{2j}-\delta_{1j}-a_{mj}}\prod_{l=1}^{p}\mu_l^{\mu_l}\left(\prod_{j=1}^{m_l}\left(\frac{c_{jl}}{\delta'_{jl}}\right)^{\delta'_{jl}}\right)}{\prod_{j=q+1}^{k} x_j^{a_{mj}}}, \qquad m = 1,2,\ldots,n \tag{149}$$

or

$$\left(\frac{\delta_m}{c_m}\right)^{1-\delta_m} = \frac{\prod_{j=q+1}^{k} x_j^{a_{mj}}}{\prod_{\substack{i=1\\ i\neq m}}^{n}\left(\frac{c_i}{\delta_i}\right)^{\delta_i}\prod_{j=1}^{q}(r_j n_j)^{\delta_{2j}-\delta_{1j}-a_{mj}}\prod_{l=1}^{p}\mu_l^{\mu_l}\left(\prod_{j=1}^{m_l}\left(\frac{c_{jl}}{\delta'_{jl}}\right)^{\delta'_{jl}}\right)}, \qquad m = 1,2,\ldots,n \tag{150}$$

The desired result follows by substituting

$$\sum_{\substack{i=1\\ i\neq m}}^{n} \delta_i = 1 - \delta_m, \qquad m = 1,2,\ldots,n \tag{151}$$

in the exponent. Use of this result combined with the dual linear equality constraints may help define further avenues of solution.

## "Posyseparable" Programs and Partial Invariance

In this section, the term "posyseparable" is introduced, followed by techniques which may simplify the solution of the dual discrete posynomial program, including partial invariance.

**Definition:** A posynomial function, f, is called "posyseparable" if each of its independent variables may be isolated at least once, i.e., if it may be written

$$f = \sum_{i=1}^{k} c_{is} x_i^{a_i} + \sum_{i=1}^{n-k} c_i \prod_{j=1}^{k} x_j^{a_{ij}} \tag{152}$$

where the $x_i$'s, $c_{is}$'s, and $c_i$'s are positive valued. Note that the term posyseparable, as applied to posynomials, is less restrictive than the term separable (see separable programming) where each independent variable is completely isolated in a functional sense from the remaining independent variables. Many posynomials display the posyseparable property. Although this property is not a necessary condition for existence or uniqueness of global optimal solutions for discrete posynomial programs, it does allow for a straightforward dual-to-primal variable conversion, which is often a major drawback to using dual methods.

**Theorem 3.** If the objective function in a consistent and feasible discrete posynomial program is posyseparable, then the basic dual variables may be written in terms of any of the q discrete variables as

$$\delta_{ms} = c_{ms}\Bigg[\left(\frac{c_{qs}}{\delta_{qs}}\right)^{\delta_{qs}-1} \prod_{\substack{i=1\\ i\neq m,q}}^{k}\left(\frac{c_{is}}{\delta_{is}}\right)^{\delta_{is}} \prod_{i=1}^{n-k}\left(\frac{c_i}{\delta_i}\right)^{\delta_i} (r_q n_q)^{\delta_{2q}-\delta_{1q}-a_q}$$

$$\cdot \prod_{j=1}^{q-1}\left(\frac{\delta_{js}}{\delta_{qs}}\frac{c_{qs}}{c_{js}}(r_q n_q)^{a_q}\right)^{\frac{\delta_{2j}-\delta_{1j}}{a_j}} \prod_{l=1}^{p}\left(\sum_{j=1}^{m_l}\delta'_{jl}\right)^{\left(\sum_{j=1}^{m_l}\delta'_{jl}\right)}\left(\prod_{j=1}^{m_l}\left(\frac{c_{jl}}{\delta'_{jl}}\right)^{\delta'_{jl}}\right)\Bigg]^{\frac{1}{1-\sum_{i=1}^{n-k}\delta_i-\sum_{\substack{i=1\\ i\neq m}}^{k}\delta_{is}}}, m = 1,2,\ldots,k \quad [153]$$

where

$$\delta_{2h}-\delta_{1h} = a_h\delta_{hs} + \sum_{i=1}^{n-k} a_{ih}\delta_i + \sum_{l=1}^{p}\left(\sum_{j=1}^{m_l}\delta'_{jl}a_{jhl}\right), \qquad h = 1,2,\ldots,q \qquad [154]$$

**Proof:** The dual program is

$$\max v(\delta) = \prod_{i=1}^{k}\left(\frac{c_{is}}{\delta_{is}}\right)^{\delta_{is}} \prod_{i=1}^{n-k}\left(\frac{c_i}{\delta_i}\right)^{\delta_i}\left(\prod_{j=1}^{q}(r_j n_j)^{\delta_{2j}-\delta_{1j}}\right)\left(\prod_{l=1}^{p}\mu_l^{\mu_l}\left(\prod_{j=1}^{m_l}\left(\frac{c_{jl}}{\delta'_{jl}}\right)^{\delta'_{jl}}\right)\right) \qquad [155]$$

$$\delta_{hs}a_h + \sum_{i=1}^{n-k}\delta_i a_{ih} - (\delta_{2h}-\delta_{1h}) + \sum_{l=1}^{p}\left(\sum_{j=1}^{m_l}\delta'_{jl}a_{jhl}\right) = 0 \qquad h = 1,2,\ldots,q \qquad [156]$$

$$\delta_{hs}a_h + \sum_{i=1}^{n-k}\delta_i a_{ih} + \sum_{l=1}^{p}\left(\sum_{j=1}^{m_l}\delta'_{jl}a_{jhl}\right) = 0 \qquad h = q+1,q+2,\ldots,k \qquad [157]$$

$$\sum_{i=1}^{k}\delta_{is} + \sum_{i=1}^{n-k}\delta_i = 1 \qquad [158]$$

$$\mu_l = \sum_{j=1}^{m_l} \delta'_{jl} \qquad l = 1, 2, \ldots, p \tag{159}$$

Since f is posyseparable,

$$c_{ms} x_m^{a_m} = \delta_{ms} v(\delta), \qquad m = 1, 2, \ldots, k \tag{160}$$

Also,

$$x_j = r_j n_j, \qquad j = 1, 2, \ldots, q \tag{161}$$

Therefore,

$$\left(\frac{c_{ms}}{\delta_{ms}}\right) x_m^{a_m} = \prod_{i=1}^{k} \left(\frac{c_{is}}{\delta_{is}}\right)^{\delta_{is}} \prod_{i=1}^{n-k} \left(\frac{c_i}{\delta_i}\right)^{\delta_i} \left(\prod_{j=1}^{q} (r_j n_j)^{\delta_{2j} - \delta_{1j}}\right) \left(\prod_{l=1}^{p} \mu_l^{\mu_l} \left(\prod_{j=1}^{m_l} \left(\frac{c_{jl}}{\delta'_{jl}}\right)^{\delta'_{jl}}\right)\right) \tag{162}$$

But

$$c_{ms} \frac{x_m^{a_m}}{\delta_{ms}} = v(\delta) = c_{qs} \frac{x_q^{a_q}}{\delta_{qs}} \tag{163}$$

or

$$x_m^{a_m} = \frac{\delta_{ms} c_{qs}}{\delta_{qs} c_{ms}} x_q^{a_q} \tag{164}$$

Therefore,

$$\left(\frac{\delta_{ms}}{c_{ms}}\right)^{\delta_{ms}} \frac{c_{qs}}{\delta_{qs}} x_q^{a_q} = \prod_{\substack{i=1 \\ i \neq m}}^{k} \left(\frac{c_{is}}{\delta_{is}}\right)^{\delta_{is}} \prod_{i=1}^{n-k} \left(\frac{c_i}{\delta_i}\right)^{\delta_i} \left(\prod_{j=1}^{q} (r_j n_j)^{\delta_{2j} - \delta_{1j}}\right)$$

$$\bullet \left(\prod_{l=1}^{p} \left(\sum_{j=1}^{m_l} \delta'_{jl}\right)^{\sum_{j=1}^{m_l} \delta'_{jl}} \left(\prod_{j=1}^{m_l} \left(\frac{c_{jl}}{\delta'_{jl}}\right)^{\delta'_{jl}}\right)\right), \qquad m = 1, 2, \ldots, k \tag{165}$$

Since

$$x_q = r_q n_q \tag{166}$$

$$\delta_{ms} = c_{ms}[\left(\frac{C_{qs}}{\delta_{qs}}\right)^{\delta_{qs}-1} \prod_{\substack{i=1\\ i\neq m,q}}^{k} \left(\frac{C_{is}}{\delta_{is}}\right)^{\delta_{is}} \prod_{i=1}^{n-k}\left(\frac{C_i}{\delta_i}\right)^{\delta_i} (r_q n_q)^{\delta_{2q}-\delta_{1q}-a_q}$$

$$\bullet \prod_{j=1}^{q-1}(r_j n_j)^{\delta_{2j}-\delta_{1j}} \prod_{l=1}^{p}\left(\sum_{j=1}^{m_l}\delta'_{jl}\right)^{\sum_{j=1}^{m_l}\delta'_{jl}} (\prod_{j=1}^{m_l}(\frac{C_{jl}}{\delta'_{jl}})^{\delta'_{jl}})]^{\frac{1}{1-\sum_{i=1}^{n-k}\delta_i-\sum_{\substack{i=1\\ i\neq m}}^{k}\delta_{is}}}, \qquad m = 1,2,\ldots,k \qquad [167]$$

But

$$r_j n_j = \left(\frac{\delta_{js}C_{qs}}{\delta_{qs}C_{js}}(r_q n_q)^{a_q}\right)^{\frac{1}{a_j}} \Rightarrow \qquad [168]$$

$$\delta_{ms} = c_{ms}[\left(\frac{C_{qs}}{\delta_{qs}}\right)^{\delta_{qs}-1} \prod_{\substack{i=1\\ i\neq m,q}}^{k} \left(\frac{C_{is}}{\delta_{is}}\right)^{\delta_{is}} \prod_{i=1}^{n-k}\left(\frac{C_i}{\delta_i}\right)^{\delta_i} (r_q n_q)^{\delta_{2q}-\delta_{1q}-a_q}$$

$$\bullet \prod_{j=1}^{q-1}\left(\frac{\delta_{js}}{\delta_{qs}}\frac{C_{qs}}{C_{js}}(r_q n_q)^{a_q}\right)^{\frac{\delta_{2j}-\delta_{1j}}{a_j}} \prod_{l=1}^{p}\left(\sum_{j=1}^{m_l}\delta'_{jl}\right)^{\left(\sum_{j=1}^{m_l}\delta'_{jl}\right)} \left(\prod_{j=1}^{m_l}\left(\frac{C_{jl}}{\delta'_{jl}}\right)^{\delta'_{jl}}\right)]^{\frac{1}{1-\sum_{i=1}^{n-k}\delta_i-\sum_{\substack{i=1\\ i\neq m}}^{k}\delta_{is}}}, m = 1,2,\ldots,k \qquad [169]$$

For

$$m \leq q-1 \qquad [170]$$

we may rewrite

$$\delta_{ms} = 1 - \sum_{i=1}^{n-k}\delta_i - \sum_{\substack{i=1\\ i\neq m}}^{k}\delta_{is} \qquad [171]$$

and the theorem is proved. Combining this result with the linear dual equations allows one to use partial invariance to solve for basic dual variables in terms of at most one $n_j$. Then, a series of differences in terms of the discrete and continuous dual objective functions may be minimized as:

$$v_d(\delta) - v_c(\delta) = \frac{c_{ms} x_m^{a_m}}{\delta_{ms}} - v_c(\delta) \qquad [172]$$

Setting this difference to zero, we may write

$$c_{ms} x_m^{a_m} - \delta_{ms} v_c(\delta) = \frac{\delta_{ms} c_{qs}}{\delta_{qs}} (r_q n_q)^{a_q} - \delta_{ms} v_c(\delta) \qquad [173]$$

$$= \delta_{ms} \left( \frac{c_{qs}}{\delta_{qs}} (r_q n_q)^{a_q} - v_c(\delta) \right) \qquad [174]$$

which is a function of $n_q$ and the dual variables only. In general, partial invariance should be used to solve the basic dual variables in terms of the dominant dual variables if one has that knowledge for the problem at hand. A good starting point is to solve for the dual variables corresponding to the discrete (primal) constraints in terms of the basic dual variables evaluated in the neighborhood of their optimal continuous solutions.

A fair amount of derivation has indicated possible directions a dual approach may take in a discrete posynomial program. The concept of posyseparability has been introduced to ease the dual-to-primal transformation. The discrete dual program has been shown to have a degree of difficulty 2q greater than the corresponding continuous dual program. A number of relatively simple examples have been used to illustrate various facets of the dual approach. Primal and dual approaches have been compared. Specific numerical and computer methods used to support the dual approach are left for future development.

**Figures 6.4-1, 2, and 3** show the optimal discrete design values of minimum system mass per unit area, and optimal bumper and wall thicknesses vs projectile diameter, bumper/wall separation, and projectile velocity, respectively, for the Nysmith predictor. The discrete availability factor, $r_1$, is 1/64 inches. **Figure 6.4-1** reflects a constant meteoroid density. In **Figure 6.4-3**, the impact angle remains constant at 0 degrees (normal).

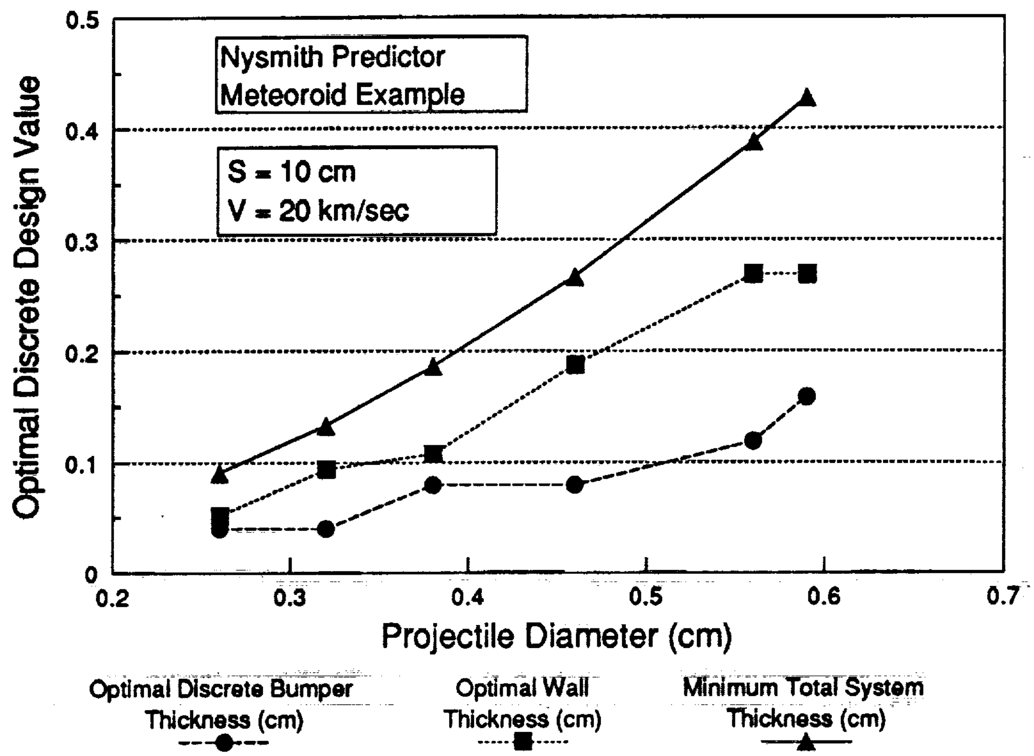



**Figure 6.4-1. Optimal Discrete Design Value vs Projectile Diameter for Nysmith Predictor**

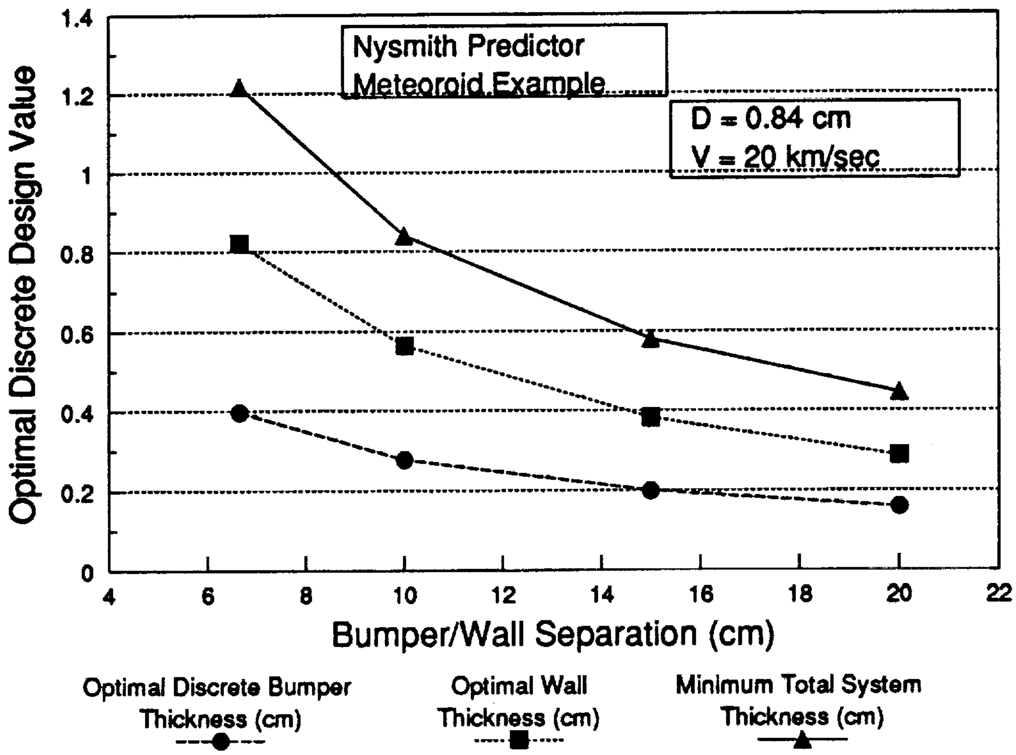



**Figure 6.4-2. Optimal Discrete Design Value vs Bumper/Wall Separation for Nysmith Predictor**

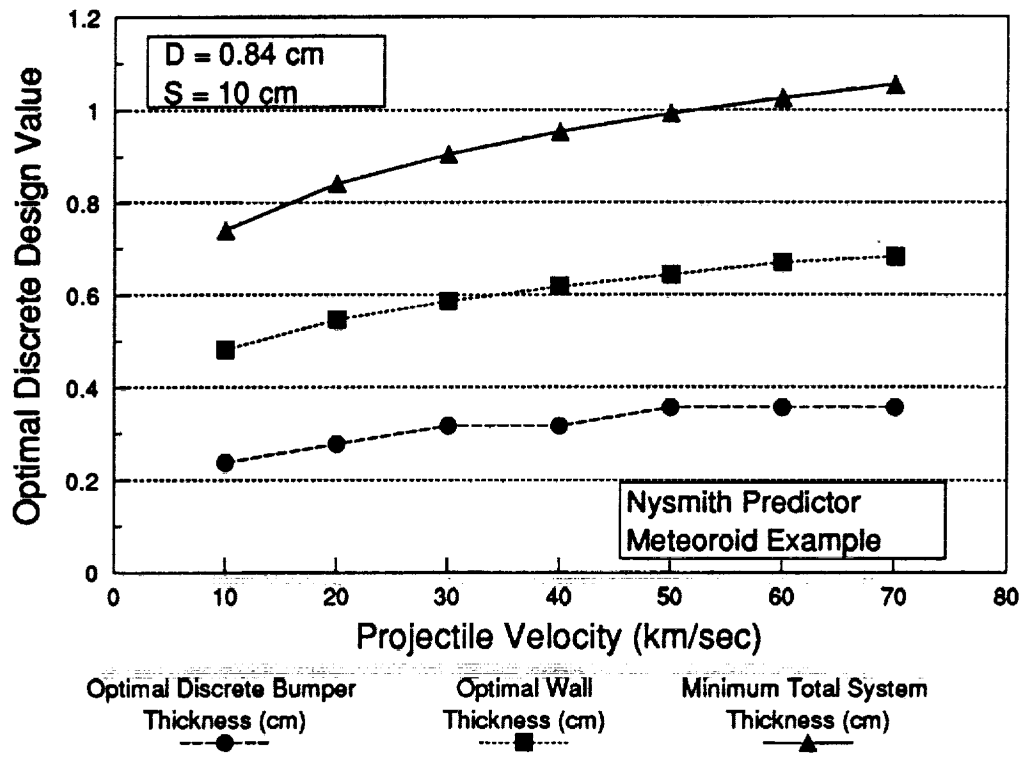



**Figure 6.4-3. Optimal Discrete Design Value vs Projectile Velocity for Nysmith Predictor**

**Figure 6.4-4** shows the sensitivity of minimum system mass per unit area to bumper thickness availability factor, $r_1$. The discrete and continuous objective functions are equal when the continuous bumper thickness is an integer multiple of the bumper thickness availability factor as shown in **Figure 6.4-5**. This occurs at numerous locations over the range considered. Note that when $r_1$ is small, the discrete bumper thickness is closer in value to the continuous bumper thickness. As $r_1$ grows, this incidence of equality naturally decreases while the deviations from the continuous minimum mass per unit area grow in value. Beyond the optimal continuous value of the bumper thickness, the objective function continues to grow indefinitely, because the availability factor is dominating the desired continuous solution.

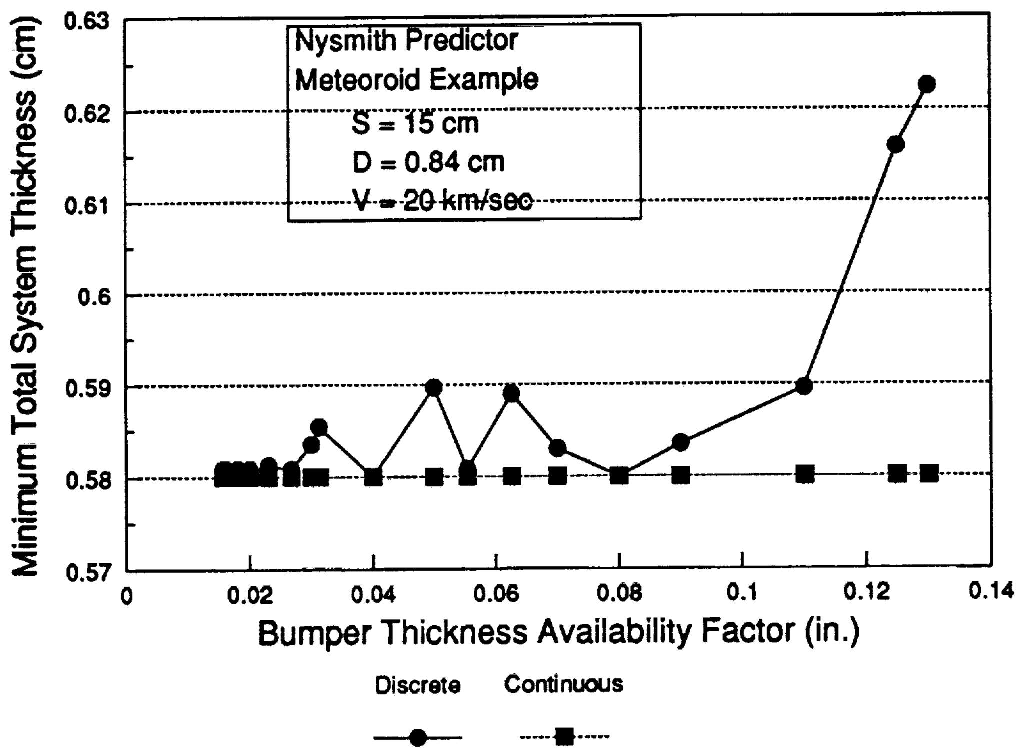



**Figure 6.4-4. Minimum Total System Mass Per Unit Area vs Bumper Thickness Availability Factor for the Nysmith Predictor**

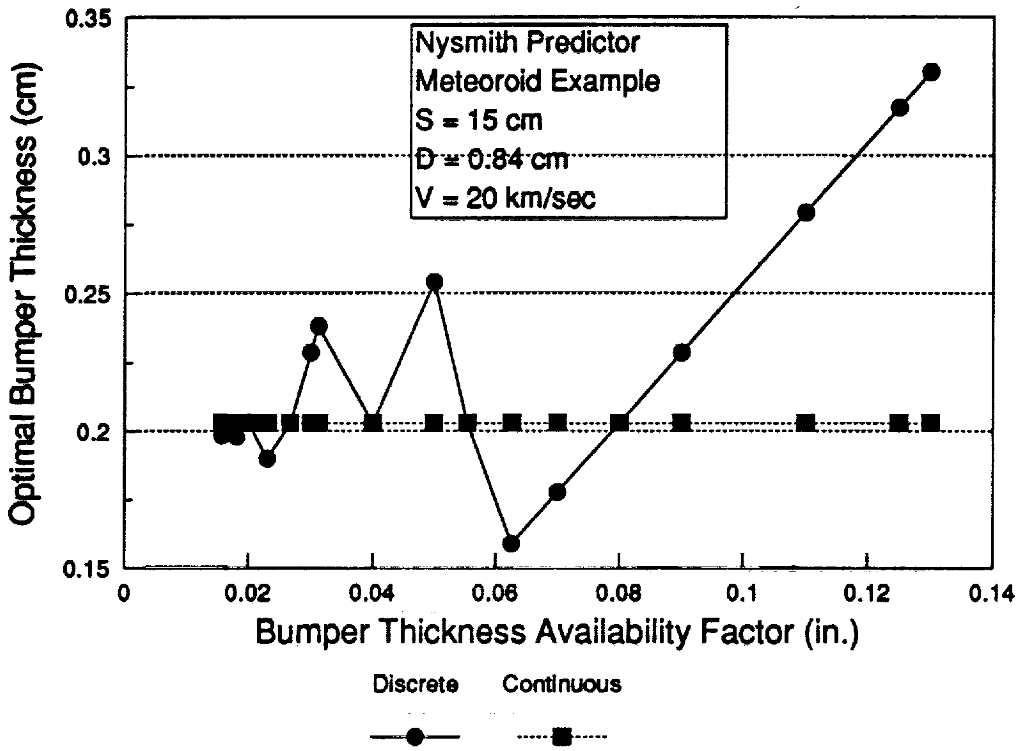



**Figure 6.4-5. Optimal Bumper Thickness vs Bumper Thickness Availability Factor for the Nysmith Predictor**

**Figures 6.4-6, 7, and 8** show the optimal discrete design values of minimum system mass per unit area, and optimal bumper and wall thicknesses vs projectile diameter, bumper/wall separation, and projectile velocity, respectively, for the Burch predictor. The bumper thickness availability factor is 1/64 in. **Figure 6.4-6** reflects a constant projectile density as given in equation [141]. In **Figure 6.4-8**, the impact angle remains constant at 0 degrees (normal).

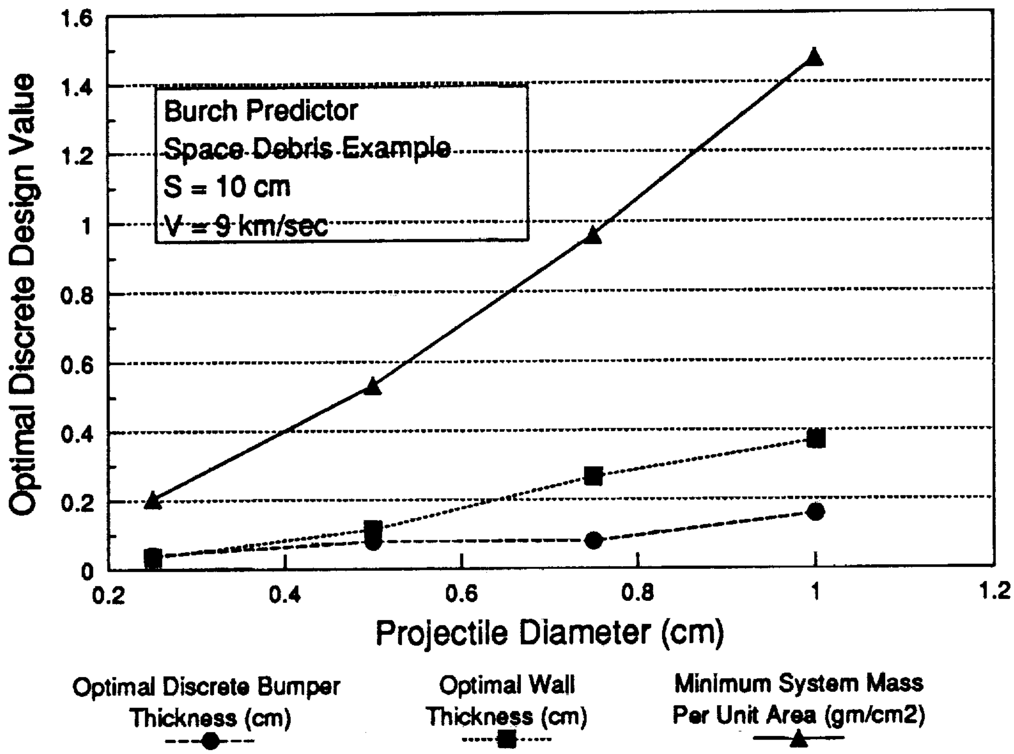



**Figure 6.4-6. Optimal Discrete Design Value vs Projectile Diameter for Burch Predictor**

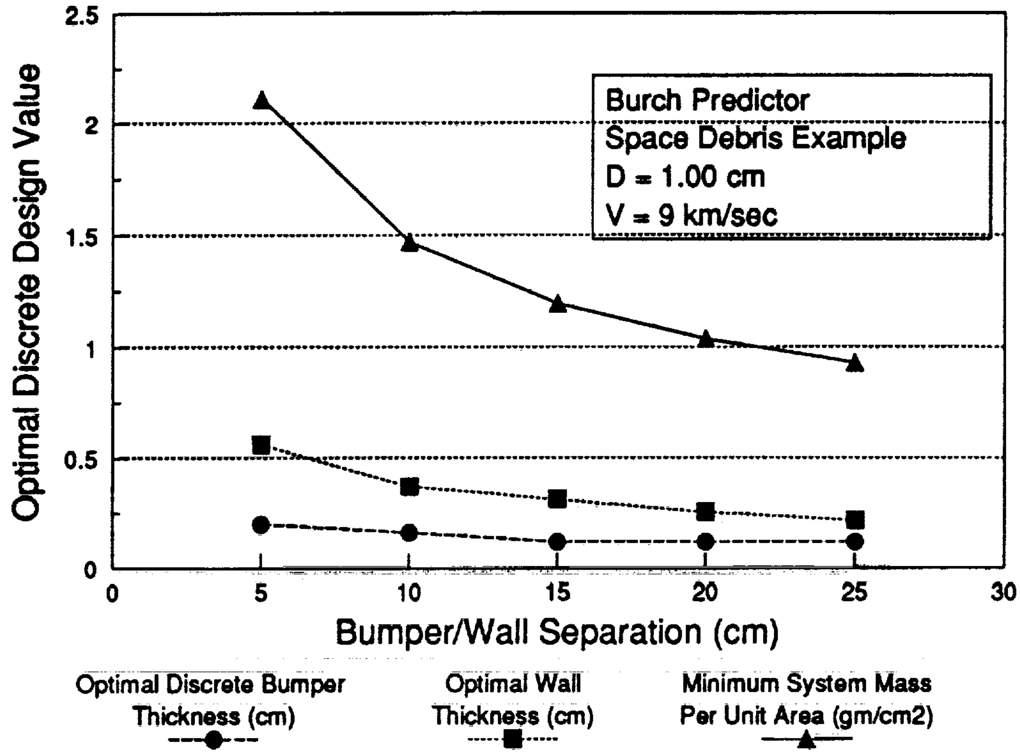



**Figure 6.4-7. Optimal Discrete Design Value vs Bumper/Wall Separation for Burch Predictor**

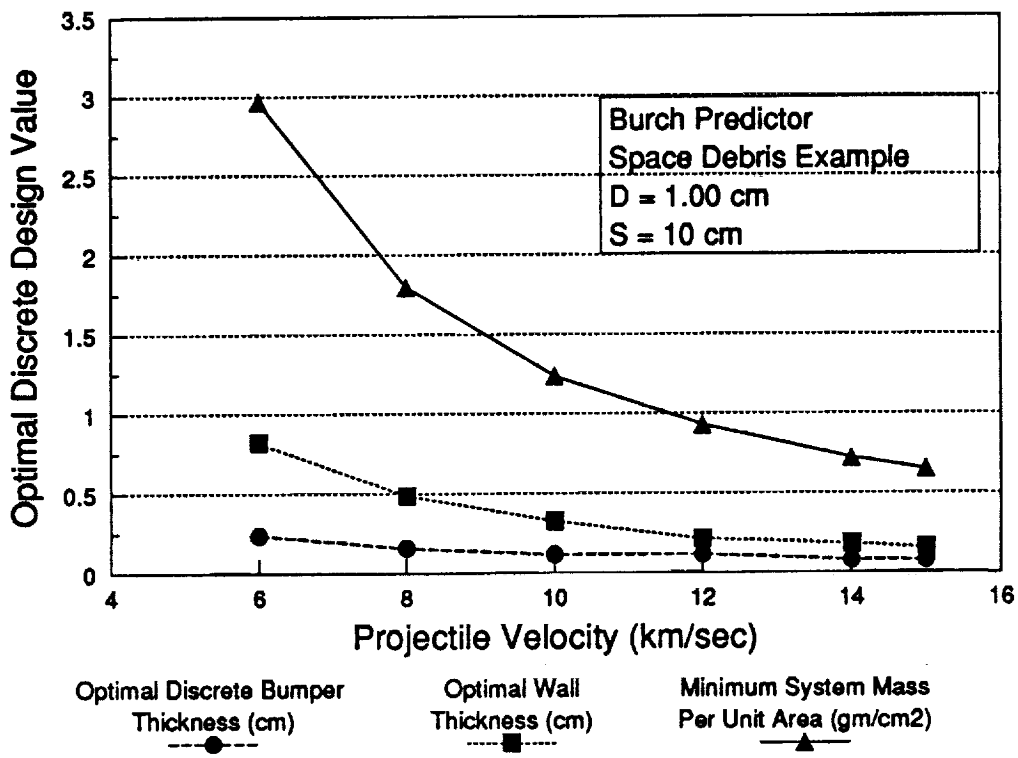



**Figure 6.4-8. Optimal Discrete Design Value vs Projectile Velocity for Burch Predictor**

**Figure 6.4-9** shows the sensitivity of minimum system mass per unit area to bumper thickness availability factor, $r_1$. The discrete and continuous objective functions are equal when the continuous bumper thickness is an integer multiple of the bumper thickness availability factor as shown in **Figure 6.4-10**. This occurs at numerous locations over the range considered. Note that when $r_1$ is small, the discrete bumper thickness is closer in value to the continuous bumper thickness. As $r_1$ grows, this incidence of equality naturally decreases while the deviations from the continuous minimum mass per unit area grow in value. Beyond the optimal continuous value of the bumper thickness, the objective function continues to grow indefinitely, because the availability factor is dominating the desired continuous solution.

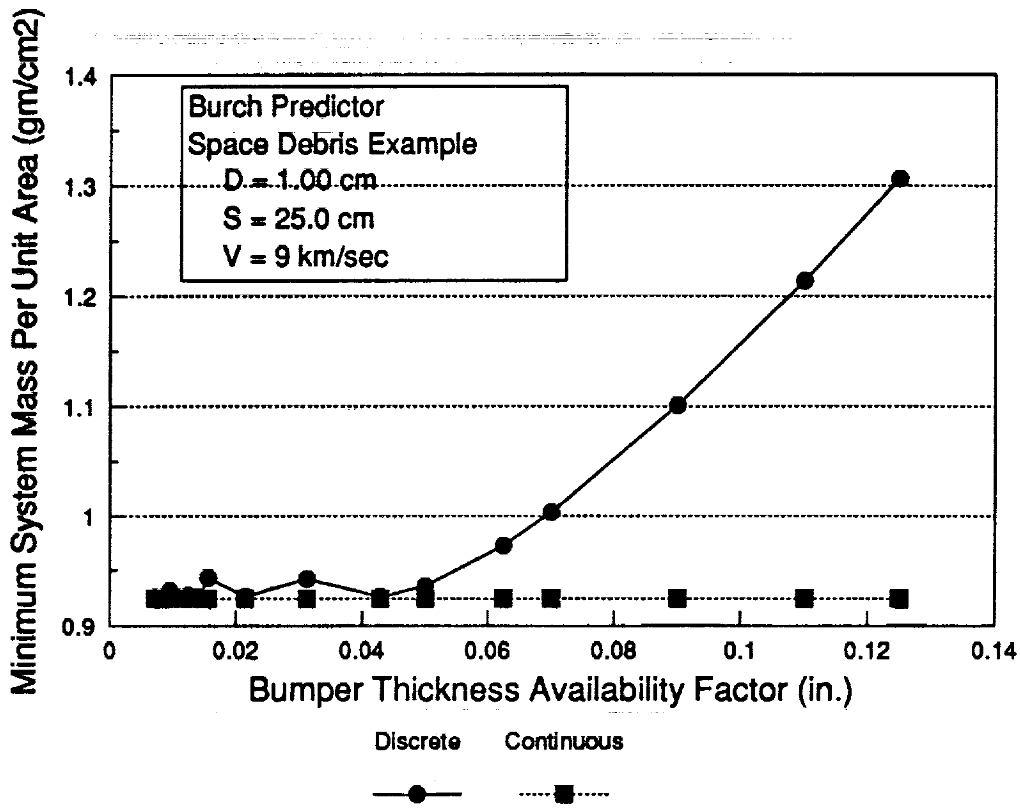



**Figure 6.4-9. Minimum System Mass Per Unit Area vs. Bumper Thickness Availability Factor for the Burch Predictor**

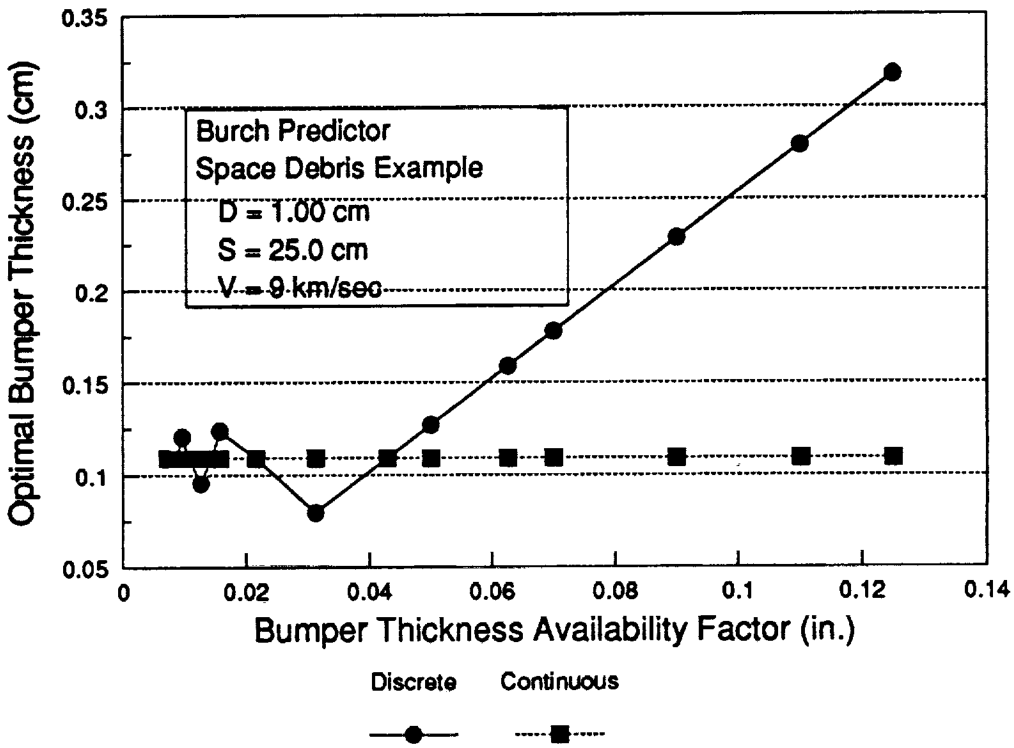



**Figure 6.4-10. Optimal Discrete and Continuous Bumper Thickness vs Bumper Thickness Availability Factor for the Burch Predictor**

The discrete Wilkinson predictor optimization algorithm is derived using the dual method.

**Theorem 4.** The combined discrete/continuous Wilkinson algorithm is given by

$$1.\quad c_1 = \frac{0.364 D^4 \rho_p^2 V \cos(\theta)}{L_2 S^2 \rho_1} \qquad [175]$$

$$2.\quad W_{0_c} = \frac{1.207 D^2 \rho_p}{S} \left( \frac{V \cos(\theta)}{L_2} \right)^{1/2} \qquad [176]$$

$$3.\quad t_{1_{0c}} = \frac{W_{0c}}{2\rho_1}, \qquad t_{2_{0c}} = \frac{W_{0c}}{2\rho_2} \qquad [177]$$

$$4.\quad If \frac{D \rho_p}{\rho_1 t_{1_{0c}}} > 1, \qquad n_1 = \left[ \frac{W_{0c}}{2\rho_1 r_1} \right]_{n.i.} \qquad [178]$$

Otherwise, go to step 10.

$$5. \quad t_{1_0} = r_1 n_1 \qquad [179]$$

$$6. \quad \delta_1 = \frac{r_1^2 \rho_1 n_1}{c_1 + r_1^2 \rho_1 n_1^2} \qquad [180]$$

$$7. \quad W_0 = \frac{\rho_1 t_{1_0}}{\delta_1}, \qquad t_{2_0} = \frac{W_0 - \rho_1 t_{1_0}}{\rho_2} \qquad [181]$$

$$8. \quad If \frac{D\rho_p}{\rho_1 t_{1_0}} > 1, \qquad quit. \qquad [182]$$

If $t_2 = r_2 n_2$ is required, then the optimal discrete values are given by

$$t_{1_0}, \qquad t_{2_0} = r_2 \left[ \frac{t_{2_0}}{r_2} + 0.5 \right]_{n.i.}, \qquad W_0 = \rho_1 t_{1_0} + \rho_2 t_{2_0}, \qquad quit. \qquad [183]$$

$$9. \quad \frac{D\rho_p}{\rho_1 t_{1_0}} \le 1 \Rightarrow t_{2_0} = \frac{t_{2_0}}{\left( \frac{D\rho_p}{\rho_1 t_{1_0}} \right)}, \qquad W_0 = \rho_1 t_{1_0} + \rho_2 t_{2_0}. \qquad [184]$$

If $t_2 = r_2 n_2$ is required, the optimal discrete solution is given by

$$t_{1_0}, \qquad t_{2_0} = r_2 \left[ \frac{t_{2_0}}{r_2} + 0.5 \right]_{n.i.}, \qquad W_0 = \rho_1 t_{1_0} + \rho_2 t_{2_0}, \qquad quit. \qquad [185]$$

$$10. \quad \frac{D\rho_p}{\rho_1 t_{1_{0c}}} \le 1 \Rightarrow t_{2_{0c}} = \frac{t_{2_{0c}}}{\left( \frac{D\rho_p}{\rho_1 t_{1_{0c}}} \right)}, \qquad W_{0c} = \rho_1 t_{1_{0c}} + \rho_2 t_{2_{0c}}. \qquad [186]$$

$$11. \quad If \qquad \frac{0.440D}{S} \left( \frac{V\cos(\theta)}{L_2} \right)^{1/2} + \frac{0.132 D^2 V \cos(\theta)}{L_2 S^2} < 1.092, \qquad [187]$$

$$n_1 = \left[ \frac{W_{0c}}{2\rho_1 r_1} \right]_{n.i.} \qquad [188]$$

and return to step 5. Otherwise,

$$n_1 = \left[ \frac{W_{0c} \pm (W_{0c}^2 - 4c_1\rho_1)^{1/2}}{2\rho_1 r_1} \right]_{n.i.} \qquad [189]$$

Check both values and return to step 5.

**Proof:** The primal objective function is given by

$$W = \rho_1 t_1 + \rho_2 t_2 = \rho_1 t_1 + \frac{c_1}{t_1} \qquad [190]$$

and is constrained by $t_1 = r_1\, n_1$. The dual program is given by

$$\max v(\delta) = \left(\frac{\rho_1}{\delta_1}\right)^{\delta_1} \left(\frac{c_1}{\delta_2}\right)^{\delta_2} (r_1 n_1)^{\delta_2' - \delta_1'} \qquad [191]$$

$$s.t. \qquad \delta_1 + \delta_2 = 1 \qquad [192]$$

$$\delta_1 - \delta_2 + \delta_1' - \delta_2' = 0 \qquad [193]$$

$$\Rightarrow \max v(\delta) = \left(\frac{\rho_1}{\delta_1}\right)^{\delta_1} \left(\frac{c_1}{1-\delta_1}\right)^{1-\delta_1} (r_1 n_1)^{2\delta_1 - 1} \qquad [194]$$

$$s.t. \qquad \delta_1 \in [0, 1] \qquad [195]$$

But

$$t_1 = r_1 n_1 \wedge \nu(\delta) = \frac{\rho_1 t_1}{\delta_1} \quad [196]$$

$$\Rightarrow \frac{\rho_1 r_1 n_1}{\delta_1} = \left(\frac{\rho_1}{\delta_1}\right)^{\delta_1} \left(\frac{c_1}{1-\delta_1}\right)^{1-\delta_1} (r_1 n_1)^{2\delta_1 - 1} \quad [197]$$

$$\Rightarrow n_1 = \frac{1}{r_1}\left[\left(\frac{\rho_1}{\delta_1}\right)^{\delta_1 - 1} \left(\frac{c_1}{1-\delta_1}\right)^{1-\delta_1}\right]^{\frac{1}{2-2\delta_1}} \quad [198]$$

$$= \frac{1}{r_1}\left[\left(\frac{\rho_1}{\delta_1}\right)\left(\frac{1-\delta_1}{c_1}\right)\right]^{\frac{\delta_1 - 1}{2-2\delta_1}} \quad [199]$$

$$= \frac{1}{r_1}[(\frac{c_1}{\rho_1}\left(\frac{\delta_1}{1-\delta_1}\right)]^{1/2} \quad [200]$$

But this gives

$$r_1^2 n_1^2 = \frac{c_1}{\rho_1}\left(\frac{\delta_1}{1-\delta_1}\right) \quad [201]$$

$$\Rightarrow \rho_1 r_1^2 n_1^2 (1-\delta_1) = c_1 \delta_1 \quad [202]$$

$$\Rightarrow \delta_1 = \frac{\rho_1 r_1^2 n_1^2}{c_1 + \rho_1 r_1^2 n_1^2} \quad [203]$$

Now, minimizing the difference between the discrete and continuous dual objective functions for Wilkinson gives

$$\min \nu(\delta) - W_{0c} = \frac{\rho_1 r_1 n_1}{\delta_1} - W_{0c} \quad [204]$$

$$= \rho_1 r_1 n_1 \left(\frac{c_1 + \rho_1 r_1^2 n_1^2}{\rho_1 r_1^2 n_1^2}\right) - W_{0c} \quad [205]$$

At zero this minimum gives

$$c_1 + \rho_1 r_1^2 n_1^2 = r_1 n_1 W_{0c} \tag{206}$$

$$\Rightarrow \rho_1 r_1^2 n_1^2 - r_1 n_1 W_{0c} + c_1 = 0 \tag{207}$$

$$\Rightarrow n_1 = \frac{W_{0c} \pm (W_{0c}^2 - 4c_1\rho_1)^{1/2}}{2\rho_1 r_1} \tag{208}$$

In order for the radical to exist, we must have

$$W_0^2 - 4c_1\rho_1 \geq 0 \tag{209}$$

$$\Rightarrow W_0^2 \geq 4c_1\rho_1 = \frac{1.456 D^4 \rho_p^2 V \cos(\theta)}{L_2 S^2} \tag{210}$$

$$Now, \quad \frac{D\rho_p}{\rho_1 t_1} > 1 \tag{211}$$

$$\Rightarrow W_0^2 = \frac{1.456 D^4 \rho_p^2 V \cos(\theta)}{L_2 S^2} \tag{212}$$

$$\Rightarrow W_0^2 = 4c_1\rho_1 \tag{213}$$

$$\Rightarrow n_1 = \frac{W_0}{2\rho_1 r_1} \tag{214}$$

and the roots are identical. For

$$\frac{D\rho_p}{\rho_1 t_1} \le 1, \tag{215}$$

$$W_0 = \rho_1 t_{1_0} + \rho_2 \frac{t_{2_0}}{\left(\frac{D\rho_p}{\rho_1 t_{1_0}}\right)} \tag{216}$$

$$= \frac{0.604D^2\rho_p}{S}\left(\frac{V\cos(\theta)}{L_2}\right)^{1/2} + \frac{0.604D^2\rho_p}{S}\left(\frac{V\cos(\theta)}{L_2}\right)^{1/2}\left(\frac{0.604D^2\rho_p}{SD\rho_p}\right)\left(\frac{V\cos(\theta)}{L_2}\right)^{1/2} \tag{217}$$

$$= \frac{0.604D^2\rho_p}{S}\left(\frac{V\cos(\theta)}{L_2}\right)^{1/2} + \frac{0.364D^3\rho_p V\cos(\theta)}{L_2 S^2} \tag{218}$$

$$\therefore W_0^2 = \frac{0.364D^4\rho_p^2 V\cos(\theta)}{L_2 S^2} + \frac{0.132D^6\rho_p^2 V^2\cos^2(\theta)}{L_2^2 S^4} + \frac{0.440D^5\rho_p^2}{S^3}\left(\frac{V\cos(\theta)}{L_2}\right)^{1.5} \tag{219}$$

and

$$W_0^2 \ge 4c_1\rho_1 \Rightarrow \tag{220}$$

$$\frac{1.092D^4\rho_p^2 V\cos(\theta)}{L_2 S^2} \le \frac{0.132D^6\rho_p^2 V^2\cos^2(\theta)}{L_2^2 S^4} + \frac{0.440D^5\rho_p^2}{S^3}\left(\frac{V\cos(\theta)}{L_2}\right)^{1.5} \tag{221}$$

$$\Rightarrow \frac{D^4\rho_p^2 V\cos(\theta)}{L_2 S^2}\left(1.092 - \frac{0.132D^2 V\cos(\theta)}{L_2 S^2} - \frac{0.440D}{S}\left(\frac{V\cos(\theta)}{L_2}\right)^{1/2}\right) \le 0 \tag{222}$$

$$\Rightarrow 1.092 \le \frac{0.132D^2 V\cos(\theta)}{L_2 S^2} + \frac{0.440D}{S}\left(\frac{V\cos(\theta)}{L_2}\right)^{1/2} \tag{223}$$

and the main result is proved. Note that the justification for using nearest integer is the parabolic form of the quadratic equation

$$f(n_1) = \rho_1 r_1^2 n_1^2 - r_1 W_{0c} n_1 + c_1 = 0, \qquad W_{0c}^2 = 4c_1\rho_1 \tag{224}$$

This may be transformed to

$$\frac{W_{0c}^2}{4\rho_1^2 r_1^2 c_1} f(n_1) = \left(n_1 - \frac{W_{0c}}{2\rho_1 r_1}\right)^2 \tag{225}$$

**Figures 6.4-11, 12, and 13** show the optimal discrete design values of minimum system mass per unit area, and optimal bumper and wall thicknesses vs projectile diameter, bumper/wall separation, and projectile velocity, respectively, for the Wilkinson predictor. In **Figure 6.4-11,** the projectile density varies with diameter according to equations [13] and [14]. In **Figure 6.4-13,** the impact angle remains constant at 0 degrees (normal). The optimal bumper and wall thicknesses for the Wilkinson predictor are approximately equal due to the similarity in bumper and wall material densities.

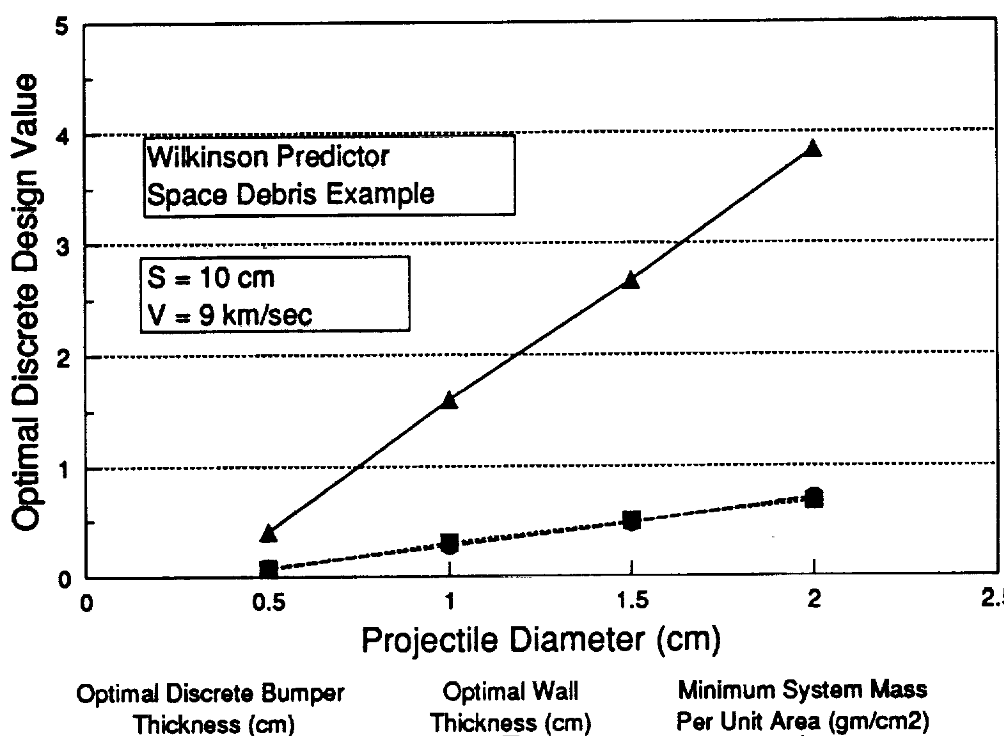



**Figure 6.4-11. Optimal Discrete Design Value vs Projectile Diameter for Wilkinson Predictor**

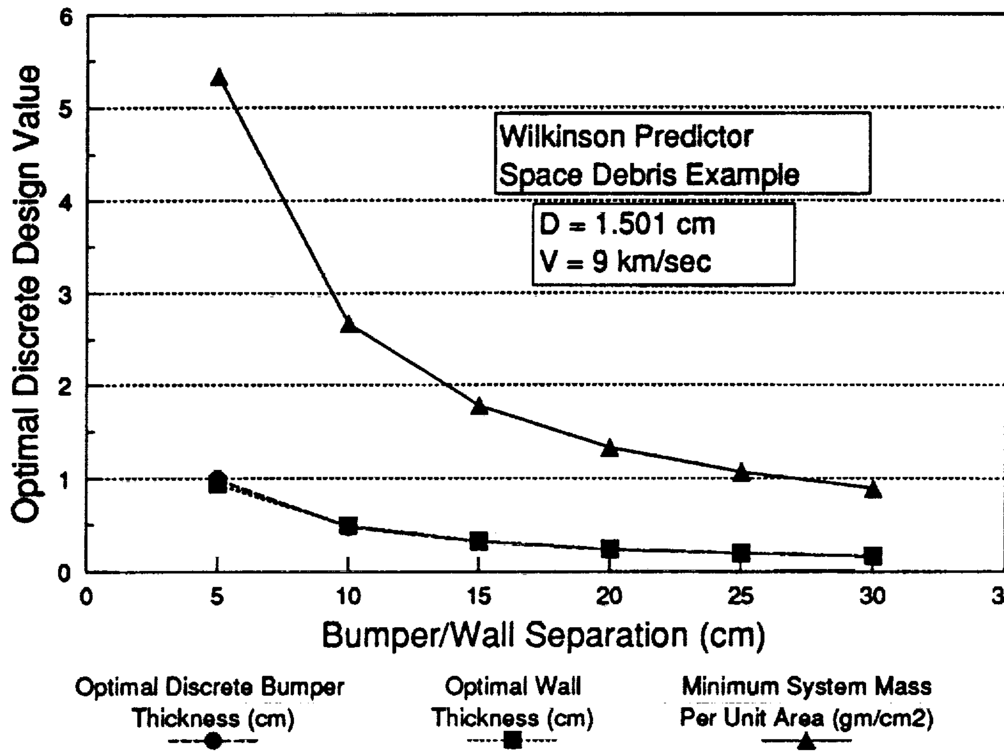



**Figure 6.4-12. Optimal Discrete Design Value vs Bumper/Wall Separation for Wilkinson Predictor**

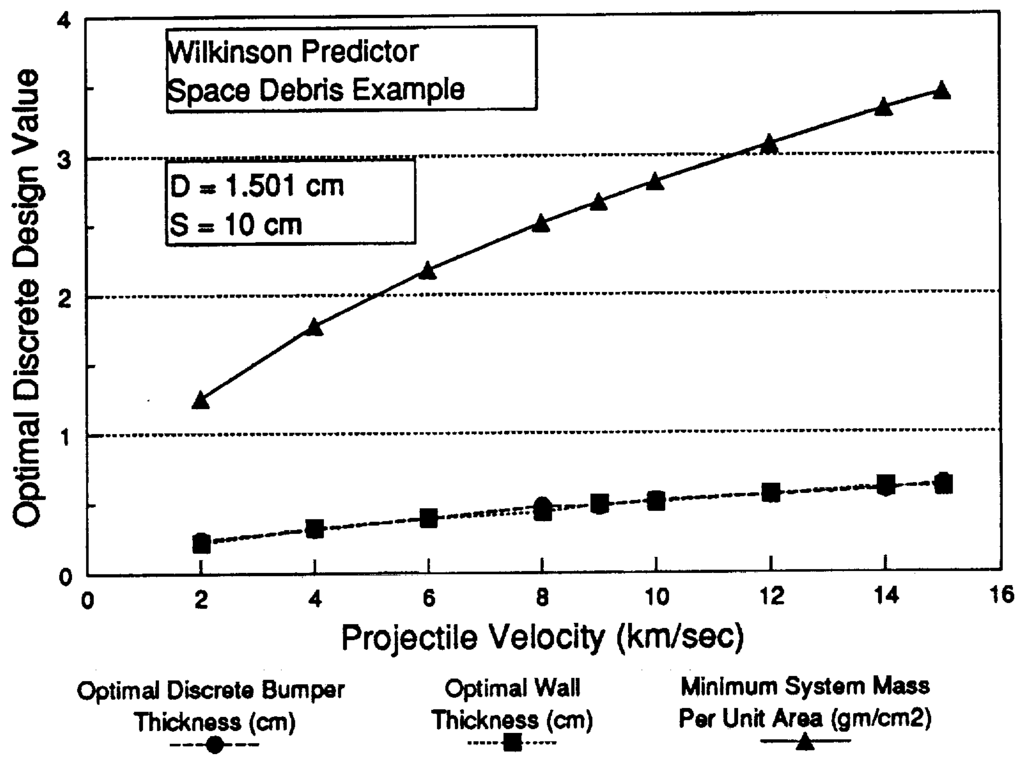



**Figure 6.4-13. Optimal Discrete Design Value vs Projectile Velocity for Wilkinson Predictor**

**Figure 6.4-14** shows the sensitivity of minimum system mass per unit area to bumper thickness availability factor, $r_1$. The discrete and continuous objective functions are equal when the continuous bumper thickness is an integer multiple of the bumper thickness availability factor as shown in **Figure 6.4-15**. This occurs at numerous locations over the range considered. Note that when $r_1$ is small, the discrete bumper thickness is closer in value to the continuous bumper thickness. As $r_1$ grows, this incidence of equality naturally decreases while the deviations from the continuous minimum mass per unit area grow in value. Beyond the optimal continuous value of the bumper thickness, the objective function continues to grow indefinitely, because the availability factor is dominating the desired continuous solution.

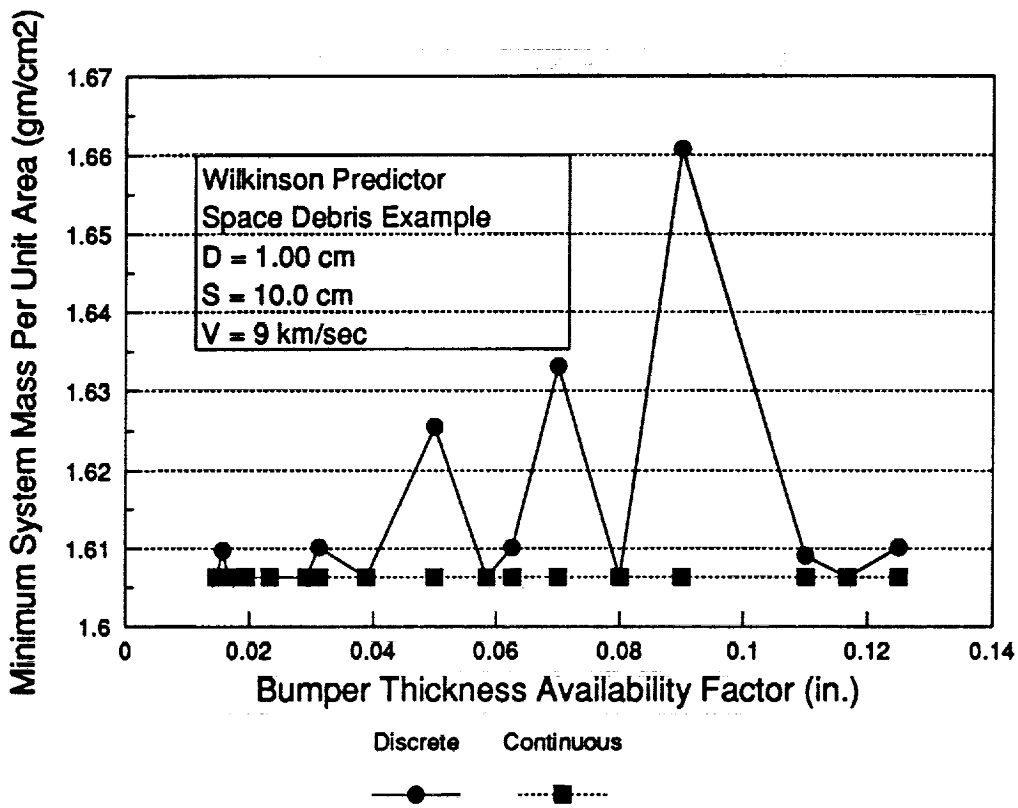



**Figure 6.4-14. Minimum System Mass Per Unit Area vs. Bumper Thickness Availability Factor for the Wilkinson Predictor**

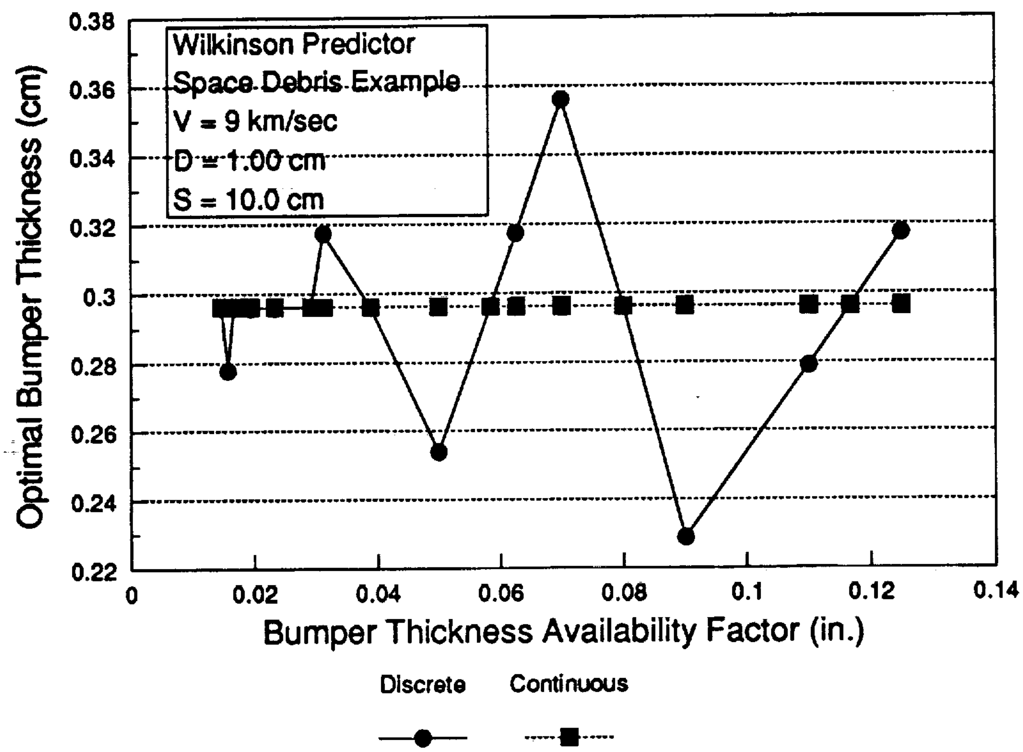



**Figure 6.4-15. Optimal Discrete and Continuous Bumper Thickness vs Bumper Thickness Availability Factor for the Wilkinson Predictor**

## 6.5 Conclusions and Recommendations for Section 6

<u>Conclusions</u>

In conclusion, global (and sometimes analytic) optimization of discrete posynomial programs can be performed using dual approaches coupled with partial invariance techniques. However, primal methods require less "pencil and paper" effort than dual methods and are more easily applied to most problems. Primal methods do not generally obtain global solutions for the discrete posynomial program. Furthermore, the dual method may be advantageous in cases where the objective function may be sufficiently separable, since posyseparable programs do not require solutions of coupled nonlinear equations in the dual-to-primal variable transformation. For protective structures design optimization problems, global nonlinear design optimization can be performed for the Wilkinson, Burch, and Nysmith impact predictors. In these cases, the optimal ratio of bumper mass per unit area to total mass per unit area may vary with mission, environment, projectile mass, and velocity regime. Additionally, there is a large incentive for increasing the bumper/wall separation from 10 to 15 cm for all three predictors investigated. All three predictors reflect increasing design sensitivity to projectile diameter and decreasing design sensitivity to bumper/wall separation. However, the Wilkinson and Nysmith predictors reflect increasing design sensitivity to projectile velocity, while the Burch relationship is decreasing.

## Recommendations

It is recommended that other primal methods be investigated, including penalty functions supported by derivative search methods and feasible direction developments for discrete posynomial programs. Additionally, computer algorithms should be implemented based on current dual codes and modifications to the discrete problem. The dual method should also be extended to signomials. In the area of spacecraft protective structures design optimization, other hypervelocity impact predictors should be investigated. The discrete methods developed in this study should also be applied to other structural design problems. Finally, alternate protective materials and configurations should be investigated.

# 7 HYPERVELOCITY IMPACT TEST SAMPLE DAMAGE ASSESSMENTS

Hypervelocity impact test sample damage assessments were performed by UAH. Posynomial regression analysis was performed by SAIC, and is available in Discrete Posynomial Programming With Applications To Spacecraft Protective Structures Design Optimization, by R. A. Mog.

The purpose of this effort is to show a posynomial regression analysis of existing hypervelocity impact test data, followed by the global optimization of the ensuing structural design problem incorporating the predictor. A posynomial (polynomial with positive coefficients and positive-valued independent variables, but not necessarily positive exponents) form is chosen for several reasons:

1. Posynomials can be globally optimized using the nonlinear geometric programming technique.
2. Many previously developed predictors (by Nysmith, Madden, Wilkinson, Richardson, etc.) are of posynomial form.
3. Posynomial regression problems may, under certain circumstances, be solved using linear regression techniques, which are easier to solve and measure statistically.

This effort focuses on the question of whether posynomial regression can be performed in a statistically significant manner. A secondary goal of the study is to provide global optimization of the design problem formulated using the derived posynomial predictor.

The development and analysis of a posynomial hypervelocity impact predictor suitable for the design of protective structures for spacecraft exposed to the meteoroid and space debris environs is presented in the reference above. The posynomial form is first developed with a number of estimated parameters. This model is next transformed into a linear regression model. Regression analysis is performed using a least squares approach to estimate the parameters, followed by analysis of variance, F-tests, and correlation coefficient examination. Residual values are then plotted against the predicted and variable values. Next, the model is transformed into a hypervelocity impact penetration predictor suitable for design. Finally, the design problem is formulated and globally optimized using posynomial programming. Results show that statistically significant posynomial impact predictors can be developed using linear regression approaches.

The main conclusion of this effort is that it is possible to develop a statistically significant posynomial hypervelocity impact predictor with a fairly large number of impact tests and a fairly small number of predictor variables. Although greater variation can be explained by considering posynomials with more than one term, the ability to transform the posynomial into a form suitable for linear regression is lost. Furthermore, since it is generally desirable to have 10 or more data points per predictor variable, the increased number of term values might actually decrease the confidence in the predictive capability of the model as measured by the analysis of variance.

# 8 ANALYSIS OF PROJECTILE SHAPE EFFECTS

SAIC developed posynomial regression techniques and combined them with posynomial optimization techniques for application to this area. These techniques are available for immediate application to the test data resulting from projectile shape effects testing. Currently, limited test data produces unclear results when attempts are made to correlate data from various projectile shapes. Results are inconclusive. Further investigation of the projectile shape effects could include methodologies found in sources such as "A Preliminary Investigation of Projectile Shape Effects In Hypervelocity Impact of a Double-Sheet Structure," by R. H. Morrison, NASA-TN-6944, August 1972, but will remain inconclusive until further test are performed.

# 9 REFERENCES

[1]Adeli, H., and Kamal, O., Efficient Optimization of Space Trusses, *Computers and Structures*, Vol. 24, No. 3, 1986.

[2]Anderson, C. E., "Ballistic Impact: The Status of Analytical and Numerical Modeling," Int. J. Impact Engng., Vol. 7, No. 1, 1988.

[3]Atrek, E., et Al, *New Directions in Optimal Structural Design*, Wiley-Interscience, 1984.

[4]Avriel, M., *Advances in Geometric Programming*, Plenum Press, New York, 1980.

[5]Avriel, M., *Nonlinear Programming Analysis and Methods*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[6]Avriel, M., "A Condensation Method for a Class of Algebraic Programs," Stanford Univ., 1975.

[7]Bartle, R. G., *The Elements of Real Analysis*, John Wiley and Sons, Inc., 1976.

[8]Bate, R. R., *Fundamentals of Astrodynamics*, Dover Publications, Inc., 1971.

[9]Bazaraa, M. S., and Jarvis, J. J., *Linear Programming and Network Flows*, John Wiley and Sons, Inc., 1977.

[10]Bazaraa, M. S., and Shetty, C. M., *Nonlinear Programming*, John Wiley and Sons, Inc., 1979.

[11]Beale, E. M. L., "Global Optimization as an Extension of Integer Programming," Towards Global Optimisation 2, 1978.

[12]Beckenbach, E., and Bellman, R., *An Introduction to Inequalities*, Yale University, 1961.

[13]Beightler, C., and Phillips, D. T., *Applied Geometric Programming*, John Wiley and Sons, Inc., 1976.

[14]Birkhoff, G., and MacLane, S., *A Survey of Modern Algebra*, Macmillan Publishing Co., Inc., 1977.

[15]Blanchard, B. S., and Fabrycky, W. J., *Systems Engineering and Analysis*, Prentice-Hall, Inc., 1981.

[16]Bless, S. J., "Hypervelocity Penetration of Ceramics," Int. J. Impact Engng., Vol. 5, 1987.

[17]Boddy, J. A., and Moss, W. L., "Weight Optimization Using Geometric Programming," Society of Aeronautical Weight Engineers, May 1970.

[18]Boyce, W. E., and DiPrima, R. C., *Elementary Differential Equations and Boundary Value Problems*, John Wiley & Sons, Inc., 1977.

[19]Brar, G. S., "Geometric Programming for Engineering Design Optimization," Ohio State University, 1981.

[20]Brimm, D. J., "Unistructure - A New Concept for Light Weight Integrally Stiffened Skin Structures," SAE Technical Paper No. 801231, October 1980.

[21]Brostow, W., *Science of Materials*, John Wiley and Sons, 1979.

[22]Brown, A. S., "Smoothing Intermetallics Brittle Edge," Aerospace America, March 1988.

[23]Budynas, R. G., *Advanced Strength and Applied Stress Analysis*, McGraw-Hill, Inc., 1977.

[24]Burns, S. A., Generalized Geometric Programming With Many Equality Constraints, *International Journal for Numerical Methods in Engineering*, Vol. 24, April 1987.

[25]Catherino, H. A., "Sizing FRP Ballistic Armor," American Society for Metals, 1985.

[26]Christiansen, E., "Evaluation of Space Station Meteoroid/Debris Shielding Materials - Final Report," NASA Contract NAS9-15800, September 1987.

[27]Churchill, R. V., et Al, *Complex Variables and Applications*, McGraw-Hill, Inc., 1976.

[28]Cole, F., et Al, A Comparison Between a Primal and a Dual Cutting Plane Algorithm for Posynomial Geometric Programming Problems, *Journal of Optimization Theory and Applications*, Vol. 47, Oct. 1985.

[29]Coronado, A. R., Gibbins, M. N., Wright, M. A., and Stern, P. H., "Space Station Integrated Wall Design and Penetration Damage Control," Boeing Aerospace Company Final Report, Contract NAS8-36426, July 1987.

[30]Cour-Palais, B. G., "Hypervelocity Impact in Metals, Glass, and Composites," Int. J. Impact Engng., Vol. 5, 1987.

[31]Cour-Palais, B., and Avans, S., "Decision Time on Orbital Debris: Shielding Against Debris," Aerospace America, June 1988.

[32]Cour-Palais, B., "Meteoroid Environment Model - 1969 (Near-Earth to Lunar Surface)," NASA SP-8013, March 1969.

[33]Dembo, R. S., Sensitivity Analysis in Geometric Programming, *Journal of Optimization Theory and Applications*, Vol. 37, May 1982.

[34]Dembo, R. S., Current State of the Art of Algorithms and Computer Software for Geometric Programming, *Journal of Optimization Theory and Applications*, Vol. 26, Oct. 1978.

[35]Dembo, R. S., Dual to Primal Conversion in Geometric Programming, *Journal of Optimization Theory and Applications*, Vol. 26, Oct. 1978.

[36]Department of Defense, "Mil-HDBK-5C, Metallic Materials and Elements for Aerospace Vehicle Structures," September 1976.

[37]Diaz, A., "Geometric Programming for Continuous Design Problems," International Symposium on Optimum Structural Design, October, 1981.

[38]Dixon, L. C. W., *Towards Global Optimisation 2*, North Holland Publishing Co., 1978.

[39]Douglas, F., "Viewport Concept for Space Station Modules," NASA TM-86559, George C. Marshall Space Flight Center, August 1986.

[40]Draper, N., and Smith, H., *Applied Regression Analysis*, John Wiley & Sons, Inc., 1981.

[41]Driver, R. D., *Ordinary and Delay Differential Equations*, Springer - Verlag New York, Inc., 1977.

[42]Duffin, R. J., Peterson, E. L., and Zener, *Geometric Programming*, John Wiley and Sons, Inc., 1967.

[43]Duffin, R. J., and Zener, "Optimization of Engineering Design by Geometric Programming," Carnegie-Mellon Univ., Feb. 1968.

[44]Duffin, R. J., and Peterson, E. L., Geometric Programming With Signomials, *Journal of Optimization Theory and Applications*, Vol. 11, Jan. 1973.

[45]Ecker, J. G., et Al, A Modified Reduced Gradient Method for Dual Posynomial Programming, *Journal of Optimization Theory and Applications*, Vol. 26, No. 2, Oct. 1978.

[46]Elliot, W. H., "Primal Geometric Programs Treated by Sequential Linear Programming," Penn. State Univ., 1976.

[47]Fair, H., "Hypervelocity Then and Now," Int. J. Impact Engng., Vol. 5, 1987.

[48]Friedberg, S. H., et Al, *Linear Algebra*, Prentice-Hall, Inc., 1979.

[49]Gantmacher, F. R., *The Theory of Matrices*, Chelsea Publishing Co., 1977.

[50]Gibbons, J. D., *Nonparametric Statistical Inference*, Marcel Dekker, Inc., 1985.

[51]Goldsmith, W., "Normal and Oblique Impact of Cylindro-Conical and Cylindrical Projectiles on Metallic Plates," Int. J. Impact Engng., Vol. 4, 1986.

[52]Gomulka, J., "Deterministic Vs. Probablistic Approaches to Global Optimisation," Towards Global Optimisation 2, 1978.

[53]Gottfried, B. S., and Weisman, J., *Introduction to Optimization Theory*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

[54]Greszuzck, L., "Impact Dynamics of Metals and Composites - UCLA Short Course Notes," July 1988.

[55]Grosh, D. L., *A Primer of Reliability Theory*, John Wiley and Sons, Inc., 1989.

[56]Haberman, R., *Elementary Applied Partial Differential Equations*, Prentice-Hall, Inc., 1983.

[57]Hack, J. E., *Mechanical Behavior of Metal Matrix Composites*, American Institute of Mining, Metallurgical, and Petroleum Engineers, Inc., 1983.

[58]Hajela, P., Geometric Programming Strategies in Large-Scale Structural Synthesis, *AIAA Journal*, Vol. 24, July 1986.

[59]Hall, M. A., and Peterson, E. L., "Traffic Equilibria Analysed Via Geometric Programming," Northwestern Univ., Feb. 1975.

[60]Herrmann, W., "Review of Hypervelocity Penetration Theories," Int. J. Impact Engng., Vol. 5, 1987.

[61]Hilado, C. J., *Boron Reinforced Aluminum Systems-Vol. 6*, TECHNOMIC Publishing Co., Inc., 1974.

[62]Hilado, C. J., *Boron Reinforced Aluminum Systems-Part II*, Vol. 11, TECHNOMIC Publishing Co., Inc., 1982.

[63]Hillier, F. S., and Lieberman, G. J., *Introduction to Operations Research*, Holden-Day, Inc., 1986.

[64]Hines, W. W., and Montgomery, D. C., *Probability and Statistics in Engineering and Management Science*, John Wiley and Sons, Inc., 1980.

[65]Hogg, R. V., *Introduction to Mathematical Statistics*, Macmillan Publishing Co., Inc., 1978.

[66]Holland, A. S. B., *Complex Function Theory*, Elsevier, North Holland, Inc., 1980.

[67]Hough, C. L., "Optimization of the Second Order Logarithmic Machine Economics Problems by Extended Geometric Programming," Texas A&M University, 1978.

[68]Huang, Y. K., *Shock Waves in Hypervelocity Impact of Metals*, University Microfilms, Inc., 1962.

[69]Ignizio, J. P., A Geometric Programming Model for Space Communication Systems, *Modeling and Simulation*, Vol. 6, April 1975.

[70]Iyengar, N. G. R., and Gupta, S. K., "Programming Methods in Structural Design," Indian Institute of Technology, 1980.

[71]Jastrzebski, Z. D., *The Nature and Properties of Engineering Materials*, John Wiley and Sons, Inc., 1979.

[72]Jones, R. M., *Mechanics of Composite Materials*, Scripta Book Co., 1975.

[73]Kaplan, W., *Advance Calculus*, Addison-Wesley Publishing, Inc., 1973.

[74]Kelly, D. W., A Dual Formulation for Generating Information about Constrained Optima in Automated Design - Based on Geometric Programming, *Computer Methods in Applied Mechanics and Engineering*, Vol. 5, May 1975.

[75]Kessler, D., "Orbital Debris Environment for Spacecraft Designed to Operate in Low Earth Orbit," NASA TM-100471, April 1989.

[76]Kessler, D., "Orbital Debris Environment for Space Station," JSC-20001.

[77]Kessler, D., "Meteoroid Environment Model-1970 [Interplanetary and Planetary]," NASA SP-8038.

[78]Keyser, C. A., *Materials Science in Engineering*, Bell and Howell Co., 1974.

[79]Kothawala, K. S., "Shaping Up: Optimization of Structural Designs," Mechanical Engineering, March 1988.

[80]Kreider., K. G., *Metallic Matrix Composites-Vol. 4*, Academic Press, Inc., 1974.

[81]Kuhlmann-Wilsdorf, D., *New Developments and Applications in Composites*, American Institute of Mining, 1979.

[82]Kyparisis, J., "Optimal Value Bounds and Posynomial Geometric Programs," George Washington University, April 1982.

[83]Lapidus, L., and Pinder, G. F., *Numerical Solution of Partial Differential Equations in Science and Engineering*, John Wiley and Sons, Inc., 1982.

[84]Lavi, A., "A User Manual for Geometric Programming," Carnegie-Mellon Univ., Pittsburgh, PA., Dec., 1974.

[85]Lawrence, R. J., "A Simple Model for the Optimization of Stand-Off Hypervelocity Particle Shields," Int. J. Impact Engng., Vol. 5, 1987.

[86]Leitmann, G., *The Calculus of Variations and Optimal Control*, Plenum Press, New York, 1981.

[87]LeMay, I., *Principles of Mechanical Metallurgy*, Elsevier North Holland, Inc., 1981.

[88]Lootsma, F. A., "Performance Evaluation of Nonlinear Optimization Methods via Pairwise Comparison and Fuzzy Number," Technische Hogeschool, 1984.

[89]Madden, R., "Ballistic Limit of Double-Walled Meteoroid Bumper Systems," NASA TN-3916, Langley Research Center, April 1967.

[90]Madden, R., "Equations for the Comparison of the Ballistic Limit of Single and Double Wall Structures," AIAA Paper No. 69-370, April 1969.

[91]Maiden, C. J., "Investigation of Fundamental Mechanism of Damage to Thin Targets by Hyper-velocity Projectiles," Final Report, GM Defense Research Lab., September 1963.

[92]Majid, K. I., *Optimum Design of Structures*, Halsted Press, 1974.

[93]Mancini, L. J., and Wilde, D. J., Interval Arithmetic in Unidimensional Signomial Programming, *Journal of Optimization Theory and Applications*, Vol. 26, No. 2, Oct. 1978.

[94]Martin, H. C., *Introduction to Matrix Methods of Structural Analysis*, McGraw-Hill, Inc., 1966.

[95]McCormick, G. P., *Nonlinear Programming Theory, Algorithms, and Applications*, John Wiley and Sons, Inc., 1983.

[96]McMasters, J. H., "The Optimization of Low-Speed Flying Devices by Geometric Programming," Purdue University, 1975.

[97]Metcalfe, A. G., *Interfaces in Metal Matrix Composites*, Academic Press, Inc., 1974.

[98]Miller, I., and Freund, J. E., *Probability and Statistics for Engineers*, Prentice-Hall, Inc., 1977.

[99]Misra, K. B., "On Optimal Reliability Design - A Review," International Federation of Automatic Control, Triennial World Congress, 6th, August 1975.

[100]Mog, R. A., Spacecraft Protective Structures Design Optimization, *AIAA Journal of Spacecraft and Rockets*, Vol. 28, No. 1, Jan.-Feb. 1991.

[101]Mog, R. A., Lovett, J. N., and Avans, S. L., "Global Nonlinear Optimization of Spacecraft Protective Structures Design," NASA TM-100387, January 1990.

[102]Mog, R. A., "Discrete Nonlinear Optimization of Spacecraft Protective Structures Design - White Paper," SAIC, Aug. 1988.

[103]Mog, R. A., "Comparison of Nonlinear Optimization Techniques for Spacecraft Protective Structures Design - White Paper," SAIC, June 1988.

[104]Mog, R. A., and Price, D. M., "Linear Programming Estimation of the Nonlinear Spacecraft Protective Systems Design Optimization Problem - White Paper," SAIC, March 1988.

[105]Mog, R. A., and Price, D. M., "Optimization Techniques Applied to Passive Measures for In-Orbit Spacecraft Survivability - Final Report," SAIC HV410-12, Contract NAS8-37378, Nov. 1987.

[106]Moon, F. C., "Wave Surfaces Due to Impact on Anisotropic Plates," J. Comp. Mat., Vol. 6, January 1972.

[107]Morris, A. J., On Condensed Geometric Programming in Structural Optimization, *Computer Methods in Applied Mechanics and Engineering*, Vol. 15, Aug. 1978.

[108]Morris, A. J., "The Optimisation of Statically Indeterminate Structures By Means of Approximate Geometric Programming," Royal Aircraft Establishment, April 1974.

[109]Morris, A. J., "Approximation and Complementary Geometric Programming," Royal Aircraft Establishment, April 1974.

[110]Morris, A. J., "Structural Optimisation by Geometric Programming," Royal Aircraft Establishment, April 1974.

[111]Morris, A. J., A Transformation for Geometric Programming Applied to the Minimum Weight Design of Statically Determinate Structures, *International Journal of Mechanical Sciences*, Vol. 17, June 1975.

[112]Morris, A. J., A Primal-Dual Method for Minimum Weight Design of Statically Determinate Structures with Several Systems of Load, *International Journal of Mechanical Sciences*, Vol. 16, Nov. 1974.

[113]Mueller, L., "ARALL: An Update," Aerospace Engineering, December 1987.

[114]Musgrave, M. J. P., "On the Propagation of Elastic Waves in Aeolotropic Media," Royal Society - Proceedings, A226, 1954.

[115]Nysmith, C. R., "An Experimental Impact Investigation of Aluminum Double-Sheet Structures," AIAA Hypervelocity Impact Conference, Cincinnati, Ohio, April 30-May 2, 1969, AIAA Paper No. 69-375.

[116]Oswald, P. F., and Kochenberger, G. A., "Optimization of Heat Transfer Designs by Geometric Programming," American Society of Mechanical Engineers, Winter Annual Meeting, November, 1972.

[117]Peterson, E. L., "Fenchel's Duality Theorem in Generalized Geometric Programming," Northwestern Univ., Evanston, IL., Oct. 1976.

[118]Peterson, E. L., "Optimality Conditions in Generalized Geometric Programming," Northwestern Univ., Evanston, IL., May, 1976.

[119]Peterson, E. L., "A Handbook of Geometric Programming," Northwestern Univ., Evanston, IL., July, 1974.

[120]Peterson, E. L., "Saddle Points and Duality in Generalized Geometric Programming," Northwestern Univ., Evanston, IL., August, 1976.

[121]Peterson, E. L., "Reduction and Decomposition of Large Generalized Geometric Programming Problems with Applications," Northwestern Univ., Evanston, IL., 1975.

[122]Peterson, E. L., "The Complementary Unboundedness of Dual Feasible Solution Sets in Convex Programming," Northwestern Univ., Evanston, IL., August, 1975.

[123]Plane, D. R., *Discrete Optimization*, Prentice-Hall, Inc., 1988.

[124]Popov, E. P., *Mechanics of Materials*, Prentice-Hall, Inc., 1978.

[125]Rainville, E. D., and Bedient, P. E., *Elementary Differential Equations*, Macmillan Publishing Co., Inc., 1974.

[126]Ramamurthy, S., "Structural Optimization Using Geometric Programming," Cornell University, 1977.

[127]Ratner, M., et Al, "Solving Geometric Programs Using GRG-Results and Comparisons," Stanford Univ., CA., May 1976.

[128]Reklaitis, G. V., "Investigation of the Computational Utility of Geometric Programming Formulation and Algorithms," Purdue Univ., West Lafayette, IN., Jan. 1981.

[129]Richardson, A. J., "Development of Dual Bumper Wall Construction for Advanced Spacecraft," Journal of Spacecraft and Rockets, Vol. 9, No. 6, June 1972.

[130]Richardson, A. J., "Theoretical Penetration Mechanics of Multisheet Structures Based on Discrete Debris Particle Modeling," Journal of Spacecraft and Rockets, Vol. 7, No. 4, April 1970.

[131]Rijckaert, M. J., and Martens, X. M., Comparison of Generalized Geometric Programming Algorithms, *Journal of Optimization Theory and Applications*, Vol. 26, No. 2, Oct. 1978.

[132]Rijckaert, M. J., and Martens, X. M., Bibliographical Note on Geometric Programming, *Journal of Optimization Theory and Applications*, Vol. 26, No. 2, Oct. 1978.

[133]Rikards, R. B., Dual Problem of Optimizing an Orthotropic Cylindrical Shell, *Mekhanika Polimerov*, September 1973.

[134]Ritger, P. D., and Rose, N. J., *Differential Equations with Applications*, McGraw-Hill, Inc., 1968.

[135]Rosenberg, E., "A Globally Convergent Condensation Method for Geometric Programming," Stanford Univ., CA., Oct. 1979.

[136]Rujikietgumjorn, S., "Development of Predictive Models for Drilling Composite Materials," Texas Technological University, 1978.

[137]Sarma, P. V. L. N., et Al, A Comparison of Computational Strategies for Geometric Programs, *Journal of Optimization Theory and Applications*, Vol. 26, No. 2, Oct. 1978.

[138]Schonberg, W. P., Taylor, R. A., Horn, J. R., "An Analysis of Penetration and Ricochet Phenomena in Oblique Hypervelocity Impact," NASA TM-100319, George C. Marshall Space Flight Center, February 1988.

[139]Schoutens, J. E., *Introduction to Metal Matrix Composite Materials*, DOD Metal Matrix Composites Information Analysis Center, 1982.

[140]Segel, L. A., *Mathematics Applied to Continuum Mechanics*, Dover Publications, Inc., 1987.

[141]Shifrin, C. A., "NASA to Evaluate Two Suit Designs for Space Station," Aviation Week and Space Technology, January 11, 1988.

[142]Smith, D. G., "Debris Impact on Earth-Orbiting Spacecraft," Contract NASA-NGT-01-002-099, 1984.

[143]Smith, G. D., *Numerical Solutions of Partial Differential Equations: Finite Difference Methods*, Oxford University Press, 1978.

[144]Smith, R. E., and West, G. S., "Space and Planetary Environment Criteria Guidelines for Use in Space Vehicle Development," NASA TM-82478, George C. Marshall Space Flight Center, 1982.

[145]Snell, M. B., and Bartholomew, P., The Application of Geometric Programming to the Structural Design of Aircraft Wings, *Aeronautical Journal*, Vol. 86, August 1982.

[146]Stewart, G. W., *Introduction to Matrix Computations*, Academic Press, Inc., 1973.

[147]Susko, M., "A Review of Micrometeoroid Flux Measurements and Models for Low Orbital Altitudes of the Space Station," NASA TM-86466, George C. Marshall Space Flight Center, September 1984.

[148]Swift, H. F., "Designing Dual-Plate Meteoroid Shields - A New Analysis," JPL Publication 82-39, March 1982.

[149]Swift, H. F., "The Effects of Bumper Material Properties on the Operation of Spaced Hypervelocity Particle Shields," AIAA Paper No. 69-379, April 1969.

[150]Swokowski, E. W., *Calculus with Analytic Geometry*, Prindle, Weber, & Schmidt, Inc., 1975.

[151]Templeman, A. B., "The Use of Geometric Programming Methods for Structural Optimization," Liverpool Univ., Sept. 1973.

[152]Templeman, A. B., and Winterbottem, S. K., "Structural Design Applications of Geometric Programming," Liverpool Univ., Nov. 1973.

[153]Unklesbay, K., et Al, Optimal Design of Journal Bearings, *International Journal of Engineering Science*, Vol. 11, September 1973.

[154]Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design with Applications*, McGraw-Hill, Inc., 1984.

[155]Walvekar, A. G., et Al, Optimal Design of Indeterminate Truss Using Geometric Programming, *Aeronautical Society of India Journal*, Vol. 24, May 1972.

[156]Watson, S. R., and Buede, D. E., *Decision Synthesis*, Cambridge University Press, 1987.

[157]Wheeden, R. L., *Measure and Integral*, Marcel Dekker, Inc., 1977.

[158]Whittle, P., *Optimization Under Constraints*, John Wiley and Sons, Ltd., 1971.

[159]Wilde, D. J., *Globally Optimal Design*, John Wiley and Sons, Inc., 1978.

[160]Wilkinson, J. P. D., "A Penetration Criterion for Double-Walled Structures Subject to Meteoroid Impact," AIAA Journal, Vol. 7, No. 10, October 1969.

[161]Woodward, R. L., "The Interrelation of Failure Modes Observed in the Penetration of Metallic Targets," Int. J. Impact Engng., Vol. 2, 1984.

[162]Wu, C. C., "Design and Modeling of Solar Sea Power Plants by Geometric Programming," Carnegie-Mellon University, April 1976.

[163]Wu, C. T., "Reduction and Restriction Methods for Simplifying and Solving Nonlinear Programming Problems," Northwestern University, 1975.

[164]Yew, C. H., "A Study of Damage in Composite Panels Produced by Hypervelocity Impact," Int. J. Impact Engng., Vol. 5, 1987.

[165]Zhu, G., "Estimation of the Plastic Structural Response Under Impact," Int. J. Impact Engng., Vol. 4, 1986.

[166]Zotos, J., *Mathematical Modeling of the Chemical, Mechanical, and Physical Properties of Engineering Alloys*, D. C. Heath and Co., 1977.

[167]Zukas, J. A., *Impact Dynamics*, John Wiley and Sons, Inc., 1982.

# 10 APPENDICES

## APPENDIX A. POLYPRIME.FOR:

## A GENERALIZED PRIMAL OPTIMIZATION TECHNIQUE USING PENALTY FUNCTIONS

```
      DIMENSION C(100),A(100,100),X(100),M(100),CI(100,100)
    DIMENSION AI(100,100,100),XKL(100),XHIGH(100),XLOW(100)
    DIMENSION RJ(100),IDIS(100),XOPT(100),XEXP(100)
    DIMENSION EPS1(100),EPS2(100)
    OPEN(UNIT=10,TYPE='OLD',ACCESS='SEQUENTIAL')
    OPEN(UNIT=11,TYPE='NEW',ACCESS='SEQUENTIAL')
***INITIAL SEED FOR RANDOM SEARCH***
    ISEED = 91411
***NUMBER OF CASES TO RUN***
    READ(10,*) NRUNS
    DO 10 IR=1,NRUNS
***IOPT=1 FOR RANDOM SEARCH, 2 FOR HOOKE AND JEEVES***
      READ(10,*)IOPT
***NUMBER OF TERMS IN OBJECTIVE FUNCTION***
      READ(10,*)N
***NUMBER OF INDEPENDENT VARIABLES***
      READ(10,*)K
***NUMBER OF CONSTRAINTS***
      READ(10,*)P
      DO 15 I=1,N
***COEFFICIENT FOR EACH TERM IN OBJECTIVE FUNCTION***
        READ(10,*)C(I)
        DO 20 J=1,K
***EXPONENT FOR OBJ. FUNC. BY VARIABLE AND TERM***
          READ(10,*)A(I,J)
20        CONTINUE
15      CONTINUE
      DO 25 L=1,P
***NUMBER OF TERMS BY CONSTRAINT NUMBER***
        READ(10,*)M(L)
***RIGHT-HAND-SIDE BY CONSTRAINT NUMBER***
        READ(10,*)XKL(L)
        DO 30 I=1,M(L)
***COEFFICIENT BY TERM AND CONSTRAINT NUMBER***
          READ(10,*)CI(I,L)
          DO 35 J=1,K
***EXPONENT BY TERM, VARIABLE, AND CONSTRAINT NUMBER***
            READ(10,*)AI(I,J,L)
35          CONTINUE
30        CONTINUE
```

```
25     CONTINUE
      DO 36 I=1,K
***IDIS = 1 FOR DISCRETE VARIABLES***
        READ(10,*)IDIS(I)
        IF(IDIS(I).EQ.1) THEN
***DISCRETE FACTOR BY VARIABLE***
          READ(10,*)RJ(I)
        ENDIF
36     CONTINUE
***INITIAL PENALTY FUNCTION ACCELERATION FACTOR***
      ACCEL = 1.0
      IF(IOPT.EQ.1) THEN
***RANDOM SEARCH***
        CALL RSEARCH(IDIS,ISEED,N,K,P,C,A,M,XKL,CI,AI,RJ,ACCEL,X)
        GO TO 1000
      ENDIF
      IF(IOPT.EQ.2)THEN
***HOOKE AND JEEVES***
        CALL HJ(IDIS,N,K,P,C,A,M,XKL,CI,AI,RJ,ACCEL,X)
      ENDIF
1000   CONTINUE
10   CONTINUE
     STOP
     END
     SUBROUTINE RSEARCH(IDIS,ISEED,N,K,P,C,A,M,XKL,CI,AI,RJ,ACCEL,X)
     DIMENSION C(100),A(100,100),X(100),M(100),CI(100,100)
     DIMENSION AI(100,100,100),XKL(100),XHIGH(100),XLOW(100)
     DIMENSION RJ(100),IDIS(100),XOPT(100),XEXP(100)
     DIMENSION EPS1(100),EPS2(100),MULT(100)
***FRACTION OF INTERVAL REQUIRED AND CONFIDENCE LEVEL***
     READ(10,*)FRS,XPRS
***NUMBER OF SEARCH POINTS***
     NPOINTS = IFIX(-1.0*ALOG(1.0-XPRS)/(FRS**K)+1.0)
     WRITE(6,*)'YOU WILL BE SEARCHING',NPOINTS,'POINTS'
     DO 40 I=1,K
***LOWER AND UPPER BOUNDS BY VARIABLE***
       READ(10,*)XLOW(I),XHIGH(I)
40   CONTINUE
***INITIALIZING NUMBER OF DISCRETE POINTS AND VARIABLES***
     DPOINTS=1.0
     NDVAR=0
     DO 41 I=1,K
       IF(IDIS(I).EQ.1)THEN
        NDVAR=NDVAR+1
***CALCULATES TOTAL NUMBER OF FEASIBLE DISCRETE POINTS IN INTERVAL***
        DPOINTS=(XHIGH(I)-XLOW(I))/RJ(I)*DPOINTS
       ENDIF
41   CONTINUE
***IF THE PROBLEM ISNT MIXED***
```

```
105  IF(NDVAR.EQ.K)THEN
***IF THE INTERVAL IS NOT DENSE IN DISCRETE FEASIBLE POINTS RELATIVE TO***
***THE NUMBER YOU WERE WILLING TO SEARCH ANYWAY, JUST SEARCH FEASI-
BLE POINTS***
      IF(DPOINTS.LE.NPOINTS)THEN
       DO 42 I=1,K
         MULT(I)=IFIX(XLOW(I)/RJ(I))+1
***LOWEST DISCRETE FEASIBLE POINT***
         X(I)=MULT(I)*RJ(I)
42       CONTINUE
      ICOUNT=0
      DO 44 J=1,K
***CONTINUE AS LONG AS DISCRETE POINTS ARE FEASIBLE***
47       IF(X(J).LE.XHIGH(J))THEN
         CALL OBJ(IDIS,C,A,X,CI,AI,XKL,N,K,P,M,RJ,FUNC,ACCEL,XPF)
         IF(ICOUNT.EQ.0)THEN
***INITIALIZE OPTIMAL VALUES***
          FUNCOPT=FUNC
          XPFOPT=XPF
          DO 43 I=1,K
            XOPT(I)=X(I)
43          CONTINUE
         ENDIF
         ICOUNT=ICOUNT+1
         IF(FUNC.LT.FUNCOPT)THEN
***UPDATE OPTIMAL VALUES***
          FUNCOPT=FUNC
          XPFOPT=XPF
          DO 46 L=1,K
            XOPT(L)=X(L)
46          CONTINUE
         ENDIF
***INCREMENT DISCRETE SEARCH POINTS***
         MULT(J)=MULT(J)+1
         X(J)=MULT(J)*RJ(J)
         GO TO 47
        ENDIF
***UPDATE OPTIMAL VALUES***
        DO 48 I=1,K
         X(I)=XOPT(I)
48       CONTINUE
44      CONTINUE
***      WRITE(6,*)FUNCOPT,XPFOPT,(XOPT(I),I=1,K)
     GO TO 99
     ENDIF
     ENDIF
***IF THE PROBLEM IS MIXED OR CONTINUOUS OR FULLY DISCRETE WITH A DENSE
***COVERING OF FEASIBLE POINTS IN THE INTERVAL, PROCEED WITH STANDARD
RANDOM
```

```
***SEARCH***
   DO 45 I=1,NPOINTS
     DO 50 J=1,K
***CALCULATE RANDOM SEARCH POINT***
       X(J)=XLOW(J)+RAN(ISEED)*(XHIGH(J)-XLOW(J))
50     CONTINUE
   CALL OBJ(IDIS,C,A,X,CI,AI,XKL,N,K,P,M,RJ,FUNC,ACCEL,XPF)
   IF(I.EQ.1)THEN
***INITIALIZE OPTIMAL VALUES***
    FUNCOPT=FUNC
    XPFOPT=XPF
    DO 85 L=1,K
      XOPT(L)=X(L)
85    CONTINUE
   ENDIF
   IF(FUNC.LT.FUNCOPT)THEN
***UPDATE OPTIMAL VALUES***
    FUNCOPT=FUNC
    XPFOPT=XPF
    DO 90 L=1,K
      XOPT(L)=X(L)
90    CONTINUE
   ENDIF
***    WRITE(6,*)XPFOPT
45  CONTINUE
***DOES PENALTY CONVERGE?***
99  IF(XPFOPT.LE.0.001)THEN
    WRITE(11,*)'MIN. OBJ. FUNC. VALUE =',FUNCOPT
    DO 95 L=1,K
      WRITE(11,*)'X',L,'=',XOPT(L)
95    CONTINUE
    GO TO 100
   ENDIF
***UPDATE PENALTY FUNCTION ACCELERATING FACTOR IF PENALTY DOESNT
***CONVERGE***
   ACCEL=ACCEL*10.0
   GO TO 105
100 CONTINUE
   RETURN
   END
   SUBROUTINE HJ(IDIS,N,K,P,C,A,M,XKL,CI,AI,RJ,ACCEL,X)
   DIMENSION C(100),A(100,100),X(100),M(100),CI(100,100)
   DIMENSION AI(100,100,100),XKL(100),XHIGH(100),XLOW(100)
   DIMENSION RJ(100),IDIS(100),XOPT(100),XEXP(100)
   DIMENSION EPS1(100),EPS2(100)
   IDIFF=0
   DO 110 I=1,K
***READ INITIAL POINT, INITIAL EXPLORATORY VALUES, AND FINAL EXPLOR-
ATORY
```

```
***VALUES***
      READ(10,*)X(I),EPS1(I),EPS2(I)
      XOPT(I)=X(I)
110   CONTINUE
    CALL OBJ(IDIS,C,A,X,CI,AI,XKL,N,K,P,M,RJ,FUNC,ACCEL,XPF)
***     WRITE(11,*)'1',FUNC
***INITIALIZE OPTIMAL VALUES***
    FUNCOPT=FUNC
    XPFOPT=XPF
***     WRITE(11,*)FUNCOPT,XPFOPT
    DO 115 I=1,K
***PERFORM EXPLORATORY SEARCH FROM BASE POINT***
      XEXP(I)=X(I)
      X(I)=X(I)+EPS1(I)
      CALL OBJ(IDIS,C,A,X,CI,AI,XKL,N,K,P,M,RJ,FUNC,ACCEL,XPF)
***     WRITE(11,*)'2',FUNC
      IF(FUNC.GT.FUNCOPT)THEN
***GO IN OTHER DIRECTION***
        X(I)=X(I)-2.0*EPS1(I)
***         DO 1134 KLM=1,K
***         WRITE(11,*)'KLM=',KLM,X(KLM)
*** 1134      CONTINUE
        CALL OBJ(IDIS,C,A,X,CI,AI,XKL,N,K,P,M,RJ,FUNC,ACCEL,XPF)
***     WRITE(11,*)'3',FUNC
        IF(FUNC.LT.FUNCOPT)THEN
***UPDATE OPTIMAL VALUES***
          FUNCOPT=FUNC
          XOPT(I)=X(I)
          XPFOPT=XPF
          IDIFF=1
***           DO 1125 JIK=1,K
***           WRITE(11,*)'111',FUNCOPT,XOPT(JIK),XPFOPT
***  1125       CONTINUE
          GO TO 111
        ENDIF
      GO TO 111
      ENDIF
***UPDATE OPTIMAL VALUES***
      XOPT(I)=X(I)
      XPFOPT=XPF
      FUNCOPT=FUNC
      IDIFF=1
***           DO 1126 JIK=1,K
***           WRITE(11,*)'115',FUNCOPT,XOPT(JIK),XPFOPT
***  1126       CONTINUE
111    CONTINUE
115   CONTINUE
135   IF(IDIFF.EQ.1)THEN
     DO 120 I=1,K
```

```
***IF NEW POINT IN EXPLORATION IS DIFFERENT FROM BASE POINT, PERFORM
***PATTERN SEARCH***
        XEXP(I)=XEXP(I)+2.0*(XOPT(I)-XEXP(I))
        X(I)=XEXP(I)
120     CONTINUE
      ENDIF
***OTHERWISE, REDUCE EXPLORATORY VALUES (EPSILONS)***
      IF(IDIFF.EQ.0)THEN
       GO TO 140
      ENDIF
      IDIFF=0
      CALL OBJ(IDIS,C,A,X,CI,AI,XKL,N,K,P,M,RJ,FUNC,ACCEL,XPF)
***     WRITE(11,*)'4',FUNC
      IF(FUNC.LT.FUNCOPT)THEN
***UPDATE OPTIMAL VALUES***
       FUNCOPT=FUNC
       XPFOPT=XPF
       DO 121 I=1,K
        XOPT(I)=X(I)
***          DO 1127 JIK=1,K
***          WRITE(11,*)'121',FUNCOPT,XOPT(JIK),XPFOPT
***  1127       CONTINUE
121     CONTINUE
175     DO 125 I=1,K
***PERFORM (NEW) EXPLORATION***
        X(I)=X(I)+EPS1(I)
        CALL OBJ(IDIS,C,A,X,CI,AI,XKL,N,K,P,M,RJ,FUNC,ACCEL,XPF)
***     WRITE(11,*)'5',FUNC
        IF(FUNC.GT.FUNCOPT)THEN
***GO IN OTHER DIRECTION***
         X(I)=X(I)-2.0*EPS1(I)
***        DO 1136 KLM=1,K
***        WRITE(11,*)'KLM=',KLM,X(KLM)
***  1136       CONTINUE
         CALL OBJ(IDIS,C,A,X,CI,AI,XKL,N,K,P,M,RJ,FUNC,ACCEL,XPF)
***     WRITE(11,*)'6',FUNC
         IF(FUNC.LE.FUNCOPT)THEN
***UPDATE OPTIMAL VALUES***
          FUNCOPT=FUNC
          XOPT(I)=X(I)
          XPFOPT=XPF
***          DO 1128 JIK=1,K
***          WRITE(11,*)'175',FUNCOPT,XOPT(JIK),XPFOPT
***  1128       CONTINUE
          IDIFF=1
         ENDIF
         GO TO 125
        ENDIF
***UPDATE OPTIMAL VALUES***
```

```
        XOPT(I)=X(I)
        IDIFF=1
        FUNCOPT=FUNC
        XPFOPT=XPF
***          DO 1129 JIK=1,K
***          WRITE(11,*)'140',FUNCOPT,XOPT(JIK),XPFOPT
*** 1129       CONTINUE
125  CONTINUE
140  IF(IDIFF.EQ.0)THEN
       DO 130 I=1,K
***REDUCE EXPLORATORY VALUES WHEN NO IMPROVEMENT IS MADE IN
EXPLORATION***
        EPS1(I)=EPS1(I)/2.0
        IF(EPS1(I).LT.EPS2(I))THEN
***CHECK ENDING CONDITION BASED ON EXPLORATORY VALUES***
         GO TO 150
        ENDIF
130   CONTINUE
***GO EXPLORE SOME MORE***
       GO TO 175
     ENDIF
     ENDIF
     DO 137 KIM=1,K
***RETAIN PREVIOUS BASE POINTS FOR FUTURE PATTERN MOVES***
       X(KIM)=XOPT(KIM)
137  CONTINUE
     IF(IDIFF.EQ.0)THEN
      GO TO 140
     ENDIF
***MAKE PATTERN MOVE***
     GO TO 135
150  CONTINUE
***CHECK PENALTY VALUE FOR CONVERGENCE***
     IF(XPFOPT.LE.0.001)THEN
      DO 1130 JIK=1,K
      WRITE(11,*)'MIN. OBJ. FUNC. VALUE =',FUNCOPT
      WRITE(11,*)'PENALTY = ',XPFOPT
      WRITE(11,*)'ACCELERATION FACTOR = ',ACCEL
1130   CONTINUE
      DO 170 L=1,K
        WRITE(11,*)'X',L,'=',XOPT(L)
170    CONTINUE
      GO TO 200
     ENDIF
***UPDATE PENALTY FUNCTION ACCELERATION FACTOR IF CONVERGENCE IS NOT
***ACHIEVED***
     ACCEL=ACCEL*10.0
     GO TO 110
200  CONTINUE
```

```
      RETURN
      END
      SUBROUTINE OBJ(IDIS,C,A,X,CI,AI,XKL,N,K,P,M,RJ,FUNC,ACCEL,XPF)
      DIMENSION C(100),A(100,100),X(100),M(100),CI(100,100)
      DIMENSION AI(100,100,100),XKL(100),XHIGH(100),XLOW(100)
      DIMENSION RJ(100),IDIS(100),XOPT(100),XEXP(100)
      DIMENSION EPS1(100),EPS2(100)
      FUNC=0.0
      DO 55 IJ=1,N
***INITIALIZE PRODUCT VALUES***
        XPROD=1.0
        DO 60 IJK=1,K
******          WRITE(6,*)X(IJK),A(IJ,IJK)
***ZERO EXPONENTS GIVE PRODUCT VALUES OF 1.0***
          IF(ABS(A(IJ,IJK)).LE.0.0000001)THEN
            GO TO 60
          ENDIF
***ZERO VARIABLE VALUES GIVE PRODUCT VALUES OF ZERO***
          IF(ABS(X(IJK)).LE.0.0000001)THEN
            XPROD=0.0
            GO TO 60
          ENDIF
***COMPUTERS DONT RAISE NEG. VALUES TO EXPONENTS***
          IF(X(IJK).LT.0.0)THEN
            X(IJK)=ABS(X(IJK))
          ENDIF
***COMPUTE PRODUCT VALUES FOR OBJ. FUNCTION***
          XPROD=XPROD*X(IJK)**A(IJ,IJK)
60      CONTINUE
***COMPUTE ORIGINAL OBJECTIVE FUNCTION VALUE***
        FUNC=FUNC+C(IJ)*XPROD
55    CONTINUE
      XPF=0.0
      DO 65 IJ=1,P
***INITIALIZE CONSTRAINT SUMS***
        CONSUM=0.0
        DO 70 LJ=1,M(IJ)
***INITIALIZE CONSTRAINT PRODUCTS***
          CONPROD=1.0
          DO 75 MJ=1,K
***UPDATE CONSTRAINT PRODUCTS***
            CONPROD=CONPROD*X(MJ)**AI(LJ,MJ,IJ)
75        CONTINUE
***UPDATE CONSTRAINT SUMS***
          CONSUM=CONSUM+CI(LJ,IJ)*CONPROD
70      CONTINUE
***COMPUTE PENALTY AND FUNCTION FOR <= CONSTRAINTS***
          IF((CONSUM-XKL(IJ)).GT.0.0)THEN
            XPF=XPF+ACCEL*(CONSUM-XKL(IJ))**2.0
```

```
        FUNC=FUNC+XPF
      ENDIF
65  CONTINUE
    DO 80 IJ=1,K
***COMPUTE PENALTY AND FUNCTION FOR DISCRETE CONSTRAINTS***
      IF(IDIS(IJ).EQ.1)THEN
      XPF1=ACCEL*(ABS(X(IJ)/RJ(IJ)-IFIX(X(IJ)/RJ(IJ)+0.5)))**0.5
      XPF=XPF+XPF1
      FUNC=FUNC+XPF1
      ENDIF
80  CONTINUE
    RETURN
    END
```

## APPENDIX B. IMPACT10 SOURCE CODE LISTING

```
      DIMENSION XPV(100), SOLAR(1188), XPSIV(125), XMETIV(100)
   DIMENSION XDEBOLDIV(100)
   DIMENSION ISWITCH(10)
   character BUMPER_NAME(50)*14,WALL_NAME(50)*11
   CHARACTER  BUMPER_MAT_NAME*40 , WALL_MAT_NAME*40
   CHARACTER BUMPER_TYPE_NAME*40,WALL_TYPE_NAME*40
   CHARACTER  SHAPE*40
   character  Outdir*40
   character  line1*80
   character  line2*80
   data cdate / 'Run_Date  '/
   data ctime / 'Run_Time  '/
   call gettim (ihr,imin, isec, i100th)
   call getdat (iyr, imon, iday)
        OPEN(UNIT=27,STATUS='old',ACCESS='SEQUENTIAL',
  +   FILE='config.pgm')
   read(27,2312)outdir
   read(27,2312)outdir
2312  format(A40)
   close(27)
   OPEN(UNIT=23,STATUS='OLD',ACCESS='SEQUENTIAL',FILE='CRAFT.INP')
   OPEN(UNIT=26,STATUS='OLD',ACCESS='SEQUENTIAL',FILE='GEOMETRY.INP')
   IJK = INDEX(OUTDIR,' ') -1
   OPEN(UNIT=27,STATUS='unknown',ACCESS='SEQUENTIAL',
  +   FILE= outdir(1:IJK) // 'Z9AAAAAJ.PGM')
   OPEN(UNIT=28,STATUS='unknown',ACCESS='SEQUENTIAL',
  +   FILE='PROJECT.OUT')
   OPEN(UNIT=29,STATUS='unknown',ACCESS='SEQUENTIAL',
  +   FILE='results.dat')

   open(unit=33,status='old',access='sequential',
  +    file='project.hdr')
          write(28, '(1x, A10,1x, I2.2, 1H:, I2.2, 1H:, I2.2, 1H.,
  +I2.2)') ctime, ihr, imin, isec, i100th
     write(28, '(1x, A10,1x, I2.2, 1H-, I2.2, 1H-, I4.2)')
  + cdate, imon, iday, iyr
          do 6008 i = 1,6
     read(33,6007)line1
     write(28,6007)line1
6008   continue
    close(unit=33)
       write(*,*)'          IMPACT10V -- SAIC / Huntsville'
   write(*,*)'          ------------------------------'
   write(*,*)
   write(*,*)'Status - Initializing Files'
```

```
    C-------------------------------
C   READ Static Data Files
C
                                OPEN(UNIT=14,STATUS='OLD',ACCESS='SEQUENTIAL',-
FILE='FLUXFAC.DAT')
      DO 222 KI=1,101
        READ(14,*)JI,XPSIV(JI)
222   CONTINUE
      CLOSE (UNIT = 14)
          OPEN(UNIT=14,STATUS='OLD',ACCESS='SEQUENTIAL',FILE='SOLAR1.FLX')
      DO 223 KI=1,1188
        READ(14,*)SOLAR(KI)
223   CONTINUE
      CLOSE (UNIT = 14)
          OPEN(UNIT=14,STATUS='OLD',ACCESS='SEQUENTIAL',FILE='METVEL.INP')
      DO 224 KI=1,72
        READ(14,*)XMETIV(KI)
224   CONTINUE
      CLOSE (UNIT = 14)
            OPEN(UNIT=14,STATUS='OLD',ACCESS='SEQUENTIAL',
     + FILE='DEBOLDVE.DAT')
      DO 225 KI=1,16
        READ(14,*)IV,XDEBOLDIV(IV)
225   CONTINUE
      CLOSE (UNIT = 14)
        c
c   Read PSDOC controlling switches and set impact10 variables accordingly
c
      OPEN(UNIT=14,STATUS='OLD',ACCESS='SEQUENTIAL',
     + FILE='SWITCH.INP')
      DO 226 KI=1,10
        READ(14,*)ISWITCH(KI)
226   CONTINUE
      CLOSE (UNIT = 14)
            NENVIRON = ISWITCH(1)
       IBUMPER_TYPE=ISWITCH(2)
       IBUMPER_MATERIAL=ISWITCH(3)
       IWALL_TYPE = ISWITCH(4)
       IWALL_MATERIAL=ISWITCH(5)
       ISW6 = ISWITCH(6)
       ISW7 = ISWITCH(7)
       ISW8 = ISWITCH(8)
C      NOTE:  CURRENTLY WE ARE NOT MAKING USE OF ISWITCH(9)
       IGRAPH_TYPE=ISWITCH(10)


        C   Open appropriate files for environment depending on switch
c     settings.
c
```

```
      IF (NENVIRON.EQ.1) THEN
        OPEN(UNIT=20,STATUS='OLD',ACCESS='SEQUENTIAL',
     +    FILE='NEWDEBRI.INP')
      END IF
          IF (NENVIRON.EQ.2) THEN
        OPEN(UNIT=20,STATUS='OLD',ACCESS='SEQUENTIAL',
     +    FILE='OLDDEBRI.INP')
      END IF
          IF (NENVIRON.EQ.3) THEN
       OPEN(UNIT=20,STATUS='OLD',ACCESS='SEQUENTIAL',
     +   FILE='NE_MET.INP')
      END IF
      C    Open appropriate files and read data from bumper database if
c    the table data is used rather than the single material (parametric)
c    settings.
C
      C-----------------
C    BUMPER DATABASE
C-----------------
      IF (IBUMPER_TYPE.EQ.1) THEN
        BUMPER_TYPE_NAME='Bumper Material Database'
        IF (IBUMPER_MATERIAL.EQ.1) THEN
          BUMPER_MAT_NAME='Aluminum Alloy'
          OPEN(UNIT=30,STATUS='OLD',ACCESS='SEQUENTIAL',
     +      FILE='ALBUMP.INP')
          OPEN(UNIT=32,STATUS='OLD',ACCESS='SEQUENTIAL',
     +      FILE='ALBM.TBL')
        ELSE IF (IBUMPER_material.EQ.2)THEN
          BUMPER_MAT_NAME='Titanium Alloy'
          OPEN(UNIT=30,STATUS='OLD',ACCESS='SEQUENTIAL',
     +      FILE='TIBUMP.INP')
          OPEN(UNIT=32,STATUS='OLD',ACCESS='SEQUENTIAL',
     +        FILE='TITBM.TBL')
        ELSE IF (IBUMPER_MATERIAL.EQ.3)THEN
          BUMPER_MAT_NAME='Steel Alloy'
          OPEN(UNIT=30,STATUS='OLD',ACCESS='SEQUENTIAL',
     +      FILE='STBUMP.INP')
          OPEN(UNIT=32,STATUS='OLD',ACCESS='SEQUENTIAL',
     +      FILE='STBM.TBL')
        ELSE IF (IBUMPER_MATERIAL.EQ.4) THEN
          BUMPER_MAT_NAME='Inconel Alloy'
          OPEN(UNIT=30,STATUS='OLD',ACCESS='SEQUENTIAL',
     +      FILE='INBUMP.INP')
          OPEN(UNIT=32,STATUS='OLD',ACCESS='SEQUENTIAL',
     +      FILE='INCBM.TBL')
        ELSE IF (IBUMPER_MATERIAL.EQ.5)THEN
          BUMPER_MAT_NAME='Graphite Alloy'
          OPEN(UNIT=30,STATUS='OLD',ACCESS='SEQUENTIAL',
     +      FILE='GRBUMP.INP')
```

```
      OPEN(UNIT=32,STATUS='OLD',ACCESS='SEQUENTIAL',
    +   FILE='GRALBM.TBL')
     ELSE
      write(*,*)'Input Error from PSDOC interface - '
      write(*,*)'Program Terminated with internal error.'
      write(*,*)'Bad Ibumper_material switch.            '
      goto 11
     END IF
    C  Read Number of bumpers in selected table
      Read(30,31)nbump
31     format(I10)

    c  Read Table files for Material names
c  Skip 7 line header first
          DO 5010 KI=1,7
        READ(32,*)
5010     CONTINUE
      DO 627 KI=1,nbump
        READ(32,6000,END=628)BUMPER_NAME(KI)
627     CONTINUE
6000     format(a14)
628     CLOSE (UNIT = 32)


    c For parametric settings, open bumper.inp using same file handle as table
c  setting.  This will allow us to use the same code regardless of the method
c  chosen.
    C-----------------------------
C   SINGLE  BUMPER MATERIAL
C-----------------------------
     else IF  (IBUMPER_TYPE.EQ.2) THEN
      BUMPER_TYPE_NAME='Single Bumper Material'
      OPEN(UNIT=30,STATUS='OLD',ACCESS='SEQUENTIAL',
    +   FILE='BUMPER.INP')
      NBUMP = 1
     else
      write(*,*)'Input Error from PSDOC interface - '
      write(*,*)'Program Terminated with internal error.'
      write(*,*)'Bad Ibumper_type switch.           '
      goto 11
     END IF

    C    Open appropriate files and read data from wall database if
c    the table data is used rather than the single material (parametric)
c    settings.
C
C***************
C  SELECT  WALL  MATERIAL  (SINGLE /DATABASE)
C***************
```

```
C
C
     IF (IWALL_TYPE.EQ.1)THEN
       WALL_TYPE_NAME='Wall Material Database'
       IF (IWALL_MATERIAL.EQ.1)THEN
         WALL_MAT_NAME='Aluminum Alloys'
         OPEN(UNIT=35,STATUS='OLD',ACCESS='SEQUENTIAL',
     +     FILE='ALWALL.INP')
         OPEN(UNIT=39,STATUS='OLD',ACCESS='SEQUENTIAL',
     +     FILE='alwall.tbl')
       else IF (IWALL_MATERIAL.EQ.2)THEN
         WALL_MAT_NAME='Advanced Launch System'
         OPEN(UNIT=35,STATUS='OLD',ACCESS='SEQUENTIAL',
     +     FILE='ALSWALL.INP')
         OPEN(UNIT=39,STATUS='OLD',ACCESS='SEQUENTIAL',
     +     FILE='ALSWALL.TBL')
       else
        write(*,*)'Input Error from PSDOC interface - '
        write(*,*)'Program Terminated with internal error.'
        write(*,*)'Bad Iwall_material switch.          '
        goto 11
       end if
C  Read Number of walls in selected table
         Read(35,31)nwall
              DO 5015 KI=1,7
           READ(39,*)
5015        CONTINUE
         DO 637 KI=1,NWALL
           READ(39,6002,END=638)WALL_NAME(KI)
637        CONTINUE
638        CLOSE (UNIT = 39)
6002        format(a11)

     C--------------------
C   SINGLE WALL MATERIAL
          else IF (IWALL_TYPE.EQ.2)THEN
      WALL_TYPE_NAME='Single Wall Material'
      OPEN(UNIT=35,STATUS='OLD',ACCESS='SEQUENTIAL',
     +  FILE='WALL.INP')
      NWALL = 1
    else
      write(*,*)'Input Error from PSDOC interface - '
      write(*,*)'Program Terminated with internal error.'
      write(*,*)'Bad Iwall_type switch.          '
      goto 11
    END IF
```

```
   c  if using Table option, open new table1.out file and write headers
  if ((ibumper_type.eq.1).or.(Iwall_type.eq.1)) then
    OPEN(UNIT=37,STATUS='unknown',ACCESS='SEQUENTIAL',
 +     FILE='TABLE1.OUT')
C   Read headers into table1.out file
    open(unit=33,status='old',access='sequential',
 +     file='table1.hdr')
          write(37, '(1x, A10,1x, I2.2, 1H:, I2.2, 1H:, I2.2, 1H.,
 +I2.2)') ctime, ihr, imin, isec, i100th
     write(37, '(1x, A10,1x, I2.2, 1H-, I2.2, 1H-, I4.2)')
 + cdate, imon, iday, iyr
         do 6005 i = 1,4
     read(33,6007)line1
     write(37,6007)line1
6005   continue
6007   format(a80)
    read(33,6007)line1
    line2 = line1(1:12) // bumper_mat_name // line1(53:80)
    write(37,6007)line2
    read(33,6007)line1
    line2 = line1(1:12) // wall_mat_name // line1(53:80)
    write(37,6007)line2
    do 6006 i=1,7
     read(33,6007)line1
     write(37,6007)line1
 6006  continue
    close(unit=33)
  end if
    C
C***************
C  READ  GEOMETRICAL  SHAPE
C***************
C
    IF (ISW6.EQ.1) THEN
      SHAPE = 'Cylinder'
    else
      write(*,*)'Input Error from PSDOC interface - '
      write(*,*)'Program Terminated with internal error.'
      write(*,*)'Bad Isw6 switch.          '
      goto 11
    END IF

C***************
C  READ IMPACT  MODEL
C***************
C
```

```
          IF (ISW7.EQ.1) THEN
      IMPACT_MODEL='Single Impact Model'
    else IF (ISW7.EQ.2) THEN
      IMPACT_MODEL='Three Impact Regions'
    else
      write(*,*)'Input Error from PSDOC interface - '
      write(*,*)'Program Terminated with internal error.'
      write(*,*)'Bad Isw7 switch.        '
      goto 11
    end if
    IF (ISW8.EQ.1) THEN
      DEBRIS_ENVIRONMENT='Boeing Model'
      NCODE = 2
    else
      write(*,*)'Input Error from PSDOC interface - '
      write(*,*)'Program Terminated with internal error.'
      write(*,*)'Bad Isw8 switch.        '
      goto 11
    END IF
      C----------------
C  SELECT GRAFTOOL OUTPUT FILE FORMAT
C----------------
C
C    Write the definitions and the headings in the temporary output
C    file ( z9aaaaaj.pgm ) in the c:\psdoc\input directory.
c    This format is critical to PSDOC front-end...don't change!
          IF (IGRAPH_TYPE.EQ.1)THEN
        WRITE(27,*)'/ The Output Variables Are Defined As: '
        WRITE(27,*)'/ '
        WRITE(27,*)'/  Column 1:    run = Run-#'
        WRITE(27,*)'/  Column 2:    T1 = Optimal Bumper Thickness'
        WRITE(27,*)'/  Column 3:    T2 = Optimal Wall Thickness'
        WRITE(27,*)'/  Column 4:    OBMPUA = Optimal Bumper Mass'//
   +            ' Per Unit  Area'
        WRITE(27,*)'/  Column 5:   OWMPUA = Optimal Wall Mass'//
   +            ' Per Unit Area'
        WRITE(27,*)'/  Column 6:    WT = Minimum System Mass Per'
   +            //' Unit Area'
        WRITE(27,*)'/  Column 7 :   WTCMC = Mnimum CMC Weight'
        WRITE(27,*)'/  Column 8 :   OBR = Optimal Bumper Ratio'
        WRITE(27,*)'/  Column 9 :   OWR = Optimal Wall Ratio'
        WRITE(27,*)'/  Column 10 :  D = Critical Design Projectile'
   +            //' Diameter'
        WRITE(27,*)'/  Column 11:   RHOP = Projectile Density'
        WRITE(27,*)'/  Column 12:   XGROWTH = Space Debris Growth'
   +            // ' Rate'
```

```
        WRITE(27,*)'/ Column 13:  IMONTH1 = Initial Operation Month'
     WRITE(27,*)'/ Column 14:  IYEAR1 = Initial Operation Year'
     WRITE(27,*)'/ Column 15:  IMONTH2 = Final Operation Month'
     WRITE(27,*)'/ Column 16:  IYEAR2 = Final Operation Year'
      WRITE(27,*)'/ Column 17:  ALT = Spacecraft Orbital'//
  +         ' Altitude'
      WRITE(27,*)'/ Column 18:  XINCL = Spacecraft Orbital'//
  +         ' Inclination'
      WRITE(27,*)'/ Column 19:  XP0 = Spacecraft Probability'//
  +         ' Of No Penetration'
      WRITE(27,*)'/ Column 20:  AREAK = Spacecraft Exposed'//
  +         ' Area'
      WRITE(27,*)'/ Column 21:  S = Spacecraft Bumper / Wall'//
  +         ' Separation'
      WRITE(27,*)'/ '
      WRITE(27,*)'/ '
      WRITE(27,*)'/ '
      WRITE(27,*)'/ Valid  X -  Columns  Are:  '
      WRITE(27,*)'/ 11,16,17,18,19,20  '
      WRITE(27,*)'/ Valid  Y -  Columns Are:  '
      WRITE(27,*)'/  2,3,4,5,6,7,8,9  '
      WRITE(27,*)'/ '
      WRITE(27,*)'/ '
      WRITE(27,*)'/ '
      WRITE(27,776)'/RUN-# ',' T1 ',' T2 ',' OBMPUA ','OWMPUA',
  +    'WT','WTCMC',' OBR ',' OWR ',' D ',' RHOP ','XGROWTH',
  +    'IMONTH1',' IYEAR1',' IMONTH2','IYEAR2'
  +    ,' ALT ',' XINCL ',' XP0 ',' AREAK ',' S '
776    FORMAT(21(A12,1X))
    C--------------------------
C   WRITE THE DEFINITIONS AND THE HEADINGS IN THE OUTPUT
C   FILE ( RESULTS.OUT ) IN THE LOTUS (123) FORMAT
        else IF (IGRAPH_TYPE.EQ.2)THEN
      WRITE(27,*)' " ',' The Output Variables Are'//
  +     ' Defined As: ',' " '
      WRITE(27,*)' " '
      WRITE(27,*)' " ',' T1 = Optimal Bumper Thickness',' " '
      WRITE(27,*)' " ',' T2 = Optimal Wall Thickness',' " '
      WRITE(27,*)' " ',' OBMPUA = Optimal Bumper Mass Per Unit '
  +     //'Area',' " '
      WRITE(27,*)' " ',' OWMPUA = Optimal Wall Mass Per '//
  +     'Unit Area',' " '
      WRITE(27,*)' " ',' WT = Minimum System Mass Per Unit '//
  +     'Area',' " '
      WRITE(27,*)' " ',' WTCMC = Minimum CMC Weight',' " '
      WRITE(27,*)' " ',' OBR = Optimal Bumper RATIO',' " '
      WRITE(27,*)' " ',' OWR = Optimal Wall RATIO',' " '
      WRITE(27,*)' " ',' D = Critical Design Projectile '//
  +     'Diameter',' " '
```

```
      WRITE(27,*)' " ',' RHOP = Projectile Density',' " '
      WRITE(27,*)' " ',' XGROWTH = Space Debris Growth Rate',' " '
       WRITE(27,*)' " ',' IMONTH1 = Initial Operation Month'
       WRITE(27,*)' " ',' IYEAR1 = Initial Operation Year'
       WRITE(27,*)' " ',' IMONTH2 = Final Operation Month'
       WRITE(27,*)' " ',' IYEAR2 = Final Operation Year'
      WRITE(27,*)' " ',' ALT = Spacecraft Orbital Altitude',' " '
      WRITE(27,*)' " ',' XINCL = Spacecraft Orbital '//
     +      'Inclination',' " '
      WRITE(27,*)' " ',' XP0 = Spacecraft Probability Of No '//
     +      'Penetration',' " '
      WRITE(27,*)' " ',' AREAK = Spacecraft Exposed Area',' " '
      WRITE(27,*)' " ',' S = Spacecraft Bumper / Wall '//
     +      'Separation',' " '
      WRITE(27,*)' " " '
      WRITE(27,*)' " " '
      WRITE(27,*)' " " '
       WRITE(27,*)'/ Valid  X -  Columns  Are: '
       WRITE(27,*)'/ 11,16,17,18,19,20  '
       WRITE(27,*)'/ Valid  Y -  Columns Are:  '
       WRITE(27,*)'/  2,3,4,5,6,7,8,9  '
      WRITE(27,*)' " " '
      WRITE(27,*)' " " '
      WRITE(27,*)' " " '
      WRITE(27,777)'"RUN-#"','"T1"','"T2"','"OBMPUA"','"OWMPUA"',
     + '"WT"', '"WTCMC"','"OBR"','"OWR"','"D"','"RHOP"','"XGROWTH"',
     + '"IMONTH1"','"IYEAR1"','"IMONTH2"','"IYEAR2"','"ALT"',
     + '"XINCL"','"XP0"','"AREAK"','"S"'
777    FORMAT(21(A12,1X))
         else
     write(*,*)'Input Error from PSDOC interface - '
     write(*,*)'Program Terminated with internal error.'
     write(*,*)'Bad Igraph switch.          '
     goto 11
         END IF

     C   CALCULTE  THE NUMBER OF MATERIALS " NMATS "
         NMATS = NBUMP * NWALL

     C  ##############################################
C #   MAIN LOOP BEGINS               #
C ##############################################
     C   IN HERE WE ARE INITIALIZING THE COUNTER VARIABLE "I"

          I = 1
    iiii = 0
    write(*,*)
```

```
 20       CONTINUE
       iiii = iiii + 1
        write(*,21)iiii
21      format('+Status - Running Case:  ',I4)
c
c      Reset files 20,23,26 (newdebri.inp, olddebri.inp,
c      ne_met.inp, craft.inp, and geometry.inp)
c      when running in table mode.  If parametric, always rewind 30, 35
c      (bumper.inp, and wall.inp) for proper execution.
              if (ibumper_type.eq.1) then
         rewind(20)
         rewind(23)
         rewind(26)
       else if (ibumper_type.eq.2) then
         rewind(30)
       end if
       if (iwall_type.eq.2) then
         rewind(35)
       end if

              SEEDVAL = 73
       call Seed (SeedVal)
              t1    = 0
       t2   = 0
       obmpua = 0
       owmpua = 0
       wt    = 0
       wtcmc  = 0
       obr   = 0
        C ----------------------
             IF (NCODE.EQ.1) then
C             NYSMITH
        C     PROJECTILE DIAMETER IN CM **** READ(10,*)
       READ(22,*,end=11)D
        C     BUMPER / Wall SEPARATION  **** READ(10,*)H
       READ(23,*,end=11)S
        C     **** READ(10,*)RHO1'
       READ (24,*,end=11)RHO1
        C     **** READ(10,*)RHO2'
       READ (25,*,end=11)RHO2
        C     **** READ(10,*)CMCLEN
       READ (26,*,end=11)CMCLEN
        C     **** READ(10,*)CMCRAD
       READ (26,*,end=11)CMCRAD

        c     WRITE(11,*)'     NYSMITH'
c       WRITE(11,*)
c       WRITE(11,*)'     INPUT'
c       WRITE(11,*)
```

```
c     WRITE(11,*)'     Projectile Diameter In CM = ',D
c     WRITE(11,*)'     Bumper/Wall Separation In CM = ',H
c     WRITE(11,*)'     Bumper/Wall Separation In CM = ',S
c     WRITE(11,*)
         T1T = 0
    T2T = 0

         DO 26 J=1,16
          V = FLOAT(J)
     C     CALL NYSMITH(V,D,H,RHO1,RHO2,T1,T2,WT,WTCMC)
          CALL NYSMITH(V, D, H, RHO1, RHO2, T1, T2)
          T1T = T1T + T1 * XPV(J)
     T2T = T2T + T2 * XPV(J)
     26      CONTINUE

         T1 = T1T
    T2 = T2T
    WT = RHO1 * T1 + RHO2 * T2
    R12 = CMCRAD
    R22 = R12 + T2
    R11 = R22 + H
    R21 = R11 + T1
    WTCMC = 3.1416 * (CMCLEN / 1000.0)
    WTCMC=WTCMC*(RHO1*(R21**2.-R11**2)+RHO2*(R22**2.-R12**2.))
     c     WRITE(11,*)'     OUTPUT'
c     WRITE(11,*)
c     WRITE(11,*)'     Bumper Thickness = ',T1,'CM'
c     WRITE(11,*)'     Wall Thickness = ',T2,'CM'
c     WRITE(11,*)'     Minimum  Weight = ',WT,'GM/Square CM'
c     WRITE(11,*)'     CMC Minimum  Weight = ',WTCMC,'KG'
c     WRITE(11,*)
c     WRITE(11,*)
c     WRITE(11,*)
     C-----------------
        else IF (NCODE.EQ.2) then
C        BOEING
     C    NENVIRON = 1  ==> EARTH ORBITAL SPACE DEBRIS (NEW)
         IF(NENVIRON .EQ. 1) THEN
          READ(20,*,end=11)XGROWTH
          READ(20,*,end=11)x
     IMONTH1 = x
          READ(20,*,end=11)x
     IYEAR1 = x
          READ(20,*,end=11)x
     IMONTH2 = x
          READ(20,*,end=11)x
     IYEAR2 = x
```

```
        READ(20,*,end=11)ALT
    READ(20,*,end=11)XINCL
    READ(20,*,end=11)XP0
    READ(23,*,end=11)AREAK

        INCL = IFIX(XINCL + .5)
    XPSI = XPSIV(INCL)

        CALL DEBRIS(XGROWTH,SOLAR,XPSI,IMONTH1,IYEAR1,IMONTH2,
  +   IYEAR2,ALT,XINCL,XP0,AREAK,D,XPV,IVMAX)

        RHOP = 2.8
         IF(D .GT. 1.0) THEN
      RHOP = 2.8/(D**0.74)
     END IF
       else if (NENVIRON .EQ. 2) THEN
C       NENVIRON = 2 ==>  EARTH ORBITAL SPACE DEBRIS(OLD)

        READ(20,*,end=11)T
    READ(20,*,end=11)XP0
    READ(23,*,end=11)AREAK
        CALL DEBRISOLD(T, XP0, AREAK, D)
        RHOP = 2.8
         DO 555 KIJK=1,16
       XPV(KIJK) = XDEBOLDIV(KIJK)
555      CONTINUE
       else if (nenviron.eq.3) then
C     NENVIRON = 3 ==>  Near Earth Meteoroid
        READ(23,*,end=11)AREAK
    READ(20,*,end=11)T
    READ(20,*,end=11)ALT
    READ(20,*,end=11)XP0

        DENS = .5
        CALL METEOROID(AREAK, T, XP0, ALT, DENS, D, L)

        RHOP = DENS
        IVMAX = 72

         DO 544 KIJK=1,72
           XPV(KIJK) = XMETIV(KIJK)
  544    CONTINUE

   end if
       READ(23,*,end=11)S
   READ(26,*,end=11)CMCLEN
   READ(26,*,end=11)CMCRAD
       READ(30,*,end=11)RHO1
c      RHO1 = C_RHO1(K)
```

```
         READ(30,*,end=11)SY1
c       SY1 = C_SY1(K)
         READ(30,*,end=11)E1
c       E1 = C_E1(K)
         READ(35,*,end=11)RHO2
c       RHO2 = C_RHO2(K)
         READ(35,*,end=11)XL2
c       XL2 = C_XL2(K)
         READ(35,*,end=11)SY2
c        SY2 = C_SY2(K)

         XN = .85
         SY1 = SY1 * 144000.0
      SY2 = SY2 * 144000.0
      E1 = E1 * 6.880285E+10
         T1T = 0.0
      T2T = 0.0
         DO 36 J=1,IVMAX
           V = FLOAT(J)
           IF(XINCL .GT. 40.0)THEN
        THETA = ACOS(-1.0  * V / IVMAX) - 1.57
        IF(THETA .GT. 1.57)THEN
          THETA = 1.57
        END IF
       ELSE
        THETA = ACOS(-1.0  * V / 15.4) - 1.57
       END IF

    C 3676   CALL BOEING(V,D,RHOP,RHO1,RHO2,S,XL2,SY1,SY2,THETA,
C   +           XN,E1,CMCRAD,T1,T2,WT,WTCMC)

     3676   CALL BOEING(V,D,RHOP,RHO1,RHO2,S,XL2,SY1,SY2,THETA,
   +          XN,E1,CMCRAD,T1,T2,WT)

         T1T = T1T + XPV(J) * T1
      T2T = T2T + XPV(J) * T2
     36      CONTINUE
         T1 = T1T
      T2 = T2T
      WT = RHO1 * T1 + RHO2 * T2
      R12 = CMCRAD
      R22 = CMCRAD + T2
      R11 = CMCRAD + T2 + S
      R21 = CMCRAD + T1 + T2 + S
      VB=3.1416*(CMCLEN/1000.)*(R21**2.-R11**2.)
      VW=3.1416*(CMCLEN/1000.)*(R22**2.-R12**2.)
      WTCMC = RHO1 * VB + RHO2 * VW

     991    CONTINUE
```

```
        else IF (NCODE.EQ.3) then
      C          MADDEN
****** MADDEN MINIMIZES SUM OF THICKNESSES ONLY *******

      C   45    READ(10,*)D

      C    PROJECTILE DIAMETER IN CM **** READ(10,*)D
           READ(22,*,end=11)D
      c    READ(10,*)RHOP
      C    ****** (10,*)S
      READ(23,*,end=11)S
      c    READ(10,*)RHO
      c    WRITE(11,*)'    MADDEN'
c     WRITE(11,*)
c     WRITE(11,*)'    INPUT'
c     WRITE(11,*)
c     WRITE(11,*)'    Projectile Diameter In CM = ',D
c     WRITE(11,*)'    Projectile Density In IN GM/Cubic CM = ',RHOP
c     WRITE(11,*)'    Bumper/Wall Density In GM/Cubic CM = ',RHO
c     WRITE(11,*)'    Bumper/Wall Separation In CM = ',S
c     WRITE(11,*)
           T1T = 0.0
      T2T = 0.0

           DO 46 J=1,16
            V = FLOAT(J)
      C     CALL MADDEN(V,D,RHOP,S,RHO,T1,T2,WT,WTCMC)
            CALL MADDEN(V, D, RHOP, S, RHO, T1, T2)
            T1T = T1T + T1 * XPV(J)
       T2T = T2T + T2 * XPV(J)
       46   CONTINUE

           T1 = T1T
      T2 = T2T
      WT = T1 + T2
      R12 = 211.0
      R22 = 211.0  + T2
      R11 = 211.0  + T2 + S
      R21 = 211.0  + T1 + T2 + S
      VB=4.27*(R21**2.-R11**2.)
      VW=4.27*(R22**2.-R12**2.)
      WTCMC = RHO * (VB + VW)
       c    WRITE(11,*)'    OUTPUT'
c     WRITE(11,*)
c     WRITE(11,*)'    Bumper Thickness = ',T1,'CM'
c     WRITE(11,*)'    Wall Thickness = ',T2,'CM'
c     WRITE(11,*)'    Minimum  Weight = ',WT,'CM'
c     WRITE(11,*)'    CMC Minimum  Weight = ',WTCMC
```

```
c     WRITE(11,*)
c     WRITE(11,*)
c     WRITE(11,*)
      C-----------
         else IF (NCODE.EQ.4) then
C        WILKINSON

       C     **** READ(10,*)D
     READ(22,*,end=11)D
       C     **** READ(10,*)RHOP
     READ(22,*,end=11)RHOP
       C     **** (10,*)RHO1
     READ(24,*,end=11)RHO1
       C     **** (10,*)RHO2
     READ(25,*,end=11)RHO2
       C     **** (10,*)S
     READ(23,*,end=11)S
       C     **** (10,*)XL2
     READ(25,*,end=11)XL2
       C     **** (10,*)CMCLEN
     READ(26,*,end=11)CMCLEN
       C     **** (10,*)CMCRAD
     READ(26,*,end=11)CMCRAD


       c     WRITE(11,*)'     WILKINSON'
c     WRITE(11,*)
c     WRITE(11,*)'     INPUT'
c     WRITE(11,*)
c     WRITE(11,*)'     Projectile Diameter In CM = ',D
c     WRITE(11,*)'     Projectile Density In GM/Cubic CM = ',RHOP
c     WRITE(11,*)'     Bumper Density In GM/Cubic CM = ',RHO1
c     WRITE(11,*)'     Wall Density In GM/Cubic CM = ',RHO2
c     WRITE(11,*)'     Bumper/Wall Separation In CM = ',S
c     WRITE(11,*)'     Wall Material Constant = ',XL2
c     WRITE(11,*)
         T1T = 0.0
     T2T = 0.0

        DO 56 J=1,16
         V = FLOAT(J)

       C      CALL WILKINSON(V,D,RHOP,RHO1,RHO2,S,XL2,
C   &          T1,T2,WT,WTCMC)

            CALL WILKINSON(V,D,RHOP,RHO1,RHO2,S,XL2,
  &           T1,T2)
```

```
        T1T = T1T + T1 * XPV(J)
    T2T = T2T + T2 * XPV(J)
    56    CONTINUE

       T1 = T1T
   T2 = T2T
   WT = RHO1 * T1 + RHO2 * T2
   R12 = CMCRAD
   R22 = CMCRAD + T2
   R11 = CMCRAD + T2 + S
   R21 = CMCRAD + T1 + T2 + S
   VB=3.1416*(CMCLEN/1000.)*(R21**2.-R11**2.)
   VW=3.1416*(CMCLEN/1000.)*(R22**2.-R12**2.)
   WTCMC = RHO1 * VB + RHO2 * VW
    c     WRITE(11,*)'     OUTPUT'
c   WRITE(11,*)
c   WRITE(11,*)'      Bumper Thickness = ',T1,'CM'
c   WRITE(11,*)'      Wall Thickness = ',T2,'CM'
c   WRITE(11,*)'      Minimum  Weight = ',WT,'GM/Square CM'
c   WRITE(11,*)'      CMC Minimum  Weight = ',WTCMC,'KG'
c   WRITE(11,*)
c   WRITE(11,*)
c   WRITE(11,*)

    C-----------
       else IF (NCODE.EQ.5) then
C     MODIFIED BURCH

    C     **** READ(10,*)D
   READ(22,*,end=11)D
    C     **** (10,*)RHO1
   READ(24,*,end=11)RHO1
    C     **** (10,*)RHO2
 • READ(25,*,end=11)RHO2
    C     **** (10,*)S
   READ(23,*,end=11)S
    C     **** READ(10,*)THETA
   READ(22,*,end=11)THETA
    C     **** READ(10,*)XN
   READ(24,*,end=11)XN
    C     **** (10,*)E1
   READ(24,*,end=11)E1
    C     **** (10,*)CMCLEN
   READ(26,*,end=11)CMCLEN
    C     **** (10,*)CMCRAD
   READ(26,*,end=11)CMCRAD
```

```
  c     WRITE(11,*)'   MODIFIED BURCH'
c     WRITE(11,*)
c     WRITE(11,*)'   INPUT'
c     WRITE(11,*)
c     WRITE(11,*)'   Projectile Diameter In CM = ',D
c     WRITE(11,*)'   Bumper Density In  GM/Cubic CM = ',RHO1
c     WRITE(11,*)'   Bumper/Wall Separation In CM = ',S
c     WRITE(11,*)'   Impact Angle From Normal In Degrees = ',THETA
c     WRITE(11,*)'   Number Of Plates To Penetrate After First',
c    + ' Bumper = ', XN
c     WRITE(11,*)'   Bumper Youngs Modulus In MSI = ', E1
c     WRITE(11,*)
          E1 = E1 * 6.880285E+10
     T1T = 0.0
     T2T = 0.0

          DO 66 J=1,16
           V = FLOAT(J)
  C         CALL BURCH(V,D,RHO1,RHO2,S,THETA,
C   &          XN,E1,T1,T2,WT,WTCMC)

           CALL BURCH(V,D,RHO1,RHO2,S,THETA,
   &          XN,E1,T1,T2,T1B,F1)

           T1T = T1T + T1 * XPV(J)
      T2T = T2T + T2 * XPV(J)

      66    CONTINUE

          T1 = T1T
     T2 = T2T
     WT = RHO1 * T1 + RHO2 * T2
     R12 = CMCRAD
     R22 = CMCRAD + T2
     R11 = CMCRAD + T2 + S
     R21 = CMCRAD + T1 + T2 + S
     VB=3.1416*(CMCLEN/1000.)*(R21**2.-R11**2.)
     VW=3.1416*(CMCLEN/1000.)*(R22**2.-R12**2.)
     WTCMC = RHO1 * VB + RHO2 * VW
  c     WRITE(11,*)'    OUTPUT'
c     WRITE(11,*)
c     WRITE(11,*)'    Bumper Thickness = ',T1,'CM'
c     WRITE(11,*)'    Wall Thickness = ',T2,'CM'
c     WRITE(11,*)'    Minimum  Weight = ',WT,'GM/Square CM'
c     WRITE(11,*)'    CMC Minimum  Weight = ',WTCMC,'KG'
c     WRITE(11,*)
c     WRITE(11,*)
c     WRITE(11,*)
```

```
      C -------------
          end if
      C   HERE WE DEFINE AND CALCULATE NEW VARIABLES
C     NEEDED FOR OUTPUT
      C   OPTIMAL BUMPER MASS PER UNIT AREA
            OBMPUA = T1 * RHO1
      C   OPTIMAL WALL MASS PER UNIT AREA
            OWMPUA = T2 * RHO2
      C   OPTIMAL BUMPER RATIO
            OBR = T1 * RHO1 / WT
      C   OPTIMAL WALL RATIO
            OWR = T2 * RHO2 / WT
      C   SPACECRAFT INITIAL OPERATING CAPABILITY
            SIOC = IYEAR1
      C   SPACECRAFT MISSION DURATION
            SMD = IYEAR2 - IYEAR1 + 1

      C ------------
C ------------
C     HERE WE  WRITE THE CALCULATED OUTPUT VALUES
C     TO THE MAIN OUPUT FILE CALLED 'RESULTS.OUT'
C
      if ((ibumper_type.eq.1).or.(iwall_type.eq.1)) then
      C   Write Calculated Output Values To ' TABLE1.OUT '
             WRITE(37,782) BUMPER_NAME(iiii),WALL_NAME(iiii),T1,
     +          T2,OBMPUA,OWMPUA,WT,WTCMC
782     FORMAT((A14,1X),(A11,1X),(5F7.4,1X),F9.2)

      else
            IF (IGRAPH_TYPE.EQ.1)THEN
       778  WRITE(27,779)I,T1,T2,OBMPUA,OWMPUA,WT,WTCMC,OBR,OWR,D,RHOP,
     +  XGROWTH,IMONTH1,IYEAR1,IMONTH2,IYEAR2,ALT,XINCL,XP0,AREAK,S
            WRITE(29,779)I,T1,T2,OBMPUA,OWMPUA,WT,WTCMC,OBR,OWR,D,RHOP,
     +  XGROWTH,IMONTH1,IYEAR1,IMONTH2,IYEAR2,ALT,XINCL,XP0,AREAK,S
       779    FORMAT(I9,1X,11(F12.4,1X),4(I9,1X),5(F12.4,1X))
            else if (igraph_type.eq.2) then
       780    WRITE(27,781)I,',',T1,',',T2,',',OBMPUA,',',OWMPUA,',',
     +  WT,',',WTCMC,',',OBR,',',OWR,',',D,',',RHOP,',',
     +  XGROWTH,',',IMONTH1,',',IYEAR1,',',IMONTH2,',',IYEAR2,',',
     +  ALT,',',XINCL,',',XP0,',',AREAK,',',S
            WRITE(29,779)I,T1,T2,OBMPUA,OWMPUA,WT,WTCMC,OBR,OWR,D,RHOP,
     +  XGROWTH,IMONTH1,IYEAR1,IMONTH2,IYEAR2,ALT,XINCL,XP0,AREAK,S
       781    FORMAT(I9,1X,11(F12.4,1X),4(I9,1X),5(F12.4,1X))
            end if
      end if
          I = I + 1
       10   GOTO 20
```

```
      11   CONTINUE
      C       HERE WE WRITE THE CALCULATED VALUES OF  "V",
C         "XPV(V)", AND "THETA" TO THE OUTPUT FILE CALLED
C         "PROJECT.OUT" prior to leaving
              WRITE(28,3675)V,XPV(V),THETA
3675   FORMAT(3(F9.5,1X))

            write (*,*) ' Program Finished'
   CLOSE (UNIT = 20)
   CLOSE (UNIT = 23)
   CLOSE (UNIT = 26)
   CLOSE (UNIT = 27)
   CLOSE (UNIT = 28)
   CLOSE (UNIT = 29)
   CLOSE (UNIT = 30)
   CLOSE (UNIT = 35)
   CLOSE (UNIT = 37)
            STOP
   END
      C ------------
      C*********************
      C ------------
C
C         SUBROUTINES BEGIN HERE
C
C ------------
            SUBROUTINE DEBRIS(XGROWTH,SOLAR,XPSI,IMONTH1,IYEAR1,IMONTH2,
   &    IYEAR2,ALT,XINCL,XP0,AREAK,D,XPV,IVMAX)

      C      DIMENSION SOLAR(100),XPV(100),XPSIV(105) <--- MODIFIED
              DIMENSION SOLAR(1188), XPV(100)

              G1TOT = 0.0
     G2TOT = 0.0
     NYEARS = IYEAR2 - IYEAR1 + 1
     NMONTHS = 12*(IYEAR2-IYEAR1)+imonth2-imonth1

     DO 582 IL=1,NMONTHS
       XPHI1=10.**((ALT/200.)-
   &          (SOLAR(12*(IYEAR1-1987-1)+imonth1+IL-1)/140.)-1.5)
       XPHI = XPHI1 / (XPHI1 + 1.0 )
       G1=(1.+2.*XGROWTH)**(IYEAR1-1985+(imonth1+il-2)/12.0)
       G2=(1.+XGROWTH)**(IYEAR1-1985+(imonth1+il-2)/12.0)
       G1TOT = G1TOT + XPHI * G1
       G2TOT = G2TOT + XPHI * G2
582    CONTINUE
```

```
          FLUX = 12.0 * ALOG(XP0) / (AREAK * XPSI)
      DEN = -1.0  * (5.9499E-07 * G2TOT + FLUX)
      XNUM = .0000105 * G1TOT
      D=(XNUM/DEN)**0.4
      YG = 250.0
      YF = 0.0
      YC = .0125
      YE = .55 + .005 * (XINCL - 30.0 )
      YH=1.0-0.0000757*(XINCL-60.0)**2.0
      YA = 2.5
      YB = .3
      YD = 1.3 - .01 * (XINCL - 30.0 )
      YV0 = 7.7
          IF(XINCL .LE. 60.0)THEN
       YB = .5
       YG = 18.7
       YV0 = 7.25 + .015 * (XINCL - 30.0 )
      END IF
          IF(XINCL .LE. 80.0 .AND. XINCL .GT. 60.0)THEN
       YB = .5 - .01 * (XINCL - 60.0 )
       YG=18.7+0.0289*(XINCL-60.0)**3.0
      END IF
          IF(XINCL .GT. 100.0)THEN
       YC = .0125 + .00125 * (XINCL - 100.0 )
      END IF
          IF(XINCL .LE. 50.0)THEN
       YF=0.3+0.0008*(XINCL-50.0)**2.0
      END IF
          IF(XINCL .GT. 50.0 .AND. XINCL .LE. 80.0)THEN
       YF = .3 - .01 * (XINCL - 50.0 )
      END IF
          XSUMIV = 0.0
      IVMAX = 1
      IV = 1
584    XPV(IV)=YG*2.7183**(-1.0*((IV-YA*YV0)/(YB*YV0))**2.0)
      XPV(IV)=XPV(IV)+YF*2.7183**(-1.0*((IV-YD*YV0)/(YE*YV0))**2.0)
      XPV(IV)=XPV(IV)*(2.0*IV*YV0-IV**2.0)
      XPV(IV)=XPV(IV)+YH*YC*(4.0*IV*YV0-IV**2.0)
          IF(XPV(IV) .LE. 0.000)THEN
       XPV(IV) = 0.0
       IVMAX = IV
       GOTO 586
      END IF
          XSUMIV = XSUMIV + XPV(IV)
      IV = IV + 1
      GOTO 584
       586    DO 588 I=1,IVMAX
        XPV(I) = XPV(I) / XSUMIV
588     CONTINUE
```

```
      RETURN
    END
     C ------------
      SUBROUTINE DEBRISOLD(T, XP0, AREAK, D)
      FLUX = -1.0  * ALOG(XP0) / (AREAK * T)
     F = ALOG10(FLUX)
     C***** MS-FORTRAN DOES NOT ALLOW CONSECUTIVE MATHEMATICAL
C    OPERATORS TO BE PLACED ADJACENT TO ONE ANOTHER, i.e.
C    YOU CAN *** NOT *** HAVE:  " /- "
C
C
C      IF(F.GE.-5.46)THEN
C       D=10.**((F+5.46)/-2.52)
C      END IF
C      IF(F.GE.-5.9.AND.F.LT.-5.46)THEN
C       D=10.**((F+5.02)/-0.44)
C      END IF
C      IF(F.LT.-5.9.AND.F.GE.-6.4)THEN
C       D=10.**((F+5.78)/-0.063)
C      END IF
C      IF(F.GE.-7.0.AND.F.LT.-6.4)THEN
C       D=10.**((F+6.33)/-0.0067)
C      END IF
C      IF(F.GE.-7.3.AND.F.LT.-7.0)THEN
C       D=10.**((F+6.88)/-0.0012)
C      END IF
C      IF(F.GE.-7.6.AND.F.LT.-7.3)THEN
C       D=10.**((F+6.6)/-0.002)
C      END IF
C      IF(F.GE.-8.0.AND.F.LT.-7.6)THEN
C       D=10.**((F+5.6)/-0.004)
C      END IF
C
        IF (F.GE. - 5.46) THEN
      D=10.**((F+5.46)/(-2.52))
     END IF
        IF(F.GE.-5.9.AND.F.LT.-5.46)THEN
      D=10.**((F+5.02)/(-0.44))
     END IF
        IF(F.LT.-5.9.AND.F.GE.-6.4)THEN
      D=10.**((F+5.78)/(-0.063))
     END IF
        IF(F.GE.-7.0.AND.F.LT.-6.4)THEN
      D=10.**((F+6.33)/(-0.0067))
     END IF
        IF(F.GE.-7.3.AND.F.LT.-7.0)THEN
      D=10.**((F+6.88)/(-0.0012))
     END IF
```

```
        IF(F.GE.-7.6.AND.F.LT.-7.3)THEN
  D=10.**((F+6.6)/(-0.002))
 END IF
        IF(F.GE.-8.0.AND.F.LT.-7.6)THEN
  D=10.**((F+5.6)/(-0.004))
 END IF
       RETURN
 END
  C -----------
  C      SUBROUTINE METEOROID(SA,T,PO,ALT,DENS,D)
        SUBROUTINE METEOROID(AREAK, T, XP0, ALT, DENS, D, L)


        T = 31536000.0 * T
 FLUX = -1.0  * ALOG(XP0) / (AREAK * T)
 RA = 6371.0  / (6371.0  + ALT)
 GE = .568 + .432 * RA
 THETA = ATAN(6371.0  / SQRT(ALT * (ALT + 2.0  * 6371.0 )))
 S = (1.0  + COS(THETA)) / 2.0
 FLUX = FLUX / (GE * S)
 F = ALOG10(FLUX)
        IF (F.GE. - 4.403) THEN
c       WRITE(11,*)'     MASS IS TOO SMALL'
  GOTO 1001
 END IF
        IF(F.GT.-7.103.AND.F.LT.-4.403)THEN
  RAD = 2.509 - .25 * (14.339 + L)
  XM=10.**((-1.584+SQRT(RAD))/.125)
 END IF
        IF(F.LE.-7.103.AND.F.GE.-14.37)THEN
C       XM=10.**((14.37+F)/-1.213)  <--- NOT ALLOWED IN MS-FORTRAN
  XM=10.**( (14.37+F)/(-1.213) )
 END IF
        IF (F.LT. - 14.37) THEN
c       WRITE(11,*)'     MASS IS TOO LARGE'
  GOTO 1001
 END IF
        D=(1.91*XM/DENS)**.333
1001    CONTINUE
        RETURN
 END
  C-------------
  C      SUBROUTINE NYSMITH(V,D,H,RHO1,RHO2,T1,T2,WT,WTCMC)
        SUBROUTINE NYSMITH(V, D, H, RHO1, RHO2, T1, T2)

  C      DMAX=0.24*H*V**-0.2  <--- NOT ALLOWED IN MS-FORTRAN
        DMAX=0.24*H*V**(-0.2)
```

```
          IF (D.GT.DMAX) THEN
c       WRITE(11,*)'   NO SOLUTION--PROJ. DIA. TOO LARGE FOR NYSMITH'
      else
       T1=(1.93*V**0.18*D**1.91/H**0.91)*((RHO2/RHO1)**0.65)
       T2 = 1.86 * T1 * RHO1 / RHO2
      END IF
          RETURN
      END
       C-------------

       ***** PEN4 *****
       C       SUBROUTINE BOEING(V,D,RHOP,RHO1,RHO2,S,XL2,SY1,SY2,THETA,
C    &           XN,E1,CMCRAD,T1,T2,WT,WTCMC)

          SUBROUTINE BOEING(V,D,RHOP,RHO1,RHO2,S,XL2,SY1,SY2,THETA,
   +        XN,E1,CMCRAD,T1,T2,WT)
          T1 = .16
      V = V * 3280.0
      D = D / 30.48
      RP = D / 2.0
      RHOP = RHOP * 1.94
      RHO1 = RHO1 * 1.94
      RHO2 = RHO2 * 1.94
      NITSP = 0
      NITSP = NITSP + 1
      NP1 = 0
      T1P = T1 / 30.48
      T2P = FT2P(RHOP,V, RP, SY1,THETA, RHO1,SY2,D,RHO2,T1P)
      WT = RHO1 * T1P + RHO2 * T2P
          IF (NITSP.EQ.1) THEN
       T1P1 = 1.1 * T1P
       T2P1 = FT2P(RHOP,V,RP,SY1,THETA,RHO1,SY2,D,RHO2,T1P1)
       WT1 = RHO1 * T1P1 + RHO2 * T2P1
      END IF
          IF (WT1.GT.WT) THEN
           T1P1 = .82 * T1P1
       T2P1 = FT2P(RHOP,V,RP,SY1,THETA,RHO1,SY2,D,RHO2,T1P1)
       WT1 = RHO1 * T1P1 + RHO2 * T2P1
       590      IF (WT1.GT.WT) THEN
            GOTO 601
       ELSE
        T1P = T1P1
        T2P = T2P1
        WT = WT1
        NP1 = NP1 + 1
            IF (NP1.EQ.100) THEN
c***        WRITE(11,*)'   NO CONVERGENCE IN PEN4'
         GOTO 557
        END IF
```

```
        T1P1 = .9 * T1P1
   T2P1 = FT2P(RHOP,V,RP,SY1,THETA,RHO1,SY2,D,RHO2,T1P1)
   WT1 = RHO1 * T1P1 + RHO2 * T2P1
   GOTO 590
  END IF
      ELSE
  579     T1P = T1P1
  T2P = T2P1
  WT = WT1
  T1P1 = 1.1 * T1P1
  T2P1 = FT2P(RHOP,V,RP,SY1,THETA, RHO1,SY2,D,RHO2,T1P1)
  WT1 = RHO1 * T1P1 + RHO2 * T2P1
       IF (WT1.GT.WT) THEN
   GOTO 601
  ELSE
   NP1 = NP1 + 1
        IF (NP1.EQ.100) THEN
c***    WRITE(11,*)'   NO CONVERGENCE IN PEN4'
    GOTO 557
   END IF
        GOTO 579
  END IF
      END IF
  601     CONTINUE
      D = 30.48 * D
 RHOP = RHOP / 1.94
 RHO1 = RHO1 / 1.94
 RHO2 = RHO2 / 1.94
 T1P = 30.48 * T1P
 T2P = 30.48 * T2P
 IF(T1P/D.LE.0.4)VF=4100
 IF(T1P/D.GT.0.4)VF=4986*(T1P/D)**0.21
 VF = VF + 4000.0
      IF (V.LE.VF) THEN
c***  WRITE(11,*)'   INSIDE OF PEN4 LIMITS'
  T1 = T1P
  T2 = T2P
  GOTO 1102
 END IF
  557     CONTINUE

  ***** WILKINSON *****
      V = V / 3280.0
 T1=0.604*D**2.*RHOP/(S*RHO1)
 T1 = T1 * SQRT(V * COS(THETA) / XL2)
 T2 = T1 * RHO1 / RHO2
 RATIO = D * RHOP / (T1 * RHO1)
 IF (RATIO.GT.1.0) GOTO 1458
 IF(RATIO.LE.1.0)T2=T2/RATIO
```

```
 1458   CONTINUE
 c***      WRITE(11,*)'T1W = ',T1,'T2W = ',T2
 ***** MODIFIED BURCH *****
        THI=0.816*(0.5236*RHOP*D**3.0)**0.352*(RHOP**0.167)
      THI=THI*(V**0.875)/(0.8467*RHO1**0.5)
      THI = THI / 2.54
      TLO = 0.0
      TINTERVAL = THI - TLO
       C
C
C      ISEED=91411   <--- THIS IS NO LONGER NEEDED, SINCE
C                  WE WILL USE OUR OWN SEED VALUE
C                  WHICH WILL COMPLY WITH MS-FORTRAN

        VB = V * 3280.0
      DB = D / 2.54
      CM = SQRT(E1 / RHO1)
      CM = CM / 30.48
      SB = S / 2.54
      RHOP = RHOP * .036215
      RHO1 = RHO1 * .036215
      RHO2 = RHO2 * .036215
        IF (THETA.LE.0.001) GOTO 125
        CHI = TAN(THETA) - .5
      XPENALTY = 1.0
      T1B = THI
      F1=2.42*(DB/T1B)**0.333+4.26*(T1B/DB)**0.333-4.18
      F2 = .5 - 1.87 * (T1B / DB) + (5.0  * T1B / DB - 1.6)
     + * CHI * CHI * CHI
      F2 = F2 + (1.7 - 12.0  * T1B / DB) * CHI
      F3=0.32*(T1B/DB)**0.83
      F3=F3+0.48*(T1B/DB)**0.33*(SIN(THETA))**3.0
c***      WRITE(11,*)'DB = ',DB,'XN = ',XN,'F1 = ',F1,'F2 = ',F2
c***      WRITE(11,*)'THI = ',THI,'CHI = ',CHI
        IF (F1 + .63 * F2.LT.0.001) THEN
       T2F = 2116.8 * CMCRAD / SY2
       GOTO 483
      END IF
        T2F=DB*((F1+0.63*F2)/XN)**1.7143
      T2F=T2F*(CM/VB)**2.2857
      T2F=T2F*(DB/SB)**0.7143
 483   XNN=F3*(DB/T2F)*(CM/VB)**1.333
      YDELTA = 0.0
        IF(XNN.GT.0.850)YDELTA=1.000
        TOTPEN=YDELTA*XPENALTY*(XNN-0.85)**2.00
      WTB = RHO1 * T1B + RHO2 * T2F + TOTPEN
      WTMIN = WTB
```

```
      T1BEST = THI
      T2BEST = T2F
      TOTPENBEST = TOTPEN
       482   DO 481 IPENALTY=1,460
       C
C          T1B=TINTERVAL*RAN(ISEED)  <--- RAN IS NOT USED IN
C MS - FORTRAN

      CALL RANDOM(RANVAL)
            T1B = TINTERVAL * RANVAL
            F1=2.42*(DB/T1B)**0.333+4.26*(T1B/DB)**0.333-4.18
      F2 = .5 - 1.87 * (T1B / DB) + (5.0  * T1B / DB - 1.6)
     +  * CHI * CHI * CHI
      F2 = F2 + (1.7 - 12.0  * T1B / DB) * CHI
      F3=0.32*(T1B/DB)**0.83
      F3=F3+0.48*(T1B/DB)**0.33*(SIN(THETA))**3.0
            IF (F1 + .63 * F2.LT.0.001) THEN
       T2F = 2116.8 * CMCRAD / SY2
       GOTO 484
      END IF
            T2F=DB*((F1+0.63*F2)/XN)**1.7143
      T2F=T2F*(CM/VB)**2.2857
      T2F=T2F*(DB/SB)**0.7143
484   XNN=F3*(DB/T2F)*(CM/VB)**1.333
      YDELTA = 0.0
            IF(XNN.GT.0.850)YDELTA=1.000
             TOTPEN=YDELTA*XPENALTY*(XNN-0.85)**2.00
       WTB = RHO1 * T1B + RHO2 * T2F + TOTPEN
            IF (WTB.LT.WTMIN) THEN
       WTMIN = WTB
       T1BEST = T1B
       T2BEST = T2F
       TOTPENBEST = TOTPEN
      END IF
       481    CONTINUE
            IF (TOTPENBEST.GT.0.001) THEN
       XPENALTY = XPENALTY * 10.0
       IF (XPENALTY.GT.1.0E12) THEN
       GOTO 485
       END IF
       GOTO 482
      END IF
       485    T1B = T1BEST
      T2B = T2BEST
       c***       WRITE(11,*)'T1B = ',T1B,'T2B = ',T2B,'K = ',XPENALTY
c*****      WRITE(11,*)'TOTPENBEST = ',TOTPENBEST
            GOTO 499
       125    CONTINUE
```

```
   c*****      WRITE(11,*)'RHO1 = ',RHO1
c*****      WRITE(11,*)'RHO2 = ',RHO2

        XK1=(DB/XN)**1.71*(CM/VB)**2.29/SB**0.71
    VDELTA = 0.0
    DELTA3 = .52
1099  DELTA2 = 2.33 * (1.0  - 1.57 * DELTA3)
    DELTA1 = 1.33 * (2.0  * DELTA3 - 1.0 )
    VDELTA1=(1./DELTA1)**DELTA1*(2.8*XK1/(DELTA2*DB**0.57))
   + **DELTA2
    VDELTA1=VDELTA1*(1.58*XK1*DB**0.57/DELTA3)**DELTA3
    VDELTA1=VDELTA1*(RHO1**DELTA1)*(RHO2**(DELTA2+DELTA3))
        IF (VDELTA1.LT.VDELTA) THEN
     DELTA1 = 1.33 * (2.0  * DELTA3 - 1.04)
     T1B = DELTA1 * VDELTA / RHO1
     T2B = (VDELTA - T1B * RHO1) / RHO2
     GOTO 499
    END IF
        VDELTA = VDELTA1
    DELTA3 = DELTA3 + .02
        IF (DELTA3.GT.0.63) THEN
     T1B = DELTA1 * VDELTA / RHO1
     T2B = (VDELTA - T1B * RHO1) / RHO2
     GOTO 499
    END IF
        GOTO 1099
     499    CONTINUE
     ***** COMPARISON OF MODIFIED BURCH AND WILKINSON  *****
     199    CONTINUE
        T10W = T1 / 2.54
    IF (THETA.LT.0.001) GOTO 486
    F10W=2.42*(DB/T10W)**0.333+4.26*(T10W/DB)**0.333-4.18
    F20W = .5 - 1.87 * (T10W / DB) + (5.0  * T10W / DB - 1.6)
   + * CHI * CHI * CHI
    F20W = F20W + (1.7 - 12.0  * T10W / DB) * CHI
    F30W=0.32*(T10W/DB)**0.83
    F30W=F30W+0.48*(T10W/DB)**0.33*(SIN(THETA))**3.0
        IF (F10W + .63 * F20W.LT.0.001) THEN
     T2FT10W = 2116.8 * CMCRAD / SY2
     GOTO 487
    END IF
        T2FT10W=DB*((F10W+0.63*F20W)/XN)**1.7143
    T2FT10W=T2FT10W*(CM/VB)**2.2857
    T2FT10W=T2FT10W*(DB/SB)**0.7143
487   T2BT10W = T2FT10W * 2.54
    XNNT10W=F30W*(DB/T2FT10W)*(CM/VB)**1.333
```

```
          IF (XNNT10W.GT.0.85) THEN
    T2BT10W = 0.0
   END IF
          RATIOB = (DB * RHOP) / (RHO1 * T1B)
   T2WT10B=0.364*D**3.*RHOP*V*COS(THETA)/(XL2*RHO2*S**2.)
          IF(RATIOB.GT.1.0)T2WT10B=T2WT10B*RATIOB
          IF(T2BT10W.GT.T2)T2=T2BT10W
          T2B = T2B * 2.54
          IF(T2WT10B.GT.T2B)T2B=T2WT10B
          T1B = T1B * 2.54
   RHOP = RHOP / .036215
   RHO1 = RHO1 / .036215
   RHO2 = RHO2 / .036215
          IF (RHO1 * T1B + RHO2 * T2B.LT.RHO1 * T1 + RHO2 * T2) THEN
    T1 = T1B
    T2 = T2B
   END IF
          GOTO 155
    486   F10W=1.58*(DB/T10W)**0.57+2.80*(T10W/DB)**0.57
   T2BT10W=(F10W/XN)**1.71*(CM/VB)**2.29*DB**1.71
   T2BT10W=T2BT10W/SB**0.71
   T2BT10W = T2BT10W * 2.54
   RATIOB = (DB * RHOP) / (RHO1 * T1B)
   T2WT10B=0.364*D**3.*RHOP*V*COS(THETA)/(XL2*RHO2*S**2.)
          IF(RATIOB.GT.1.0)T2WT10B=T2WT10B*RATIOB
          IF(T2BT10W.GT.T2)T2=T2BT10W
          T2B = T2B * 2.54
          IF(T2WT10B.GT.T2B)T2B=T2WT10B
          T1B = T1B * 2.54
   RHOP = RHOP / .036215
   RHO1 = RHO1 / .036215
   RHO2 = RHO2 / .036215
          IF (RHO1 * T1B + RHO2 * T2B.LT.RHO1 * T1 + RHO2 * T2) THEN
    T1 = T1B
    T2 = T2B
   END IF
    155    CONTINUE
    1102     IF (T2.LE.0.01) THEN
           T2 = 2116.8 * CMCRAD / SY2
c*****       WRITE(11,*)'T1P = ',T1,'T2P = ',T2
    END IF
    c***      WRITE(11,*)'T1 = ',T1,'T2 = ',T2
    156   RETURN
   END
    C ------------
          FUNCTION FT2B (DB, T1B, XN, CM, VB, SB)
          F1=2.42*(DB/T1B)**0.33+4.26*(T1B/DB)**0.33
   F1 = F1 - 4.18
   FT2B=(F1/XN)**1.71*(CM/VB)**2.29*DB**1.71/SB**0.71
```

```
      RETURN
   END
    C ------------
      FUNCTION FT2P (RHOP,V,RP,SY1,THETA,RHO1,SY2,D,RHO2,T1P)
      A=1.33*RHOP*(V*RP)**2.
   B = 8.0  * SY1 * EXP(-.0003125 * V) / COS(THETA)
   C=1.33*RHOP*RP**2.0
   D1 = RP * RHO1 / COS(THETA)
   XK1=1.67*(RHOP/(2.*SY2))**0.31
   XK1=XK1*(0.281*D*RHOP/RHO2)**0.33
   XK1 = XK1 * COS(THETA)
   C1P1 = (A - B * T1P) / (C + D1 * T1P)
      IF (C1P1.LE.0.001) THEN
    FT2P = 0.0
    GOTO 999
   END IF
      FT2P=XK1*C1P1**0.31
    999   RETURN
   END

    C ------------
    C      SUBROUTINE MADDEN(V,D,RHOP,S,RHO,T1,T2,WT,WTCMC)
      SUBROUTINE MADDEN(V, D, RHOP, S, RHO, T1, T2)

      V = V * 100000.0
   T1=0.009*SQRT(V)*RHOP*D**2.0
   T1=T1/(S*RHO**1.5)
   T2 = T1
      RETURN
   END
    C ------------
    C      SUBROUTINE WILKINSON(V,D,RHOP,RHO1,RHO2,S,XL2,
C   &          T1,T2,WT,WTCMC)
      SUBROUTINE WILKINSON(V,D,RHOP,RHO1,RHO2,S,XL2,
  &          T1,T2)

      T1=0.604*D**2.*RHOP/(S*RHO1)
   T1 = T1 * SQRT(V / XL2)
   T2 = T1 * RHO1 / RHO2
   RATIO = D * RHOP / (T1 * RHO1)
      IF (RATIO.GT.1.0) GOTO 3683
      IF(RATIO.LE.1.0)T2=T2/RATIO
    3683   CONTINUE
      RETURN
   END

    C ------------
    *****  MODIFIED BURCH  *****
```

```
      C       SUBROUTINE BURCH(V,D,RHO1,RHO2,S,THETA,
C     &          XN,E1,T1,T2,WT,WTCMC)

              SUBROUTINE BURCH(V,D,RHO1,RHO2,S,THETA,
     &          XN,E1,T1,T2,T1B,F1)

              VB = V * 3280.0
        DB = D / 2.54
        CM = SQRT(E1 / RHO1)
        CM = CM / 30.48
        SB = S / 2.54
              IF (THETA.LE.0.001) GOTO 425
              CHI = TAN(THETA) - .5
        F2=0.5-1.87*(T1B/D)+(5.*T1B/D-1.6)*CHI**3.0
        F2 = F2 + (1.7 - 12.0  * T1B / D) * CHI
        F3=0.32*(T1B/D)**0.83
        F3=F3+0.48*(T1B/D)**0.33*(SIN(THETA))**3.0
        T2F=D*((F1+0.63*F2)/XN)*(CM/V)**2.29
        T2F=T2F*(D/S)**0.71
        T2N=F3*(CM/V)**1.33*D/XN
              IF(T2N.GE.T2F)T2B=T2N
              IF(T2N.LT.T2F)T2B=T2F
              T2B = T2B * 2.54
              IF(T2B.GT.T2)NREGION=3
              IF(T2B.GT.T2)T2=T2B
              GOTO 499
         425    CONTINUE
              NITSB = 0
        XK1=(DB/XN)**1.71*(CM/VB)**2.29/SB**0.71
        VDELTA = 0.0
        DELTA3 = .52
1099    DELTA2=2.33*(1.-1.57*DELTA3)
        DELTA1 = 1.33 * (2.0  * DELTA3 - 1.0 )
        VDELTA1=(1./DELTA1)**DELTA1*(2.8*XK1/(DELTA2*DB**0.57))
     + **DELTA2
        VDELTA1=VDELTA1*(1.58*XK1*DB**0.57/DELTA3)**DELTA3
        VDELTA1=VDELTA1*(RHO1**DELTA1)*(RHO2**(DELTA2+DELTA3))
              IF (VDELTA1.LT.VDELTA) THEN
         DELTA1 = 1.33 * (2.0  * DELTA3 - 1.04)
         T1 = DELTA1 * VDELTA / RHO1
         T2 = (VDELTA - T1 * RHO1) / RHO2
         GOTO 499
        END IF
              VDELTA = VDELTA1
        DELTA3 = DELTA3 + .02
```

```
      IF (DELTA3.GT.0.63) THEN
 T1 = DELTA1 * VDELTA / RHO1
 T2 = (VDELTA - T1 * RHO1) / RHO2
 GOTO 499
END IF
      GOTO 1099
 499    CONTINUE
      T1 = T1 * 2.54
T2 = T2 * 2.54
      RETURN
     END
```