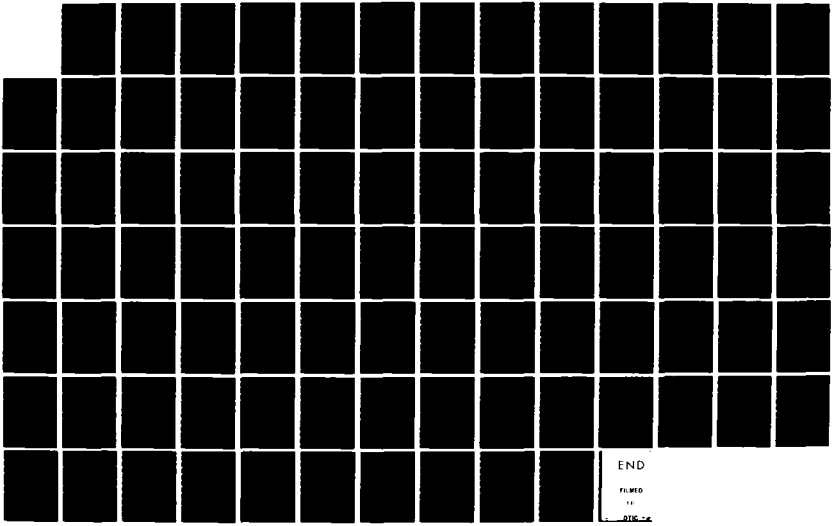
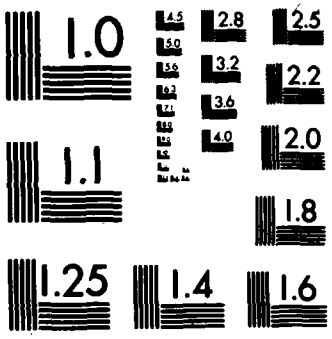
AD-A131 275 ALTERNATE IMPLOSION MODELS FOR THE PLANE PARALLEL DIODE 1/1
(U) JAYCOR ALEXANDRIA VA R E TERRY 02 APR 82
J207-82-002 DNA-6153F DNA001-79-C-0189

UNCLASSIFIED F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

DNA 6153F

ALTERNATE IMPLOSION MODELS FOR THE PLANE PARALLEL DIODE

Robert E. Terry
John U. Guillory
JAYCOR
205 South Whiting Street
Alexandria, Virginia 22304

2 April 1982

Final Report for Period 1 November 1980—1 November 1981

CONTRACT No. DNA 001-79-C-0189

AD A131275

DTIC
ELECTE
AUG 11 1983
S D
B

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

THIS WORK WAS SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE B323082466 T99QAXLA00020 H2590D.

DTIC FILE COPY

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, DC 20305

83 08 11 008

Destroy this report when it is no longer needed. Do not return to sender.

PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY, ATTN: STTI, WASHINGTON, D.C. 20305, IF YOUR ADDRESS IS INCORRECT, IF YOU WISH TO BE DELETED FROM THE DISTRIBUTION LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DNA 6153F	2. GOVT ACCESSION NO. A121273	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ALTERNATE IMPLOSION MODELS FOR THE PLANE PARALLEL DIODE		5. TYPE OF REPORT & PERIOD COVERED Final Report for Period 1 Nov 80—1 Nov 81
		6. PERFORMING ORG. REPORT NUMBER J207-82-002
7. AUTHOR(s) Robert E. Terry John U. Guillory		8. CONTRACT OR GRANT NUMBER(s) DNA 001-79-C-0189
9. PERFORMING ORGANIZATION NAME AND ADDRESS JAYCOR 205 South Whiting Street Alexandria, Virginia 22304		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Subtask T99QAXLA000-20
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, DC 20305		12. REPORT DATE 2 April 1982
		13. NUMBER OF PAGES 90
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A since UNCLASSIFIED

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This work was sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B323082466 T99QAXLA00020 H2590D.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Plasma Physics	Hydrodynamic Codes
Imploding Plasma Loads	Modeling
Radiative Sources	Implosion Model
Photon Sources	Lagrangian Model

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The evolution of imploding Z-pinch plasmas in a plane parallel diode is modeled in two ways through the computer code ZDIPR, described here. The first consists in a complete electromagnetic field solution via the appropriate Hertz vector potential, combined with a radiation coupled hydrodynamic solution of the Braginskii fluid equations. The second, in an approximate electrodiffusive solution for the axial electric field, combined with the radiation coupled hydrodynamic model. The hydrodynamic solution is

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

effected by an implicit advance of ion temperature, electron temperature plus chemical potential, and (in the electrodiffusive limit) the axial electric field. This is subcycled under the control of either a predictor/corrector or fully implicit advance of the radial fluid velocity and Lagrangian position coordinate. Radiation transport is calculated by probability of escape algorithms operating on a preselected set of frequency domains. The electromagnetic field advance is effected through wave equation quadrature (for Z and $\partial_t Z$) followed by a numerical differentiation of this potential (smoothly interpolated to the fluid mesh) which provides E_z and B_θ.

DERIVATIVE SUB t

sub t

TABLE OF CONTENTS

Section		Page
	PREFACE	iii
I.	INTRODUCTION	1
	A. Motivation	1
	B. Computational Approach	6
II.	ACCURATE METHODS FOR NUMERICAL DIFFERENTIATION	13
	A. Survey of the Theory of Smooth Interpolants	13
	B. A Family of Generalized Splines R(x,y)	17
	C. The Interactive Interpolation Package (VAX Implementation)	20
	D. Applications of Interest in Modeling Diode Imploded Plasmas	22
III.	THE HERTZ DRIVEN IMPLODING PLASMA RADIATOR (PROGRAM ZDIPR)	27
	A. Overview	27
	B. Theoretical Summary	29
	C. Electromagnetic Algorithms and Subroutines	32
	D. Transport Coefficients	34
	E. Fluid Density and Flow Field Time Derivatives	37
	F. Fluid Temperature Time Derivatives	42
	G. The Subcycle Calculating Thermal and Electro-diffusive Evolution in the Model Plasma	48
	H. The Complete Fluid Evolution Package - HYDROPUSH	55
	I. The Planned Sequential Benchmark of This Method	65

TABLE OF CONTENTS (Continued)

		Page
Appendix I.	VAX Interactive Interpolation Package.	69
Appendix II.	Derivation of the Electrodiffusive Limit	75
Appendix III.	Area Weighted Differences.	79

PREFACE

We wish to gratefully acknowledge the important and central contributions to this effort by J. Davis, P. C. Kepple and D. Duston at NRL in providing fundamental equation of state and radiation emission data as well as timely and cogent advice throughout the developmental phase. We are also grateful for the painstaking and effective aid given by J. Apruzese and R. Clark at NRL in understanding and using the radiation transport algorithms discussed here.

In addition the program development help of D. W. Mack, S. Slinker and L. Roelofs at JAYCOR is gratefully acknowledged, as are the several subroutines of the IMSL library used here.

Dist
A

I. INTRODUCTION

A. Motivation

Existing models of imploding wire or puff gas plasmas[1,2] leave a number of physical processes unexplored or only partially examined in their assessment of diode imploded plasmas as radiation sources. The one-dimensional implosion code to be described here is an attempt to increase the physical relevance of such models and to enhance the versatility and robustness of the numerical schemes used to implement them.

Among the relatively unexplored physics questions, one finds those of photon/plasma energy exchange, plasma thermoelectric radial stresses and the effect of separately evolved ion and electron temperatures on opacity calculations. The incompletely examined physics involves: the processes of current penetration into the plasma load, the correlations of diode waveforms and plasma motions (generated by reflected power in the diode cavity), the energy partition between thermal motion and ionization potential, and the role of marginally stable drift-speed limitations in determing the implosion dynamics. These considerations can be organized into several concrete goals for the calculation.

1) Find how important are the details of photon/plasma energy exchange in structuring the implosion trajectory and (spatially resolved) energetics. The transitions in qualitative characteristics (ranging from refrigerative collapse to strong reflection) noted in the core-corona model[3] were quite sensitive to the opacity calculation. In a full 1-D implosion calculation the resolution of the enhanced "thermal conduction" due to direct photon/plasma energy exchange will provide a more accurate picture of the plasma emission profile.

2) Determine the role of separate ion and electron temperatures in establishing this emission profile and in governing the plasma dynamics. As compared with a single-temperature picture, one expects the proper two-temperature treatment to alter the pressure gradients in low density regions and change the opacity in higher density regions, perhaps altering the spectral mix of line and continuum radiation.

3) Examine the effect of the chemical potential (properly included in the plasma energy balance as a function of electron energy and ion density) on the implosion energetics and the emission profile.[4] An accurate partition of incoming energy will probably provide lower (more physical) temperatures at peak implosion than earlier models.

4) Develop an accurate picture of the current penetration mechanisms. Diffusive transport of $E_z(r, t)$ (or B_θ) is dominant to a first approximation, but overall implosion performance is sensitive to the effective skin depth.[5] The role of the free wave field components in setting this skin depth has not been examined, and the rate of inward propagation for the drift-speed limitations in J_z may be affected by these components.

5) Resolve the sources of reflected power within the plasma load and relate them to the available measured waveforms from experiment. An electromagnetic calculation of all fields within the plane parallel diode opens one more channel of diagnostic information, and may provide a means of inferring plasma motion from measured waveforms. It will also provide a more accurate statement of energy conservation because the energy contained in the net free wave component will be included.

6) Examine the effect of radial ambipolar electric fields, and the resulting stress, on the implosion dynamics, particularly in the low density regions where drift speed limits are expected. The radial electric fields arise from thermoelectric effects and axial drifts of charge carriers. Large temperature gradients, allowed by weak cross field thermal conduction and enhanced ohmic heating in the corona plasma, may provide a significant radial electric field and resulting inward stress on the plasma.

Once these physical questions are addressed coherently, a smooth upgrading in the sophistication of the radiation emission and transport model becomes a fruitful exercise. With the radiative, hydrodynamic, and electrodynamic models structured from first principles there are no "free knobs" and the subsequent benchmarks with an appropriately configured experiment, if successful, provide a firm basis for similar theory in more difficult and perhaps more practical diode configurations.

Apart from these physical questions are various topics arising in the numerical implementation of whatever implosion model is selected. First is the choice of grid; a typical simulation will show the model plasma compressed several orders of magnitude during the course of a calculation. A strict Eulerian fluid calculation can thus lose its spatial resolution at the final collapse, when this resolution is most needed. Moreover large density gradients are commonplace, and it is probable that the dynamics of the low density region is dominated by the balance of density gradients and magnetostrictive stresses. A Lagrangian description is thus preferable in an implosion/explosion calculation, so long as the external stresses on the system can be defined readily over the spatial domain of interest. On the other hand the fundamental field (E_z) generating the implosion is defined

in the fixed laboratory frame and is most naturally discretized on an Eulerian mesh. The electromagnetic problem therefore requires an interpolation capability which can map external stresses onto a convecting fluid and extract field sources from this moving fluid. The natural compromise adopted here is a completely Lagrangian fluid model convecting through a laboratory-frame (Eulerian) electromagnetic grid. The electromagnetic grid can be made sufficiently non-uniform to concentrate the E_z information density near the axis while the smoothly distorted Lagrangian fluid grid is a natural choice for an accurate resolution of density gradients.

A second concern is the choice of E_z or $B_\theta(J_z)$ as the fundamental electrodynamic variable. In our view the proper selection is E_z because it is the natural field from which to establish drift-speed limitations on the current density. This consideration plays a fundamental role in both the development of the Hertz vector formalism and in the choice of the diffusive limit discussed below (III.A.). Using B_θ and $J_z = \frac{c}{4\pi}(\nabla \times B_\theta)_z$ in a drift speed limitation algorithm has two disadvantages. The current is derived from a potentially noisy differentiation process. The limit $J_z = en_e C_s$ does not imply a completely local value for E_z because the conductivity depends on B_θ, and hence on J_z at other grid points. It is clearly possible to remove the second problem by iteration, but the physical process providing the drift speed limit is a local (fine scale) one. By proceeding first from the local values of E_z, then iteratively establishing small corrections to the (possibly limited) J_z from non-local considerations, one achieves a more natural and probably less noisy convergence to the limit.

A third point is the selection of forward time integration methods. A common preference for conservatively advanced explicit integrators must be examined carefully in the context of this radiatively coupled problem.

As the level of sophistication in the radiative transfer physics is increased a considerable expense is incurred for each fluid time derivative. This expense will scale linearly with the number of frequency domains and quadratically with the spatial resolution. In order to optimize the complete algorithm it is therefore necessary to make the best possible use of every time derivative available from the space-time mesh. Moreover, as distinct from calculations dominated by processes internal to the fluid, the problem of simulating an electromagnetically driven implosion requires a self-consistent time evolution of both field and fluid. This class of problems is more likely to be handled efficiently and accurately by an implicit method, or at least by a predictor/corrector scheme. This choice has only the disadvantage of being more cumbersome to implement, and, as described below, the methods chosen make a smooth transition from "nearly explicit" to "fully implicit." As dictated by the performance characteristics of the complete algorithm, the user will be able to optimize the accuracy and cost tradeoff by several means.

A final consideration is the accuracy of spatial differentiation within either the hydrodynamic or electrodynamic algorithms. If these algorithms can make good use of non-uniform meshes in space and time, then one can sustain useful accuracy with fewer points. The common differentiation methods based on finite differences are quite accurate when a dense mesh is admissible, but if one wishes to keep down the cost of radiative transfer calculations a higher order, smoother scheme is preferable. Some useful alternatives are discussed in Chapter II.

B. Computational Approach

The specific physical content of the model has been described elsewhere.[6,7,8] The task of implementation can be readily broken down into the three problem areas treated below - electromagnetics, hydrodynamics, and radiative transfer. The requirements for each problem area and a summary of the algorithms employed are given.

The electrodynamic problem statement is simple: given voltage waveforms at the periphery of the diode, establish the E and B fields on the interior spatial domain for the duration of the implosion process. By driving the diode E/M model at one radius with a voltage waveform, the current drawn by the load and the voltage waveforms at any interior point are available as predicted observables/diagnostics of the code and the experiment. Conventional approaches replace the vacuum portion of the diode with a circuit relation derived from Faraday's law, but a complete electromagnetic calculation provides more information. If this later goal is chosen then the issue of efficient and effective calculation becomes an important question. With the implosion model discussed here the choice has been to admit either option, viz. the full electromagnetic computation or the electrodiffusive limit and circuit equation.

For the first option, it is instructive and in many ways practical to utilize the generalized Hertz vector potential. First, this new potential reduces the electrodynamics to a single component of a single vector wave equation. This reduction simplifies the calculation by placing no self-consistency constraints on the time evolution of the fields E_z, B_θ. The required self-consistency is automatic when E_z, B_θ are derived from a single potential Z_z. Second, the coupling term between field and plasma is properly defined to all orders in the radial fluid convection velocity

$\beta = V_{fluid}/C$, and smoothly admits marginally stable drift-speed limited conduction and (quasi-static) thermo-electric contributions. Properly, as fast time scale phenomena which track the instantaneous state of the system, these effects should be calculated from a single time slice of information. Without the Hertz vector formulation, an electromagnetic calculation evolving E_z, B_θ separately can only accomplish this to the extent that it satisfies the temporal self-consistency constraint mentioned above. Third, the coupling term is a spatial convolution with a fixed Green's function. The spatial averaging will generally serve to lower the noise level of the calculation by damping any chatter that may appear on the discrete source array extracted from the plasma representation. Fourth, while the use of Z does require the extraction of high order spatial derivatives from discrete data, the accuracy of such spatial differencing can be improved by the use of so-called smooth interpolators.[9] The estimation of local E_z fields from the parent Hertz vector by means of these splines is routinely accurate to $[10^{-6}]$, even between the input data points.

The Hertz-vector-based electromagnetic calculation proceeds from a spatially discrete set of information $\{Z, \partial_\tau Z, J, \partial_\tau J, \partial_\tau^2 J\}$, given at a single time. The first two elements of this set are defined at fixed radial points; the last three, established (sequentially) on a convecting (Lagrangian) grid of plasma cell centers. The local values of E, B at any such time level are determined by $(Z, \partial_\tau Z, J)$ and partially determine the plasma stress and heating rates. From the values of $\partial_\tau J$, $\partial_\tau^2 J$, a spatial convolution produces the Hertz vector source terms. These source terms are interpolated for use in an explicit wave quadrature formula which advances the Hertz vector and its partial time derivative in order to establish Z, $\partial_\tau Z$ at any subcycle time step (used in the thermal conduction scheme) or at the next major time level.

The second electrodynamic option is that of E-field diffusion, which derives from the Hertz potential wave equation in the limit that incoming and outgoing waves are in detailed balance. This limit is approximate but reasonably accurate. It neglects the net displacement current, making any detailed outgoing wave calculation impossible because no wave source is extracted from the plasma. Reflected power is manifested solely as a diminished voltage on the plasma load and the energy delivered is simply $\int^t dt\ (I_{plasma} \cdot V_{gen})$. Here only the plasma current waveform is available as diagnostic information, insofar as the voltage at the convecting plasma vacuum interface is not usually available from experiment. The major simplification is that Z need not be explicitly employed; it is sufficient to diffuse the E_z field and calculate the B_θ field in a (self-consistent) quasi-static manner. Diffusion of E_z rather than B_θ is preferable because the criteria for drift speed limitation stem most directly from the application of an "E-field-to-local-J" mapping.

Diffusion based electrodynamics proceeds concurrently with subcycled thermal conduction. A circuit relation sets the time varying value of E_z at the outer boundary of the plasma, and the time slice information $\{E_{th}, E', B, \dot{\beta}, \dot{\sigma}, \sigma\}$ defined on the (convecting) plasma cell centers determines the material derivative of E_z. Here $E_z' = E + \beta_r \times B_\theta$ is the field in the convecting frame; E_{th} is the axial thermoelectric field. The new field E_z at the next time level is determined implicitly by the time sequence of material derivatives. The plasma mesh evolves in response to the stresses partially determined by the local values of E, B. In this mode of operation some iteration is required in obtaining both DE_z/Dt and B_θ in contrast to the electromagnetic calculation. There only the quasi-static portion of B_θ requires iterative refinement due to the magnetic field dependence of the plasma conductivity.

In either electromagnetic or diffusive mode the imposition of drift-speed limited conduction is straightforward here because the $E_z(r,t)$ profile is established a priori, before J_z is estimated. The rational basis for preference between these two can be derived only when the goals of I.A.(4) above are reached and when the comparative computation costs of well tuned subroutines are established.

The hydrodynamic problem is readily formulated (once E, B are considered given) in terms of well-known techniques. The primary challenge is to produce a quiet, relatively robust algorithm capable of concentrating the computational effort on the faster time scales of the problem, minimizing the number of expensive derivative evaluations required, and retaining spatial resolution of very highly compressed plasma loads.

In view of these requirements a Lagrangian formulation of the two-fluid transport equations for number density, flow velocity, and temperature has been implemented. There are several simplifications to the complete set of transport equations which are appropriate to the imploding diode plasma, but the hydrodynamic model is essentially that of Braginskii. In particular the radial flow field is assumed to be identical for ions and electrons, the electron density is assumed to be in quasineutral equilibrium with the ion density (apart from small ambipolar charge separations), and the system of moment equations is closed with an ideal gas equation of state for each fluid. This reduces the fluid variables to $\{n_I, V_r, T_I, T_e, \mathcal{Z}, \varepsilon_I\}$. However T_e is not a very good thermodynamic variable because it is tightly coupled to $\mathcal{Z}$, the effective ionization state and to ε_I, the specific chemical potential. In order to follow the ionization dynamics efficiently, let $\theta_e = T_e + \frac{2}{3}\frac{\varepsilon_I}{\mathcal{Z}}$ and evolve θ_e instead. The approximation of the collisional radiative equilibrium[10] (CRE) model is to assume a very rapid establishment

of $\mathcal{Z}$ and ε_I from (given) n_I and T_e. Hence the usual heating sources and sinks for T_e, when $\mathcal{Z}$ is fixed, become (in effect) heating sources and sinks for θ_e in the context of CRE ionization dynamics. From a sufficiently accurate representation, preprocessed by the complete CRE model and smoothly interpolated, one may simply advance θ_e (the "grand canonical temperature") and compute T_e, $\mathcal{Z}$, ε_I and all other transport coefficients from the θ-value obtained at any timestep. This later method is presently implemented. As is common practice in Lagrangian calculations, one may also remove the ion density from the set of evolved quantities by conserving the total number of ions in any cell, unless a rezoning is done, and calculating the density n_I from the time varying cell volume. Hence the irreducible set of fluid variables to be evolved is $\{V_r, T_I, \theta_e\}$, with the $r \equiv \int^t dt_1 V_r(t_1)$ defining the variable mesh positions to be used in computing the density n_I. For the choice of diffusive electrodynamics, this set is expanded to $\{E_z, V_r, T_I, \theta_e\}$ and the separate evolution of $\{Z, \partial_t Z\}$ through the Hertz vector wave equation is discarded.

The complete set of magnetoplasma, $\mathcal{Z}$-dependent transport coefficients is available in a single subroutine, and those required for the simplest relevant problem are the axial electrical conductivity, the radial ion and electron thermal conductivities and the radial thermoelectric coefficient. The expansion of the model to include radial ambipolar electric fields, viscous stresses and viscous heating only requires activation of the axial friction coefficient and five viscosity coefficients. Moreover, because the relaxation time is the cornerstone of the transport calculation an extension into more strongly coupled plasma domains is quite straightforward on a cell-by-cell basis. The use of a modified relaxation time based on the local Coulomb logarithm is a useful first approximation in strongly coupled plasma

transport[11] and can be relevant here in some simulations of wire or foil annular implosions.

The hydrodynamic calculation across a major time step proceeds from the time slice $\{N_I, V_r, T_I, \theta_e\}$. Here N_I (the time invariant total number of cell ions), T_I and θ_e are defined at the cell centers, while V_r and r(t) are defined at the cell interfaces. From this basic data the model calculates T_e, $\mathcal{J}$, ε_I, all transport coefficients, all heating rates and all fluid stress terms (after E and B have been established). The major time step is selected from a competition among the magnetosonic Courant condition, fractional cell area changes, local truncation error in estimating future accelerations (DV_r/Dt), and the most rapidly varying electromagnetic source terms. Once the major time step is set, a time window is defined over which thermal conduction, compressional heating, radiative heating or cooling, and ion-electron exchange heating are subcycled at the cell centers (under an assumed cell boundary evolution). This subcycle is implemented using an (implicit) variable order Adams predictor-corrector or Gear method. At the completion of the subcycle the explicit flow field advance (and implied cell boundary motion) is either retained or iteratively refined (becoming the starting point for an implicit method). If desired the iteration proceeds until self-consistency is achieved. The self-consistency criterion is solution of the first order (non-linear) difference equation for the fluid acceleration, to a specified tolerance at all mesh points. For the diffusion based electrodynamics the subcycle includes the E_z field evolution as well.

A central element in the hydrodynamic model is the question of radiation transport. The algorithm presently implemented is a compromise between the full CRE calculation and the much simplified local approximation[12] of the SPLAT code. First, the emission function preprocessed by the

CRE model is represented over the relevant density and temperature domain. One such emission curve is required for each radiation category one wishes to calculate. A simple set is: {lines $h\nu < 1$ keV, lines $h\nu > 1$ keV, free-bound continuum, & Bremsstrahlung}. Next,for each radiation category a damping coefficient is computed for each cell. These damping coefficients are path integrated along a selected ray to produce a probability of escape for that photon category. From the escape probability field a matrix of coupling coefficients is derived. Once the coupling matrix is given, selected inner products with the vector of cell emissions for a particular category produce the cell-to-cell photon exchange and the net radiative loss to the plasma from any zone in each radiation category.

This formulation has the advantages of being easily expanded into a multiple group algorithm and easily modified to include other elements. At present Aℓ and Ar are available, although the representation of θ_e is not yet available for Ar. The disadvantages lie in the added calculation of the coupling matrix and in the need to establish a fresh matrix for each radiation category. It is hoped that the use of implicit methods for the overall hydrodynamic advance will offset the cost of these radiative loss calculations by requiring fewer time derivative evaluations to achieve useful accuracy.

II. ACCURATE METHODS FOR NUMERICAL DIFFERENTIATION

A. Survey of the Theory of Smooth Interpolants

All interpolation methods are based on some criterion of distance between functions. The free parameters of the interpolation scheme are to be chosen in such a way as to minimize the "distance" between the data function $F(x)$ and the interpolant $z(x)$. Smooth interpolants are derived by inventing a distance criterion which measures the oscillation in a function as well as its mean value. The sequence of distance measures (or norms)

$$\tilde{E} = \left(\int_a^b dx\, |F(x) - z(x)|^2\right)^{\frac{1}{2}}$$

$$\tilde{I} = \left(\sum_{n=0}^{\infty} B_n \int_a^b dx\, |F^{(n)}(x) - z^{(n)}(x)|^2\right)^{\frac{1}{2}} \tag{II.1}$$

$$\tilde{I}_w = \left(\sum_{n=0}^{\infty} B_n \int_a^b dx\, w_n(x)\, |F^{(n)}(x) - z^{(n)}(x)|^2\right)^{\frac{1}{2}}$$

represents a progressively more general set of functional similarity criteria. (Here the superscript in parentheses indicates the nth derivative.) Conventional interpolation schemes stem from $\tilde{E}$ (Hilbert's norm) while smooth interpolants arise in connection with $\tilde{I}$ (Talmi-Gilat norm) or $\tilde{I}_w$, to be discussed here. The B_n, $w_n(x)$ are rather freely chosen, subject only to the constraint of convergence.

Each of these norms can be represented also as an inner product operation involving smoothness functionals, i.e.,

$$I_w\,(g, h) \equiv \sum_{n=0}^{\infty} B_n \int_a^b dx\, w_n(x)\, g^{(n)}(x)\, h^{(n)}(x), \tag{II.2}$$

so that, e.g., $\tilde{I}_w^2 \equiv I_w\,(F-z, F-z)$. It is these smoothness functionals that provide the concrete algorithms to be used for interpolation. In particular,

choose a set $g_\ell(x)$ of complete functions (linearly independent and orthonormal over the interval $[a, b]$). Any interpolant $z(x)$ can be expanded in terms of this set, and the expansion coefficients are chosen in such a way as to minimize a smooth norm of $z(x)$ (e.g., $\tilde{I}_w(z)$) subject to constraints. The constraints are the known data concerning $F(x)$, $z(x_j) = F(x_j)$, $z^{(n)}(x_k) = F^{(n)}(x_k)$ for specified sets $j = 1, 2, 3...J$, $k = 1, 2, 3...K$, $n = 0, 1, 2...N$.

This defines an Euler-Lagrange problem which can be transformed (using the orthonormality of the g_k) into a simple linear system for the Lagrange multipliers $\{\lambda^n_{jk}\}$. The matrix for this linear system is determined by evaluating

$$R(X, Y) \equiv \sum_{\ell=0}^{\infty} \frac{g_\ell(X)\, g_\ell(Y)}{\tilde{I}_w(g_\ell, g_\ell)} \tag{II.3}$$

and its partial derivatives at the mesh points, i.e., $x = \{x_j\}$, $y = \{x_j\}$. It is the choice for B_n and $w_n(x)$, therefore, which concretely specifies the interpolation spline $R(X,Y)$. The original work of Talmi and Gilat[9] shows several sets $\{B_n\}$ which are useful in that they render $R(X,Y)$ summable in closed form. A more versatile result is obtained when one invokes $\tilde{I}_w$ rather than only $\tilde{I}$ as they did.

Instead of searching for a set $\{B_n\}$ giving a useful $R(X,Y)$, reverse the process and choose a spline $R(X, Y) = R(X-Y)$ likely to be similar to the smooth functions expected to support the data. The norm $\tilde{I}_w$ is easily shown to be so general as to allow the $\{B_n\}$ to be extracted from the spline $R(X-Y)$, rather than the original (reversed) procedure. For example, select $w_n(x) = (1 - x^2)^n$, $[a,b] = [-1, +1]$, and $g_\ell(x) = P_\ell(x)$ (a Legendre

Polynomial). Then

$$I_w(g_\ell, g_\ell) = \frac{\pi^{\frac{1}{2}}}{\ell + \frac{1}{2}} \sum_{m=0}^{\ell} B_m \frac{(\ell + m)!}{(\ell - m)!}$$

depends only on those B_m for $0 < m < \ell$. Since any function $R(X,Y)$ can be expanded on $[a,b]$ in terms of transformed coordinates $U = X-Y$, $V = X+Y$ one may always adjust the $\{B_n\}$ such that the $g_\ell(X)\ g_\ell(Y)$ expansion reproduces the required coefficients for the expansion of R in terms of U alone.

Using $R(X,Y)$ as the fundamental element one may cast the general spline and its defining linear system in a very simple form,

$$Z(X) = \sum_{j=1}^{N} \sum_{m=0}^{\sigma} \lambda_j^m \ \partial_{x_j}^{(m)} R(X,X_j)$$

$$F_\ell^{(m)} = \sum_{k=1}^{N} \sum_{n=0}^{\sigma} \left[\partial_{x_\ell}^{(m)} \ \partial_{x_k}^{(n)} \ R(x_\ell,x_k) \right] \lambda_k^n \ .$$

Here the function $R(x,y)$ is assumed available in closed form, the $F_\ell^{(m)}$ represent input data giving the data function and its derivatives (through order σ) at any of the x_ℓ. The notation convention here is that if any $F_\ell^{(m)}$ is left out, so also is its corresponding λ_ℓ^m .

In summary, the λ_ℓ^m represent the unknown Lagrange multipliers required to minimize the interpolant with respect to the $\tilde{I}_w$ norm implied by one's choice for $R(x,y)$, subject to the data constraints embodied in the $F_\ell^{(m)}$.

The usual application of this involves specifying derivatives only at the end points, but this is only a matter of choice and one may readily extend the theory to include integral constraints as well ($m < 0$).

A compact matrix notation for this algorithm allows one to state some simple rules for its use as a numerical differentiation scheme. Let

$$\left[\partial_{x_i}^{(m)} \; \partial_{x_j}^{(n)} R(x_i,x_j)\right] \equiv R_{ij},$$

$$\left[\partial_{x_i}^{(p)} \; \partial_{x_j}^{(n)} \; R(x_i,x_j)\right] \equiv R_{ij}^{p} \quad ,$$

$$\left[\partial_{x}^{(p)} \; \partial_{x_j}^{(n)} \; R(x,x_j)\right] \equiv R_{xj}^{p} \text{ (generally not a square matrix)},$$

and $F_{\ell}^{(m)} \equiv F_k$, $\lambda_{\ell}^{(m)} \equiv \lambda_k$ with $k = 1, 2, 3 \ldots m + \ell, \ldots J + N\cdot$(number of derivative conditions given). The defining linear system is simply $R\cdot\lambda = F$, while the spline at any point x becomes $z(x) = \lambda \cdot \partial_{\underline{y}}^{(\lambda)} R(x,\underline{y})$. Here the operator $\partial_{y}^{(\lambda)}$ implies a derivative only for those elements of λ corresponding to derivative constraints of a particular order; for most λ elements this derivative will be of order zero, i.e., correspond to a function value constraint. If one wishes to write the analog of a finite difference formula estimating the derivatives of a data function F at the mesh points used in R, then R^{-1} is required.

$$\partial_{x_i}^{(p)} F \approx \partial_{\underline{x}}^{(p)} Z = \left[\partial_{\underline{x}_i}^{(p)} \; \partial_{\underline{y}_j}^{(\lambda)} R(\underline{x},\underline{y})\right] \cdot R^{-1} \cdot F$$

$$F^{(p)} \approx R^{p} \cdot R^{-1} \cdot F \quad . \tag{II.4}$$

Similarly the method provides a Jacobian defining the sensitivity of a derivative estimate to the data values producing it, i.e.

$$\frac{\partial F^{p}}{\partial F} = R^{p} \cdot R^{-1} \quad . \tag{II.5}$$

Derivative estimates off the original data points are given by the formula

$$\partial_{\underline{x}}^{(p)} F \simeq R_x^{\,p} \cdot R^{-1} \cdot F \quad . \tag{II.6}$$

In comparison with more common interpolation schemes, the "smooth" interpolators have several advantages. First, the algorithm is clearly vectorizable, except for the calculation of R^{-1}. In a piecewise spline scheme any evaluation point x (distinct from the x_j) must be nested in its appropriate interval before the interpolation can proceed and this represents additional (non-vectorizable) calculation overhead. Using (II.6) to evaluate the interpolant, one need not even put the values x in any particular order. A further advantage over piecewise splines is a reduction in storage space for the spline coefficients; a piecewise scheme requires the storage equivalent of $M \cdot (J + N)$, where M is the order of the spline. There are also significant increases in accuracy, which will be discussed below, and the treatment of nonuniform meshes becomes much less formidable.

B. A Family of Generalized Splines R(x,y)

One of the original splines noted by Talmi and Gilat corresponds to the choice $B_n = D^n/n!$ in I, and leads to the Gaussian spline

$$G(x,y) = \frac{1}{(\pi D)^{\frac{1}{2}}} \exp - (x-y)^2/4D \quad . \tag{II.7}$$

The parameter D plays the role of a correlation width in coupling the data points across the mesh, and thus controls the condition number of the resulting matrix R_G. Qualitatively, as D increases from some small value, the interpolant Z(x) evolves from a "pickett fence" structure toward a smoother structure, then further increases in D will require λ elements

of alternating signs in order to follow the data variations. As $D \to \infty$ the R_G matrix moves toward algorithmic singularity for any inversion method using finite precision arithmetic. Usually, for positive definite data exhibiting clear trends, an optimal D will be found (for any particular mesh) which allows a smooth fit with positive definite values λ. In the case of non-uniform meshes it is useful to let D vary across the mesh according to the local data point density. The spline becomes

$$G(x, y_j, D_j) = \frac{1}{(\pi D_j)^{\frac{1}{2}}} \exp - (x-y_j)^2/4D_j$$

so that the effective norm now involves the mean value $\overline{D}$, a $B_n \sim \overline{D}^n/n!$, and some weight function $w_n(x)$. One finds in general that, in order to construct a globally smooth interpolant on a highly distorted mesh, an adjustment of D_j which preserves the number of e-foldings per interval is advisable. Such an adjustment acts to diminish the variations in the effective bandwidth of R_G.

A few numerical examples of this interpolation method with the Gaussian (or G-spline) are sufficient to convince the user that functions F(x) with nearly null derivatives (anywhere on the domain) are rather pathological cases. This leads quite naturally to the exploration of other R(x,y) in attempting to accommodate such data. A useful set is derived by integrating the G-spline repeatedly:

$$E\ (x, y_j, D_j) = 1 + \mathrm{erf}\ (u_j)$$

$$E_1(x, y_j, D_j) = 2\ D_j^{\frac{1}{2}} \left\{ u_j \left[1 + \mathrm{erf}(u_j) \right] + \frac{e^{-u_j^2}}{\pi^{\frac{1}{2}}} \right\} \qquad \text{(II.8)}$$

$$E_2(x, y_j, D_j) = 4D_j \left\{ \left[1 + \mathrm{erf}(u_j) \right] \left[\frac{u_j^2}{2} + \frac{1}{4} \right] + \frac{u_j\ e^{-u_j^2}}{2\pi^{\frac{1}{2}}} \right\} .$$

where $u_j = (x - y_j)/2\, D_j^{\frac{1}{2}}$ determines the relative coupling of the data points. The E-splines are the smooth-norm generalizations of the familiar B-spline elements $(x - y)_+^m$ used in piecewise spline theory. They are of particular utility in accommodating data which exhibit a clean asymptotic trend or areas of null derivatives. For example in Table I is shown a (fully optimized in D) test of the E-spline sampling a function

$$F(x) = A \exp -\left(\frac{x - x_1}{B}\right)^4 + C \exp - \left(\frac{x - x_2}{D}\right)^4$$

at a progressively larger number of points. The quantity tabulated is the average fractional error in $Z^{(m)}$ for m=0, 1, 2, 3, 4, over the domain [-1, +1]. The interpolant and test function are compared on a mesh much finer than that used to sample F and to originate the calculation of λ values for Z(x).

TABLE I

rel. error in $Z^{(m)}$, averaged over [-1, +1]:

Original sample points of F	m=0	1	2	3	4
21	2.06E-3	0.346	0.344	0.58	22.3
31	1.10E-5	4.29E-4	1.16E-3	8.26E-3	3.96E-2
41	4.69E-7	8.92E-5	1.92E-4	6.79E-4	2.30E-3
51	2.53E-7	8.77E-6	6.09E-5	7.16E-4	7.77E-3
61	7.41E-7	7.49E-5	2.59E-4	3.15E-3	8.25E-3

As is evident from these results the E-spline can track accurately data of quite different functional form. This example is also notable in that a G-spline would provide a much poorer result - the constants in F(x) had

been adjusted so as to produce a very flat region in the vicinity of $x = 0.2$. The degradation in this interpolation for larger numbers of sample points is a manifestation of accumulated roundoff in the linear system solution. Moreover, in common with other methods, most of the average error is coming from the ends of the domain. The accuracy in the interior is often an order of magnitude better than the mean.

As with any fresh technique there are some points less well understood in connection with these new splines. In particular, one is a robust algorithm for the choice of D_j. At present the complete automation of these splines is not quite satisfactory. Choosing D_j based on the sample density certainly works roughly but the optimum value is apparently not independent of the input data F, and more systematic study will be required on this topic. Also the E-splines tend to overshoot functions which rise and level off sharply, but this can probably be corrected by adding an offset in the argument u_j, i.e. $u_j = (x - y + \sigma_j)/2\, D_j^{\frac{1}{2}}$.

The experience thus far has been very positive with respect to "interactive" fits (i.e. where the user chooses the σ_j and D_j rather than automating their choice), and in the case of extracting the electric field, E, from a Hertz potential on a fixed but non-uniform radial mesh. Specific examples are discussed in Section D.

C. <u>The Interactive Interpolation Package (VAX Implementation)</u>

The present vehicle for preprocessing CRE emission or equation-of-state data is a pair of command file synonyms: REPRESENT and EVALUATE. The details of their usage are documented in Appendix I. The REPRESENT algorithm processes an input file assumed to contain a mesh y_j and one to three functions F_j evaluated at the mesh points. The user is prompted for the spline choice {G-splines, E-splines, or E_2 splines} and the width parameter D(or $\bar{D}$

if the given mesh is found to be non-uniform). The appropriate R matrix is calculated for the mesh given and an IMSL linear system solver (LEQT2F) is used to calculate a λ_j vector for each function F_j input. The output consists of two files by the same root name (user supplied). The first is suffixed "H" and is a header file providing default and dimension information to the companion process EVALUATE. The second is an interpolation file containing a copy of the original mesh y_i, the D_j required and the resulting λ_j for each of the functions F_j input.

In the present configuration REPRESENT allows the user to specify boundary derivatives of first and/or second order at either limit of the y domain. The boundary condition descriptors are stored in a 2 x 2 array, the boundary derivative matrix (BDM), and embedded in the header file. The overall accuracy of the linear system solution is controlled by a specified number of digits of precision in the input file (IDGT) and the number of iterative refinements allowed to the IMSL solver (ITMAX). If invoked, the iterative refinement proceeds until the solution can be converged to IDGT precision. Usually iterative refinement is not needed and, for the G-spline, often appears to be unstable as well.

The companion process EVALUATE produces a file of interpolant values and derivatives through some specified order (≤ 4), for any of the λ_j vectors contained in an output file from REPRESENT. The evaluation domain can be any subset of the original mesh ($y_1 < y < y_J$) or can even line outside it. EVALUATE also allows the user to change the spline type if desired. For example λ_j calculated with a G-spline can be evaluated with an E-spline to estimate the (definite) integral of the original data over any subinterval.

D. Applications of Interest in Modeling Diode Imploded Plasmas

(i) Estimating an E_z field

The primary problem in using the Hertz potential is the requirement that higher order derivatives be extracted from the discrete, time-dependent potential $Z(r_i,t)$. As a test of the smooth interpolator's ability to resolve these fields, a model Hertz potential

$$F_z(x) = \frac{\mathcal{E}_0}{4}\left\{(1+\varepsilon)x^4 - \int_0^{x^4} d\mu\, \frac{e^{-\mu}-1}{\mu}\right\},$$

where $x_i \equiv \frac{r_i^{1/2}\, a_0^{1/4}}{b^{1/4}}$, was chosen as a data function. This is the Hertz potential for the E field

$$E(r,t) = \mathcal{E}_0(t)\left\{1+\varepsilon - \exp-(r^2/b)\right\},$$

exhibiting a sharp rise in the vicinity of $r \sim b^{1/2}$. In a similar manner to the test described in Table I above, the function $F_z(x)$ was sampled at 41 points on a uniform x mesh (hence a mesh in r which is denser near the origin). The interpolation was carried out using the E_2 spline, and the interpolant and its derivatives were compared to those of the original, analytically given test function on a mesh much finer than the interpolation grid. The results are shown in Table II as an average relative error and a typical interior relative error for the interpolant z(x) and its first four derivatives.

TABLE II

$Z^{(m)}$	m=0	1	2	3	4
mean error	1.36E-5	1.00E-5	5.21E-5	1.0E-3	0.239
interior error	2.47E-6	3.25E-7	2.73E-6	7.56E-6	1.03E-4

Again most of the average error occurs near the end points. The test function is well matched even between the sample points (where one's Lagrangian fluid grid will almost always be located).

The good fit to the potential holds as well in estimates of $E(x)$ and $\partial_x E$, even though

$$E = \frac{1}{4a_o x^3}\left\{\partial_x Z + x\,\partial_x^2 Z\right\}, \text{ and}$$

$$\partial_x E = \frac{1}{4a_o x^4}\left\{x^2\,\partial_x^3 Z - x\,\partial_x^2 Z - 3\,\partial_x Z\right\}$$

involve linear combinations of the derivatives of Z. Typical interior errors for E are 3E-6 and for $\partial_x E$, 1E-6. The E_2 spline has routinely extracted these E-fields in many such tests, though the spline function by itself only roughly resembles the test function.

(ii) Representation of thermodynamic data

In order to avoid advancing both T_e (electron temperature) and ε_I (specific chemical potential) concurrently in time, the use of a collisional radiative picture will hold these quantities in quasi-equilibrium. The calculation of this quasi-equilibrium self-consistently with the plasma evolution is a very expensive process; it is therefore useful to fit a paradigm calculation of ε_I and $\mathfrak{z}$ (the mean charge state) and use this as an approximate representation of the underlying atomic physics.

The fundamental variable used is a "grand canonical temperature," $\theta_e \equiv T_e + \frac{2}{3}\,\varepsilon_I/\mathfrak{z}$. It can be most easily represented by defining a branching ratio $b = \theta_e/T_e$ and examining $b(n_I,\theta_e)$. Owing to the large domain of θ_e values relevant to the model, the optimal data to fit are $\ln b$ and $\ln \theta_e$.

In Figures 1 and 2 respectively are plotted the interpolants to $\ln b$ vs. $\ln \theta_e$ (for $n_I = 10^{19}$) and the fits to $\varepsilon_I(T_e)$ and $\mathfrak{z}(T_e)$ with θ_e shown for comparison (again at $n_I = 10^{19}$). The plots are very true to the original data (5-6 significant figures) and were generated by splines of the form

$$\ln b(\theta_e, n_I) = \sum_j \lambda_j^b (n_I) \cdot G(\ln \theta_e, \ln \theta_j, D_j) \tag{II.9a}$$

$$\mathfrak{z} = \ln\left\{\sum_j \lambda_j^{\mathfrak{z}} (n_I) \cdot E(\ln T_e, \ln T_j, D)\right\} \tag{II.9b}$$

$$\varepsilon_I = \ln\left\{\sum_j \lambda_j^{\varepsilon} (n_I) \cdot E(\ln T_e, \ln T_j, D)\right\} \tag{II.9c}$$

with a sampling of 31 data points. The choice of G-splines or E-splines is guided by the general form of the data. The branching ratio b is a cleanly peaked function, so a G-spline is the natural choice. The ionization state and chemical potential curves exhibit flat portions (in between the opening of atomic shells) so the E-splines are indicated.

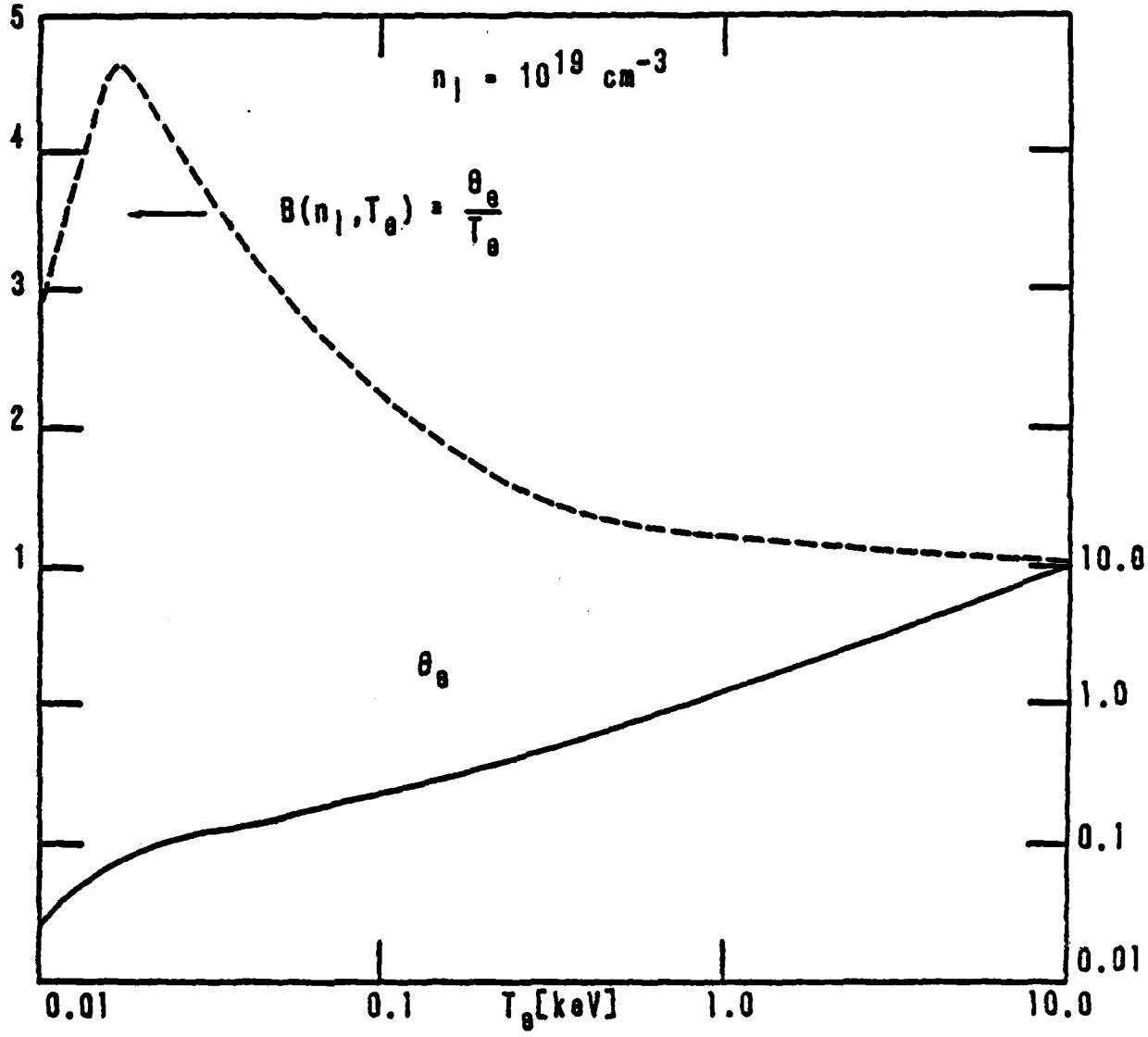

$n_i = 10^{19}$ cm^{-3}
$B(n_i, T_e) = \frac{\theta_e}{T_e}$
θ_e
5
4
3
2
1
10.0
1.0
0.1
0.01
0.01
0.1
T_e[keV]
1.0
10.0

FIGURE 1

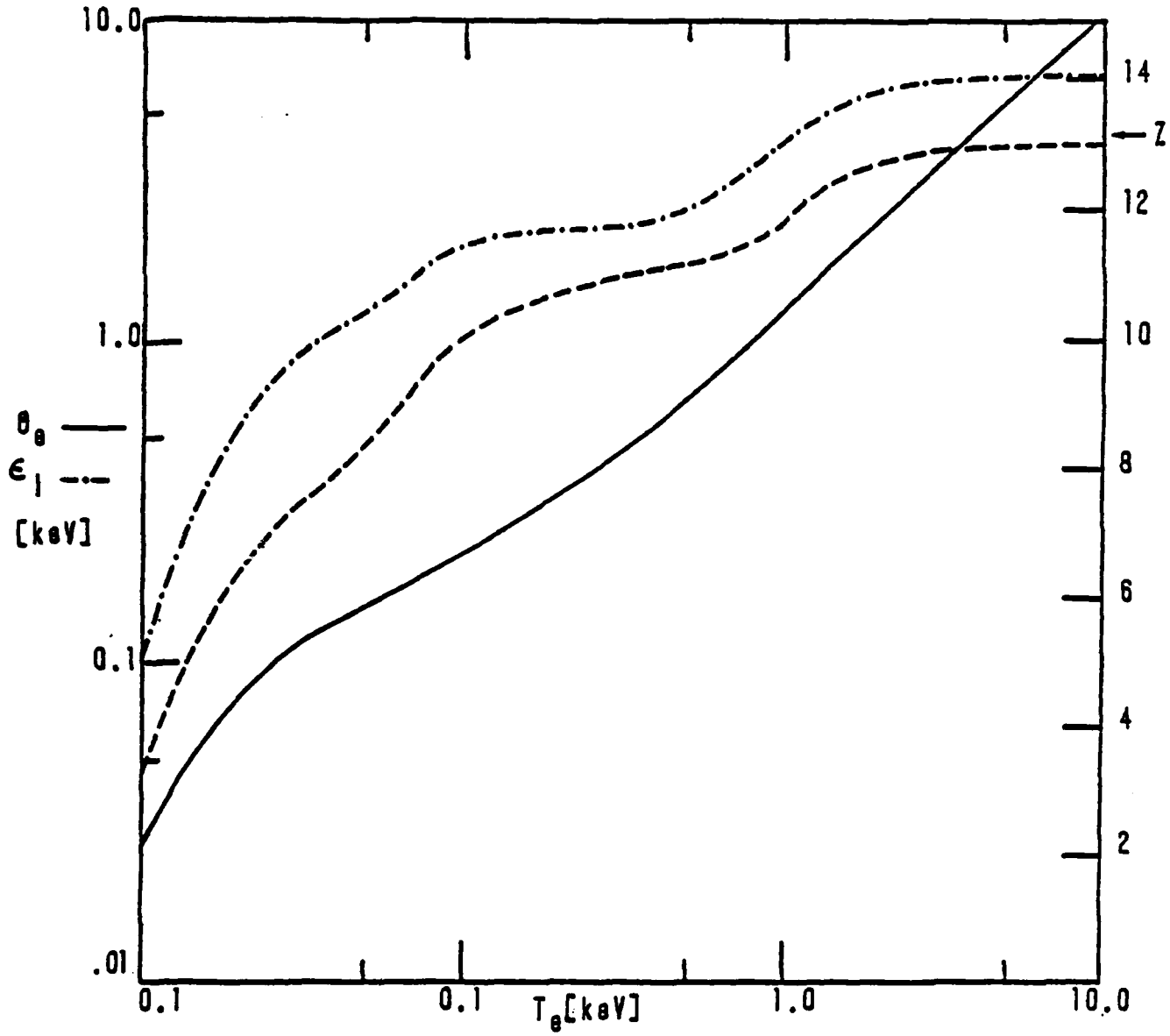

10.0
1.0
0.1
.01
θ_e
ϵ_1
[keV]
14
Z
12
10
8
6
4
2
0.1
0.1
T_e[keV]
1.0
10.0

FIGURE 2

III. THE HERTZ DRIVEN IMPLODING PLASMA RADIATOR (PROGRAM ZDIPR)

A. Overview

A computer program is being constructed to solve the one-dimensional (in radius) time-dependent interaction of a plasma annulus (with temperature-dependent degree of ionization) and an electromagnetic driving field in a cylindrical cavity. It was shown previously that the number of equations describing the fields can be reduced and the computational stability probably improved by the use of a generalized Hertz vector, Z, from which E and B field vectors can be derived by differentiations. In the limit of high plasma density the wavelike term in the field equations can be neglected; this corresponds to a field-diffusion approximation. In simple cases — constant scalar conductivity for example — this gives the usual magnetic diffusion equation.

The program must couple field or Hertz potential quantities, evaluated most conveniently on a stationary (Eulerian) grid of points, with fluid equations describing the mass-motion (V) electron and ion temperatures (T_e, I_I), ionization ($\mathcal{Z}$), and effective conductivity (Σ) of the plasma, all functions of r and t evaluated most conveniently on a comoving (Lagrangian) grid. The differential equations are integrated forward in time by a variable-timestep method (GEAR), but because the plasma thermodynamic state evolves much more rapidly than the fields do, it is updated separately, on a fast timescale, with the subcycle timesteps shorter than the major timestep T required for integration of the field and momentum equations. Doing these updates on different timescales is expected to save a significant amount of running time and expense in the computation, and may make the difference between a practical cost and an impractically large one.

In the code description that follows, the subroutines that update the plasma state on the Lagrangian "r" mesh are called MESHSTRESS

(forces),TDOT (heating), CREOS (ionization state) and FLUIDOTS (material derivatives), all of these being used by the subroutine GEARBOX which controls the updating as part of the overall fluid-advance routine HYDROPUSH.

The subroutine that updates the field or Hertz potential variables on the Eulerian "R" mesh is called ZPUSH. When the Hertz potential representation is used, the subroutine HERTZDOTS updates the source terms in the potential equation at each major timestep. When the electric field diffusion representation is used, the corresponding subroutine EDOT updates the field source $\partial_\tau J$ and alternate plasma state movers (GEARBOX → TETGEAR; FLUIDOTS → TETDOTS) are used because different electromagnetic information is required.

This code, called ZDIPR, is now mostly written and some of its portions have been separately checked. Further testing and integration of the various subroutines will be carried on in the coming weeks.

B. Theoretical Summary

The electromagnetic calculations required are either solutions of the Hertz wave equation or of the electrodiffusive equation derived from it. If a complete calculation is desired, all fields and currents are derived from solutions to

$$\partial_\tau^2 Z - u^{-1}\,\partial_u\,(u\,\partial_u\,Z) = 4\pi\int d\hat{u}\;\hat{u}\;\ell n\left(\frac{u^>}{u_n}\right)\partial_\tau\,J\quad, \qquad \text{(III.1)}$$

where $\tau = ct/R_o$, $u = R/R_o$, $\Sigma = r_o\sigma/c$, $J = \Sigma\hat{E}$, $E = E_z/Q_oR_o^{-2}$, $B = B_\theta/Qr_o^{-2}$, $E = -u^{-1}\,\partial_u(u\partial_u\,Z)$ and $Z = Z_z(r,t)/Q_o$ define the dimensionless variables and fields. If one gives up some information concerning the details of the diode fields and makes the assumption that the incoming and outgoing wave components are in detailed balance, then the wave equation above can be transformed to a diffusion equation

$$\frac{D}{D\tau}E = \{u^{-1}\,\partial_u\,u\partial_u\,E\}\,\frac{1}{4\pi\Sigma} - \hat{E}\,\partial_\tau\,\ell n\,\Sigma - \partial_\tau\,E_{th} - B\,\partial_\tau\beta\;, \qquad \text{(III.2)}$$

where E_{th} is the dimensionless thermoelectric field, $\beta = V_{fluid}/c$, $E' = E + \beta B$, $\hat{E} = E + \beta B + E_{th}$ and $\frac{D}{D\tau} \equiv \partial_\tau + \beta\partial_u$. (This diffusive limit is derived in Appendix II.)

The fluid response to the electromagnetic stresses and heating is embodied in the relations

$$\frac{D}{Dt}V = -\frac{\tilde{T}}{\tilde{m}}\,\partial_r\,\ell n\,n_I - \frac{\partial_r\tilde{T}}{\tilde{m}} - \frac{J_z B_\theta}{\tilde{m}cn_I} + \frac{eE_r}{\tilde{m}n_I} \qquad \text{(III.3)}$$

and,

$$\frac{D}{Dt}\theta_e = -\frac{2}{3}T_e(\nabla\cdot V) + \left\{\nabla\cdot(\chi_e\partial_r T_e) + \nabla\cdot(\beta^{Tu}_{\wedge} U_z)\right\}\left(\frac{3}{2}zn_I\right)^{-1} \tag{III.4a}$$

$$+\frac{2}{3\mathcal{J}n_I}(J_z\cdot E_z') + \frac{2m_e}{m_I}\left(\frac{T_I - T_e}{\tau_e}\right) + Q_{Rad}\ ,$$

$$\frac{D}{Dt}T_I = -\frac{2}{3}T_I(\nabla\cdot V) + \left(\frac{3}{2}n_I\right)^{-1}\nabla\cdot(\chi_I\partial_r T_I) + \frac{2\mathcal{J}m_e}{m_I}\left(\frac{T_e - T_I}{\tau_e}\right)\ . \tag{III.4b}$$

In these expressions $\theta_e = T_e + \frac{2}{3}\epsilon_I/\mathcal{J}$, $\tilde{T} = T_I + \mathcal{J}T_e$, $\tilde{m} = m_I + \mathcal{J}_{max}\, m_e$; $\chi_{e,I}$ is the thermal conductivity, τ_e the plasma relaxation time, E_r the ambipolar radial field (with ρ its induced charge density), and Q_{rad} is the net (local) radiative heating or cooling. The dimensional version of fields are subscripted with a vector component; dimensionless fields are not. The radial electric field is established as a solution to an integral equation derived from the radial component of Ohm's law, (cf. reference 6). The drift-speed-limited current condition is supplemented by a (nonlinear) change in Σ where the local E field requires it. The overall architecture of the model is indicated in the flow chart of Figure 3, and some detailed documentation of the physical content has been mentioned above (Chapter I).

The following sections describe in some detail all the specific algorithms which are required to implement (III.1-4). For the sake of simplicity, a common notation for all of them is given here.

The (nonuniform) time levels for any variable are indexed with a leading superscript; the Lagrangian fluid mesh is denoted by r and its material derivatives, by a superscripted dot (or dots). Spatial indexing is denoted by a trailing subscript and various cell-to-cell averaging operations are denoted by an overbar or by angle brackets.

FIGURE 3

RADIATION-COUPLED ELECTROMAGNETOHYDRODYNAMICS

PLANE PARALLEL WAVEGUIDE

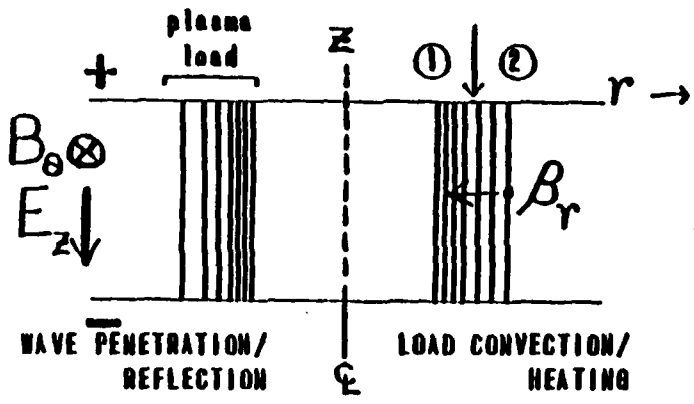
$$D^2 Z = -c^{-2} \int dr_1\, r_1\, G(r,r_1)\, \partial_t J_z(r_1,t)$$

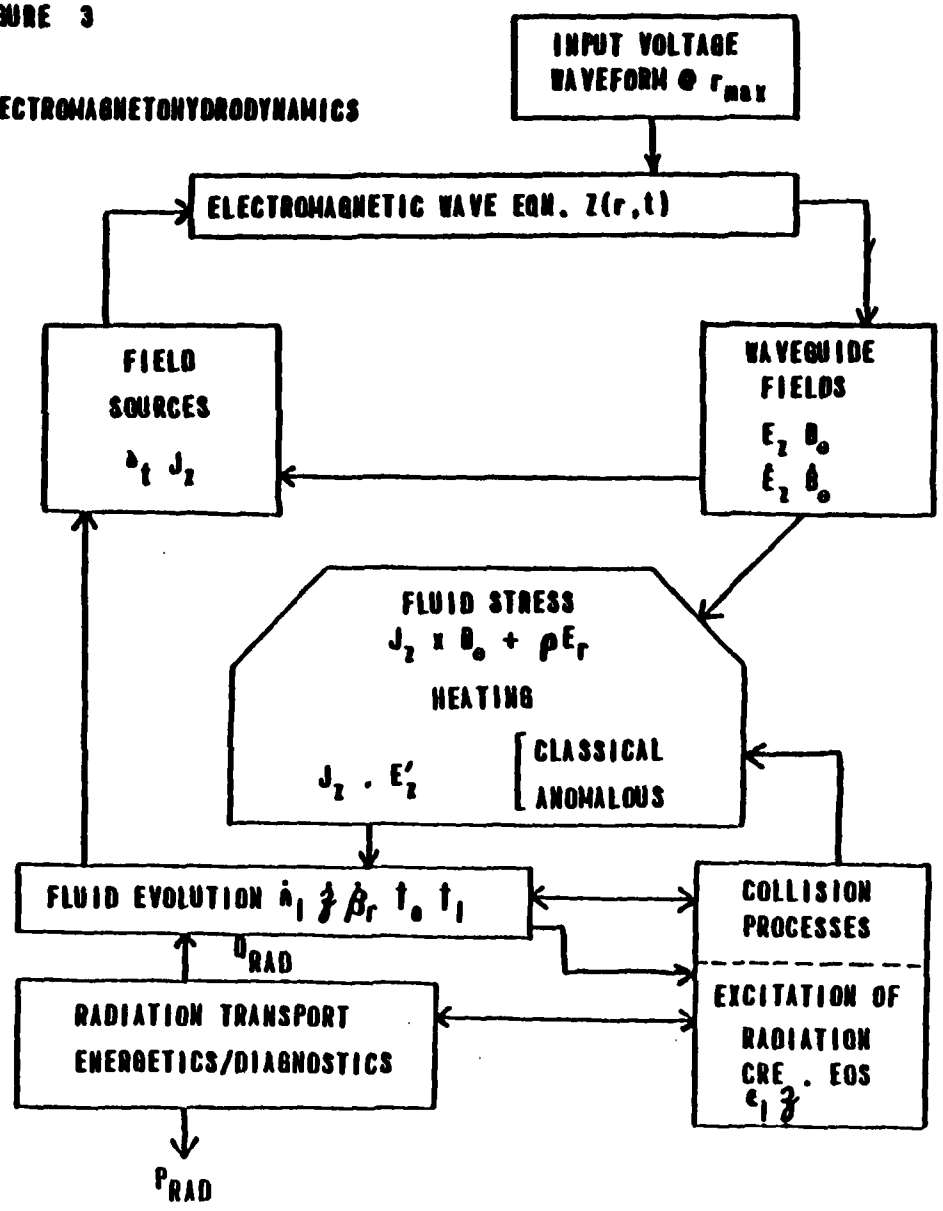
$$E_z = \nabla \times \nabla \times Z$$

$$B_\theta = \nabla \times A(Z)$$

① $J_z = \sigma E'$ $\quad E'_z = E_z + \beta_r B_\theta$

② $J_z = n a_\theta c_\theta$





Since the code performs a subcycle calculation wherein the time step is set by the IMSL Gear integrator, these intermediate time levels are denoted by a leading superscript asterisk. The subcycle interval is denoted by T, and its end points by Tlo and Thi. Other spatial grids are employed in addition to r_j. The Eulerian electromagnetic grid is denoted by R_k, and the (Lagrangian) area coordinate $a=r^2$ or (Eulerian) coordinate $\mu=\sqrt{u}=(R/R_0)^{\frac{1}{2}}$ will be used also. The evaluation of any quantity at the Lagrangian cell center will be denoted by a leading subscript "c". For example, ${}^{*}\ddot{r}_j$ denotes a comoving acceleration on the boundary of the jth cell evaluated at some arbitrary point in the subcycle: $Tlo < \tau^* < Tlo + T = Thi$; and ${}^{i}\partial_{\tau} Z_k$ denotes the first (partial) τ - derivative of the Hertz potential at the ith major time level and the kth grid point of the R mesh. On the other hand, ${}^{*}_{c}\dot{E}_j$ denotes the first material derivative of E_z at the jth cell <u>center</u> at some time τ^* in the subcycle. In general the fixed mesh (R or u or μ) will be indexed spatially by k; the comoving mesh (r or a), by j. Exceptions will be noted.

C. Electromagnetic Algorithms and Subroutines

Either the electromagnetic mode or the electrodiffusive mode requires the establishment of a local E_z field on the plasma mesh continuously through the subcycle. The electrodiffusive mode is discussed in Section III.G(iv) below; only the electromagnetic calculation requires distinct subroutines and is discussed here.

The intermediate time τ^* chosen by the subcycle integrator is the fundamental parameter controlling the Hertz potential wave equation integrator ZPUSH. The input data to ZPUSH are, in addition to τ^*, the previous Hertz potential (with its partial τ derivative) at the last major time level and

the source terms arising on the Lagrangian plasma mesh, i.e. $\{{}^iZ_k, {}^i\partial_\tau Z_k, {}^i\partial_{\tau c}J_j, {}^i\partial^2_{\tau c}J_j\}$. The output of ZPUSH is simply $*Z_k$, $*\partial_\tau Z_k$, from which one may derive the required local fields in the fluid.

The value of $\tau*$ determines a radius of integration for the wave quadrature formulae,[7] which generate the time-advanced potential and its time derivative. The source terms and initial data are functions of radius only, but the wave equation solution requires them to be mapped onto a regular array of quadrature points. The quadrature points are arranged within circles of radius $\tau*$ centered at each electromagnetic mesh point R_k. The source terms from the plasma mesh ${}^i\dot{r}_j$ are smoothly interpolated to the whole space R using a G-spline and, when combined with the existing interpolation coefficients for the initial condition data, one may define six wave quadrature functions, QF1....QF6. These are then input to WAVEQUAD, which evaluates them at the needed quadrature points and forms the inner products that estimate the Poisson integral solution for the Z wave equation.

Once $*Z_k$, $*\partial_\tau Z_k$ are output by ZPUSH, the companion routines EFIELD and BFIELD use these values and $*J_j$ to produce $*E_j$ and $*B_j$ by means of E_2 spline interpolation. Since $*J_j$ is only known from $*E_j$ and $*\Sigma_j$ the previous value for B_j is used to estimate $*\Sigma_j$ first. The quasi-static B_j component is then iterated to convergence, as discussed in Section III.G(ii), and a self-consistent $*E_j$ and $*B_j$ are established.

During a subcycle the use of ZPUSH, EFIELD and BFIELD, with ${}^i\partial_{\tau c}J_j$, ${}^i\partial^2_{\tau c}J_j$ held fixed, provides all the field information needed by the hydrodynamic portions of the program. These sources are updated however at each major time level i, i + 1, i + 2... using the subroutine HERTZDOTS, the fluid-to-field interface code. HERTZDOTS references the material derivatives

${}^{i}_{c}\dot{J}_j$ and ${}^{i}_{c}\ddot{J}_j$ and the plasma mesh ${}^{i}_{c}r_j$, ${}^{i}_{c}\dot{r}_j$ in order to compute these sources. The higher order material derivatives in J are calculated using complete information on the time derivatives of E, Σ and the drift speed limitations to J.

D. Transport Coefficients

The central elements of the field-to-fluid interface are the transport coefficients, calculated by the subroutine BRAGINSKII. This code produces values for as many as 20 transport coefficients, as functions of the ion density (n_I), electron temperature (T_e), ion temperature (T_I), an effective ionization state ($\mathfrak{z}$), and the local megnetic field B_θ. Input units are CGS, except for temperatures in [keV].

The coefficients are evaluated at the cell centers and a local coordinate system: $\hat{z} = \vec{B}_\theta/|B_\theta|$, $\hat{x}$ - any orthogonal direction, $\hat{y} = \hat{z} \times \hat{x}$ is employed. The $\hat{z}$ is called the parallel direction; $\hat{x}$, the perpendicular; and $\hat{y}$, the cross product. Output [CGS] transport coefficients are written in three common blocks (PARATRAN, PERPTRAN, CRPXTRAN) which connote these directions, and in two additional common blocks (VISCTRAN, EXCHANGE) which contain viscosity and electron/ion heat exchange rates. The logical matrix of coefficient selectors CONTROL (6, 3) in the common block TRANSPORT-CHOICE determines which coefficients are evaluated, as shown below, in Braginskii's notation.

TABLE III

Coefficient of friction	(1, 1) $\alpha_{\parallel}(\mathcal{Z}, B)$	(1, 2) $\alpha_{\perp}(\mathcal{Z}, B)$	(1, 3) $\alpha_{\wedge}(\mathcal{Z}, B)$
Thermoelectric coefficient	(2, 1) $\beta_{\parallel}^{uT}(\mathcal{Z}, B)$	(2, 2) $\beta_{\perp}^{uT}(\mathcal{Z}, B)$	(2, 3) $\beta_{\wedge}^{uT}(\mathcal{Z}, B)$
Electron thermal conduction	(3, 1) $\chi_{\parallel}^{e}(\mathcal{Z}, B)$	(3, 2) $\chi_{\perp}^{e}(\mathcal{Z}, B)$	(3, 3) $\chi_{\wedge}^{e}(\mathcal{Z}, B)$
Ion thermal conduction	(4, 1) $\chi_{\parallel}^{I}(\mathcal{Z}, B)$	(4, 2) $\chi_{\perp}^{I}(\mathcal{Z}, B)$	(4, 3) $\chi_{\wedge}^{I}(\mathcal{Z}, B)$
Electrical conductivity	(5, 1) $\sigma_{\parallel}(\mathcal{Z}, B)$	(5, 2) $\sigma_{\perp}(\mathcal{Z}, B)$	(5, 3) $\sigma_{\wedge}(\mathcal{Z}, B)$
Viscosity (ion & electron)	(6, 1) $\eta_{0-4}(\mathcal{Z}, B)$	(6, 2) $\eta_{0-4}(\mathcal{Z}, B)$	(6, 3) $\eta_{0-4}(\mathcal{Z}, B)$

If CONTROL (I, J) has the value "true", this triggers the evaluation of the indicated coefficient at all cell centers of the input mesh. Any true value in CONTROL (6, 1, 2 or 3) causes evaluation of all viscosity coefficients, and the exchange heating is always evaluated (independent of CONTROL).

Since Braginskii's theory[13] provides all these coefficients as functions of $\omega_I\tau_I$, $\omega_e\tau_e$ and $\mathcal{Z}$, the evaluation scheme also provides arrays in common blocks which show $\omega_e\tau_e$, $\omega_I\tau_I$, τ_e, τ_I and Λ, the Coulomb logarithm, at each cell center. The $\mathcal{Z}$ dependence tabulated by Braginskii has been parameterized by power laws in $\mathcal{Z}^{-1}$. These fits are implemented in the

companion routines BRALPHA, BRBETA, BRGAMMA, and BRDELTA, evaluating $\{\alpha_1'(\mathcal{Z}), \alpha_0'(\mathcal{Z}), \alpha_0''(\mathcal{Z})\}$, $\{\beta_1'(\mathcal{Z}), \beta_0'(\mathcal{Z}), \beta_0''(\mathcal{Z})\}$, $\{\gamma_1'(\mathcal{Z}), \gamma_0'(\mathcal{Z}), \gamma_0''(\mathcal{Z})\}$, and $\{\delta_0(\mathcal{Z}), \delta_1(\mathcal{Z})\}$ respectively. The constant values α_1'', β_1'', γ_1'' are also returned in the various output arrays. All of these $\mathcal{Z}$ dependent functions can be represented to reasonably accuracy ($\sim$2%) by expressions of the form $c_0 + c_1 (\mathcal{Z})^{-\lambda}$ on $[1, +\infty]$ in $\mathcal{Z}$. For the purposes of transport coefficient evaluation $\mathcal{Z}$ is restricted to be greater than 1 and so also is the Coulomb logarithm. However the systematic modification of τ_e and τ_I allows the extension of the theory and the code into domains of stronger coupling (in the plasma parameter $n\lambda_D^3$). Once the appropriate relaxation times are available the restriction on the Coulomb logarithm can be relaxed.

The modification of τ_e due to turbulence in the low density, drift speed limited regions is also a natural temptation. It represents the simplest way of obtaining a consistent and systematic transport package for turbulent loads. At present it has not been done because (*i*) the model of turbulent relaxation by a Fokker-Planck kinetic equation is questionable on many grounds, (*ii*) the magnetic field effects in turbulent relaxation are not likely to be accurately represented by Braginskii's functions, and (*iii*) the proper treatment of turbulent transport will depend on isolating the detailed properties of those microinstabilities peculiar to the cylindrical, highly-inhomogeneous plasma. Those properties, which can be calculated only when the background quasi-equilibria are available, depend on the results of the present model and cannot be specified a priori. The marginal-stability criterion $J \leq en_e c_s$ circumvents these difficulties and is thus used in place of τ_e modifications.

E. Fluid Density and Flow Field Time Derivatives

The data base for all the hydrodynamic calculations is the common block FLUID-STATE, containing $\{N_j, {}^i_cT_{I,j}, {}^i_c\theta_{e,j}, {}^ir_j, {}^i\dot{r}_j\}$. An update of the FLUID-STATE is the central result of a major time step. The relationships among the basic fluid variables on the Lagrangian mesh are illustrated in Figure 4. The simple two and three point area-weighted differencing schemes for such a mesh are discussed in Appendix III. The cell ions $\{N_j\}$ are a conserved vector of ions/cm resident in the (compressible) cell $\left[{}^ir_j, {}^ir_{j+1}\right]$, assuring strict particle conservation and a solution of the equation of continuity limited in accuracy only by the evolved values of $\{{}^i\dot{r}_j\}$. The elements of $\{N_j\}$ are assigned spatial locations given by the cell center position (defined by the equal area point) ${}^i_cr_j = \sqrt{\frac{1}{2}({}^ir_j^2 + {}^ir_{j+1}^2)}$. They change only if a regridding is called for.

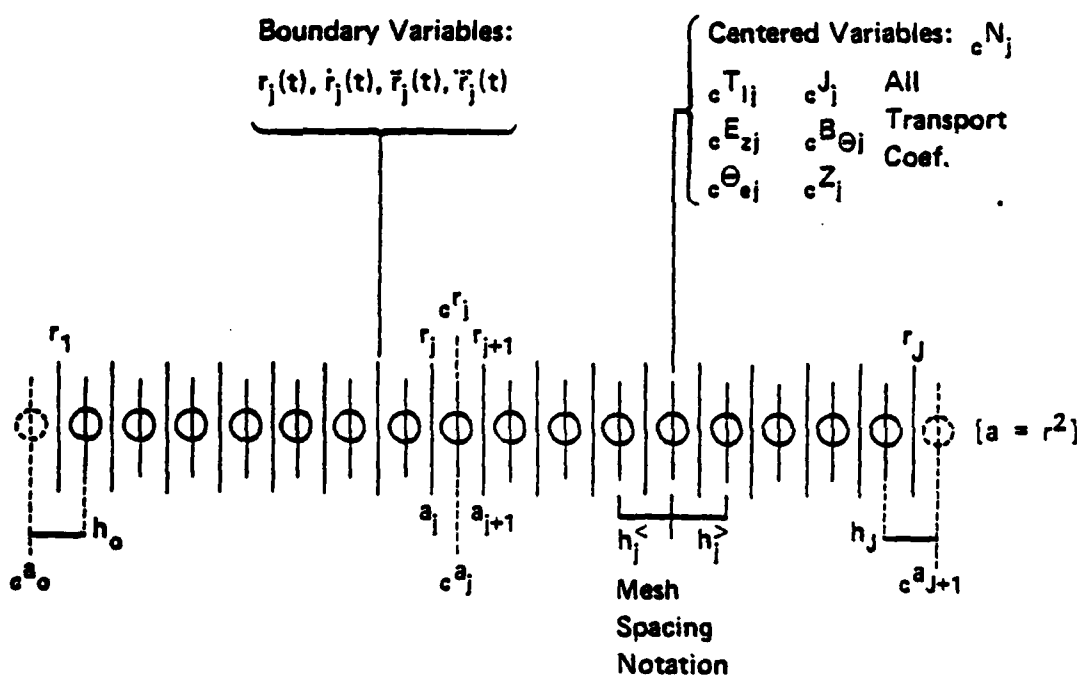


Figure 4. The Lagrangian Fluid Mesh

When needed, the stresses at the cell interfaces are calculated by the subroutine MESHTRESS. This code has two modes of operation. In the usual mode, during a subcycle, the acceleration of the interface $^{*}\ddot{r}_j$ is computed. But at a major time step, the initiation of the subcycle and the calculation of the Z field sources requires an explicit calculation of the jerk $^{i}\dddot{r}_j$ as well. The fluid stress has three major components as noted in Eq. (III.3) above. Each of these stress components: pressure gradient, magnetostrictive, and ambipolar electrostatic, has a particular realization in terms of area weighted finite differences (or smooth interpolants if more precision is needed) that is best suited to the mathematical and programming requirements of the problem.

First, the pressure gradient stresses are most naturally derived from the mesh structure shown in Figure 4. Using the $\{N_j\}$ and $\{^{i}a_j\}$ the mean fluid density at the cell center $^{i}_{c}n_{Ij} = N_j/(a_{j+1} - a_j)\pi$ is the quantity determining the local inertia of the fluid. The pressure gradient stress is thus easily shown to be estimated (to $O(\Delta r^2)$) by

$$\ddot{r}_j \,|_{fluid} = -\left(2\, r_j \frac{\tilde{\tilde{T}}_j}{\tilde{\tilde{m}}}\right)\left\{\frac{\ln n_{Ij} - \ln n_{Ij-1}}{^{c}a_j - {}^{c}a_{j-1}}\right\} - (2\, r_j/\tilde{m})\, \frac{^{c}\tilde{T}_j - {}^{c}\tilde{T}_{j-1}}{^{c}a_j - {}^{c}a_{j-1}} \qquad \text{(III.5)}$$

where the area-weighted interface temperature $\tilde{\tilde{T}}_j$ is defined by (cf. Appendix III also):

$$\bar{\bar{T}}_j = \left\{ \left(\frac{{}_c a_j - a_j}{{}_c a_j - {}_c a_{j-1}} \right) {}_c\tilde{T}_{j-1} + \left(\frac{a_j - {}_c a_{j-1}}{{}_c a_j - {}_c a_{j-1}} \right) {}_c\tilde{T}_j \right\} \qquad \text{(III.6)}$$

for j on the index domain [2, J] with J the maximum number of cells. If one insists, as in Section III.G, on a null heat flow across the outermost boundaries r_1 and r_{J+1}, then the natural boundary condition on this pressure gradient stress is to neglect the temperature difference term and simply extrapolate the logarithmic density gradient from boundary 2 and boundary J. The boundary pressure gradient stresses are then estimated by

$$\ddot{r}_1 \,|_{fluid} = -\,2\, r_1 \left\{ \frac{\ln n_{I2} - \ln n_{I1}}{{}_c a_2 - {}_c a_1} \right\} {}_c\tilde{T}_1 \qquad \text{(III.7a)}$$

$$\ddot{r}_{J+1} \,|_{fluid} = -2\, r_{J+1} \left\{ \frac{\ln n_{IJ} - \ln n_{IJ-1}}{{}_c a_J - {}_c a_{J-1}} \right\} {}_c\tilde{T}_J \quad . \qquad \text{(III.7b)}$$

Next one must estimate the magnetostrictive stresses using ${}_cJ_j$ and ${}_cB_j$ and suitable area weightings. The same averaging operation shown in (III.6) provides the formulation

$$\ddot{r}_j|_{JB} = -\overline{\left(\frac{J_z B_\theta}{c\, \tilde{m}\, n_I} \right)}_j \qquad \text{(III.8a)}$$

with the boundary conditions

$$\ddot{r}_1|_{JB} = -\left(\frac{{}_cJ_{z,1} \cdot {}_cB_{\theta,1}}{c\, \tilde{m}\, {}_cn_{I,1}} \right) \qquad \text{(III.8b)}$$

$$\ddot{r}_{J+1}|_{JB} = -\left(\frac{{}_cJ_{z,J} \cdot {}_cB_{\theta,J}}{c\, \tilde{m}\, {}_cn_{I,J}} \right) \quad . \qquad \text{(III.8C)}$$

The ambipolar stresses are somewhat more delicate in that their sources are drawn from both zone boundaries and zone centers. The strict application of the radial Ohm's law results in an integral equation for the relative charge separation at any spatial location. To a good approximation however one may neglect the recursive nature of that relation and simply evaluate the radial component of E at the cell boundaries in a quasi-static sense at any time level. In terms of the axial electron drift speed u_z and the (dimensionless) Braginskii transport coefficients $\tilde{\alpha}_\wedge$ $\tilde{\beta}_\perp$ this electric field is

$$E_{j,r} = -\left\{\left[\frac{m_e}{e}\frac{\tilde{\alpha}_\wedge u_z}{\tau_e} + \frac{\tilde{\beta}_\perp}{e}\,\partial_r T_e\right] + \frac{1}{e}\left[\partial_r\left(T_e - \frac{m_e}{m_I}T_I\right) - \frac{m_e}{m_I}T_I\,\partial_r \ln n_I + T_e\,\partial_r \ln \mathcal{Z} n_I\right]\right\} \qquad \text{(III.9)}$$

and can be defined on the cell boundaries easily, differencing for the gradients in n_e, n_I and $T_e - \frac{m_e}{m_I}T_I$ to the boundaries directly and averaging the strictly central terms to the intermediate boundary. The boundary conditions on E_{jr} are similar to those on the magnetostrictive stresses with the addition that, in compatibility with the thermal conduction algorithm discussed below, the temperature gradient source terms are assumed to vanish.

The ambipolar space charge is hence $\rho = \frac{1}{4\pi}\nabla \cdot E_r$ with E_r given on the cell boundaries. But ρ is most naturally computed at the cell centers by means of

$$c^{\rho}{}_j = \frac{1}{2\pi}\,\frac{r_{j+1}E_{j+1,r} - r_j E_{j,r}}{a_{j+1} - a_j} \quad . \qquad \text{(III.10)}$$

Forming the effective fluid charge-to-mass ratio at the cell centers, the final ambipolar stress is given by

$$\ddot{r}_j\Big|_A = E_{j,r} \cdot \overline{\left(\frac{c^{\rho}}{\bar{m}_c n_I}\right)}_j \tag{III.11a}$$

with boundary conditions

$$\ddot{r}_1\Big|_A = E_{1,r} \left\{\frac{c^{\rho_1}}{\bar{m}_c n_{I,1}}\right\} \tag{III.11b}$$

$$r_{J+1}\Big|_A = E_{J+1,r} \left\{\frac{c^{\rho_J}}{\bar{m}_c n_{I,J}}\right\} \tag{III.11c}$$

With these expressions, the final output of MESHTRESS is the sum

$$\ddot{r}_j = \ddot{r}_j\Big|_{fluid} + \ddot{r}_j\Big|_{JB} + \ddot{r}_j\Big|_A \quad ;$$

but, at the major time levels when the jerk is needed the subroutine goes further and computes

$$\dddot{r}_j = \dddot{r}_j\Big|_{fluid} + \dddot{r}_j\Big|_{JB} + \dddot{r}_j\Big|_A$$

by means of higher-order difference formulae similarly derived, and data produced by the fluid temperature time derivative calculations discussed in the next section.

F. Fluid Temperature Time Derivatives

In order to compute the heating rates at the cell centers from the FLUID-STATE, several distinct steps must be taken. These distinct calculations are, however, collected in the single subroutine TDOT. First one must recover the electron temperature, mean ionization state and chemical potential. This is done by the subroutine CREOS (Collisional Radiative Equation Of State) to which is input the ion density (${}^{*}_{c}n_{I,j}$) and the electron "grand canonical temperature" (${}^{*}_{c}\theta_{e,j}$). Using the smooth interpolants discussed in Section (II.D.ii), CREOS first obtains the logarithm of the branching ratio and calculates using (II.9a)

$$\ell n\ ({}^{*}_{c}T_{e,j}) = \ell n\ ({}^{*}_{c}\theta_{e,j}) - \ell n\ ({}^{*}_{c}b_{j}), \tag{III.12}$$

from which ${}^{*}_{c}T_{e,j}$ is obtained through exponentiation. Both ${}^{*}_{c}\bar{Z}_{j}$ and ${}^{*}_{c}\epsilon_{I,j}$ are then obtained using (II.9b,c) with the logarithmic argument implied by the relation (III.12). In addition, CREOS calculates the derivatives of the branching ratio with respect to θ_e and n_I so that one may calculate material derivatives of θ_e and T_e using

$$\dot{T}_e = \frac{\dot{\theta}_e}{b} - T_e \left\{ \left(\frac{\partial\ \ell n\ b}{\partial\ \ell n\ \theta} \right) \frac{\dot{\theta}_e}{\theta_e} + \left(\frac{\partial\ \ell n\ b}{\partial\ \ell n\ n_I} \right) \frac{\dot{n}_I}{n_I} \right\} . \tag{III.13}$$

The material derivatives of T_e are needed for the evaluation of the jerk as well as for diagnostic purposes. If the jerk evaluation is being done,

then CREOS continues with a calculation of $\partial^2 \ln b/\partial(\ln \theta_e)^2$ and $\partial^2 \ln b/\partial(\ln n_I)^2$ as well.

The next operation is the establishment of the transport coefficients on the mesh using the data base $\{{}^*_c B_{\theta,j}, {}^*_c J_j, {}^*_c n_{I,j}, {}^*_c T_{I,j}, {}^*_c T_{e,j}, {}^*_c \mathcal{J}_j\}$ supplied in part by CREOS. This procedure is complicated by the need to iterate on ${}^*_c B_{\theta j}$ (through ${}^*_c J_j$) because the conductivity ${}^*_c \Sigma_j$ depends on the magnetic field. The details of this iterative refinement are discussed in Section III.G(ii) in the appropriate context of the subcycle algorithm. In essence, any call to BRAGINSKII, given the data base defined above, will produce the needed transport coefficients to go further.

The establishment of the transport coefficients and temperatures is necessary to calculate the thermal diffusion rates. This calculation is presently done in conjunction with the compression heating (${}^*Q_{(pdV)_j} = -{}^*_c(\nabla \cdot \dot{r})_j \, {}^*_c T_j$) by the subroutines VTGRADS and VTDIFDER. The module VTGRADS (Velocity and Thermal GRADient Sources) produces arrays containing the area meshes $\{{}^*a_j, {}_c{}^*a_j\}$ and the discrete velocity, temperature and conductivity data $\{{}^*_c T_{ej}, {}^*_c \chi_{ej}, {}^*_c \chi_{Ij}, {}^*_c T_{Ij}, {}^*\dot{a}_j\}$ in a suitable form for use with either the smooth interpolation package or the area-weighted differencing module VTDIFDER (Velocity and Thermal DIFferenced DERivatives). The boundary conditions enforced by VTDIFDER are those of null heat flow across the outer boundaries ${}^*r_1$ and ${}^*r_{J+1}$. This is done by means of ghost points playing the role of cell centers ${}^*_c r_o$, ${}^*_c r_{J+1}$ (cf. Fig. 4) and the use of simple variable-mesh three-point differencing formulae. This yields the output array TDOTGF: $\{{}^*_c(\partial_a^2 T_e)_j, {}^*_c(\partial_a T_e)_j, {}^*_c(\partial_a \chi_e)_j, {}^*_c(\partial_a \chi_I)_j, {}^*_c(\partial_a T_I)_j, {}^*_c(\partial_a^2 T_I)_j, {}^*_c(\partial_a \dot{a})_j\}$. These results are used in calculating the thermal diffusion rates

$$ {}^*(Q_e)_j = {}^*(4 \chi_\perp^e \partial_a T_e + 4a(\partial_a \chi_\perp \partial_a T_e + \chi_\perp \partial_a^2 T_e))_j , \qquad \text{(III.14a)} $$

$$*(Q_I)_j = *(4\,\chi_\perp^I\,\partial_a\,T_I + 4a\,(\partial_a\,\chi_\perp^I\,\partial_a\,T_I + \chi_\perp^I\,\partial_a^2\,T_I))_j \qquad \text{(III.14b)}$$

The compression rate ($\nabla \cdot V$) is of course the last item listed above in TDOTGF.

With the gradients available, one may now evaluate the thermoelectric fields and calculate a local drift speed for the conduction current J_z using the free electron density, the complete E field $\hat{E}_z = E_z' + E_{th}$, and the classical conductivity σ. At each cell center this drift speed is compared to the local sound speed; if greater, it is lowered to the marginal stability limit $u_z \sim c_s$. With $\hat{E}_z$ fixed, the drift-speed limit thus imposes a decrease in the effective conductivity of any cell. This new conductivity is set such that $\hat{E}_z \cdot \sigma_{turb} = en_e c_s$, and it replaces the classical conductivity orginally computed. The ohmic heating is also radically altered when the drift speed limit is imposed. It is calculated here by means of the product $J_z \cdot E_z'$, which is explicitly indpendent of the conductivity but reflects it in any limitations on J_z. The usual heating rates are thus completely specified by the calculations summarized thus far; only the radiative coupling remains.

The radiative energy transport calculation must be done for each radiative category one wishes to treat. At present a vector of emission rates ${}^*_c e_j$ is calculated by the function CREMIT for each of four classifications: line radiation $h\nu < 1$ keV, line radiation $h\nu > 1$ keV, free-bound continuum, free-free continuum. This function contains preprocessed fits to the CRE model emission strengths in lines and free-bound continuum, and a simple free-free emission model. A companion function DAMPIT calculates a vector of damping rates ${}^*_c d_j$ using attenuation estimates appropriate to each radiation category. Once the data base $\{{}^*_c e_j,\ {}^*_c d_j,\ {}^*r_j,\ {}^*_c\Delta a_j = {}^*a_{j+1} - {}^*a_j\}$

is computed for a photon in category p, one may calculate a coupling matrix for this category. The coupling matrix is related to the directional derivative of the probability of escape $P_E(p, {}^*_c d_{jp} | \ell_{ij})$, where ℓ_{ij} is the path length from cell i to cell j along a chord inclined to the radius vector through cell i (Apruzese[14]). P_E is a nonlinear function of the path-integrated optical depth, depending upon the radiation category's damping mechanism as well as the local attenuation parameter values. The line radiation categories use P_E functions based on a Doppler line profile, while the continuum radiation categories use a P_E which is exponential with optical depth. The ℓ_{ij} represent geometric projection factors needed in the (discrete) path integral estimate and can be calculated for all radiation categories once the mesh ${}^*r_j$ is given. The coupling matrix can therefore be defined as

$$C_{ijp} = P_E(p, \int_{x_i}^{x_j^+} dx\tau(d(x))) - P_E(p, \int_{x_i}^{x_j^-} dx\tau(d(x))) ,$$

where x is a path distance and $\Delta x_j = x_j^+ - x_j^-$ is the cell thickness, along the inclined chord. In terms of the coupling matrix, the energy lost or gained by a cell is the inner product

$$Q_{\text{photo } j} = \sum_p ({}^*_c e_{pj} - \sum_i C_{ijp} \, {}^*_c e_{pi}) ;$$

while the energy lost from the plasma, the observable output from any cell in any radiation category, is the residual sum

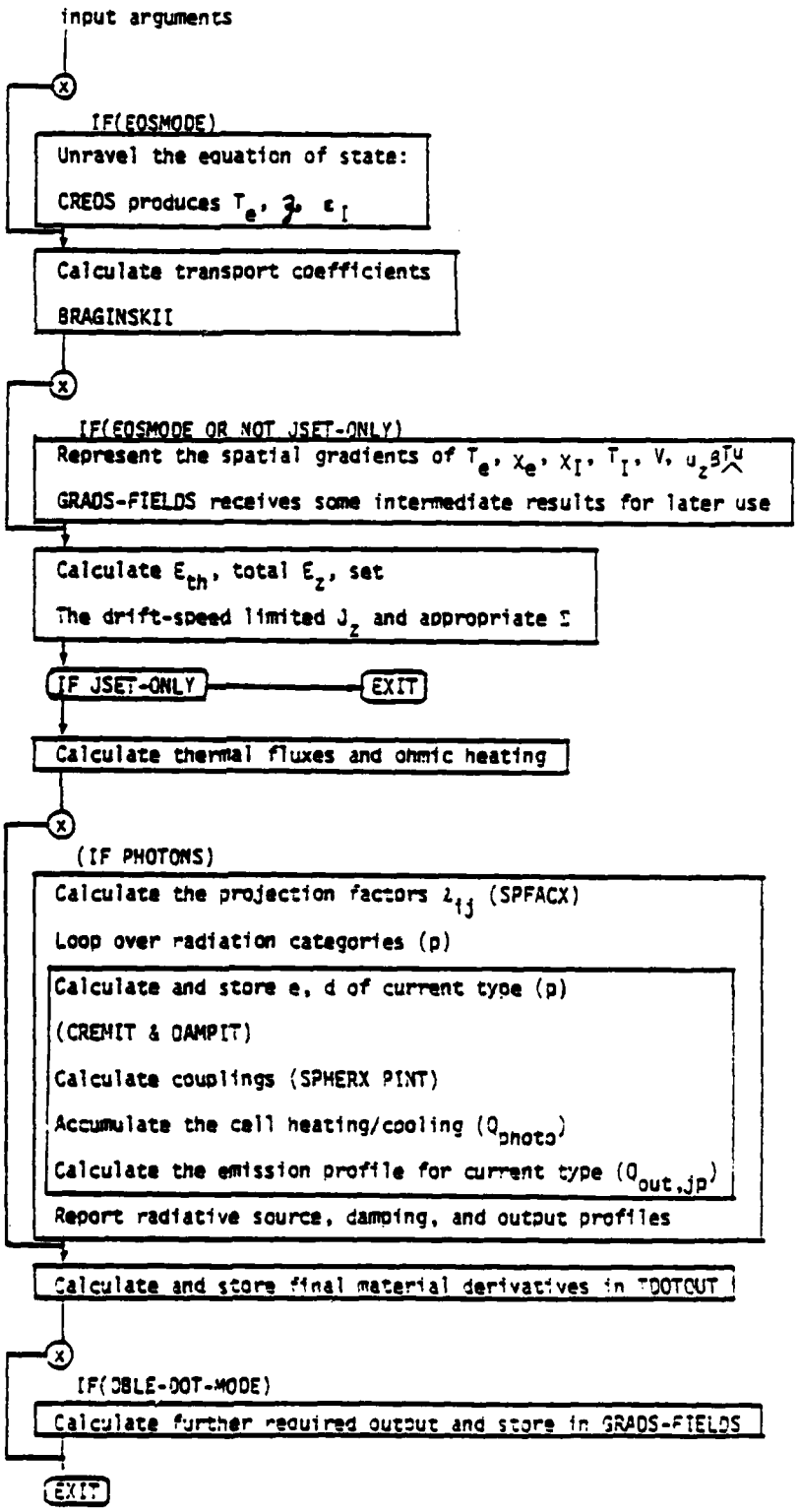
$$Q_{out,jp} = {}^*_c e_{pj} \cdot (1 - \sum_\ell C_{\ell jp}) \quad .$$

These operations are done with the aid of three subroutines, SPFACX, SPHERX, and PINT due to R. W. Clark[15]. The first computes the projection factors ℓ_{ij} from the given mesh $^*r_j$ for any particular angle of inclination of the chord through each cell. The second and third act in concert to produce the coupling matrix from the data ${}^*_c d_{jp}$ and ℓ_{ij}. The subroutine PINT produces required path integrals of escape probability in either Doppler or exponential mode, while SPHERX produces the coupling matrix.

The subroutine TDOT contains all of these calculations but provides options for bypassing those not needed for the iterative refinement of the magnetic field. These mode switches are contained in the common block TRANSPORT-CHOICE (along with the CONTROL matrix for subroutine BRAGINSKII) as the logical variables: PHOTONS, JSET-ONLY, EOSMODE, and DBLE-DOT-MODE. Of this set, DBLE-DOT-MODE is referenced by CREOS as well. The flow of the subroutine TDOT is illustrated in Fig. 5, and is essentially the same sequence of operations discussed above if no bypasses are taken. The required input is $\{{}^*_c\theta_{ej}, {}^*_c T_{Ij}, {}^*_c\dot{r}_j, {}^*\dot{r}_j, {}^*_c n_{I,j}, {}^*_c r_j, {}^*r_j, {}^*_c E_j, {}^*_c B_j\}$.

The output is divided among the common blocks TDOTOUT, containing: $\{{}^*_c\dot{n}_{I,j}, {}^*_c\dot{T}_{I,j}, {}^*_c\dot{T}_{e,j}, {}^*_c\theta_{e,j}$, and $^*Q_{rad,jp}\}$ and GRADS-FIELDS, containing temperature gradients, compression rates, thermoelectric fields, total plasma electric fields, classical or limited axial current densities, and (if DBLE-DOT-MODE is .TRUE.) information required by MESHTRESS and HERTZDOTS for higher order time derivatives.

FIGURE 5 - Flow of Subroutine TDOT



G. The Subcycle Calculating Thermal and Electrodiffusive Evolution in the Model Plasma

A subcycle is preferred for thermal conduction (and other dissipative processes) because the parabolic p.d.e. describing the evolution is often treated best by an implicit method, while the momentum advance can often be adequately treated by an explicit method. Also, from a physical viewpoint, the fastest timescales of the model plasma are those associated with electron/ion thermal exchange, (unmagnetized) electron thermal conduction and radiative losses (under conditions of strong compression). The solution of the thermal evolution relations (III.4. a, b) for a fixed level of accuracy will therefore require more time derivative evaluations than an equivalent solution of the velocity field relation (III.3), even allowing for some iterative refinement of the mesh motion. Finally, as a matter of flexibility in algorithm choice, the subcycle allows one to separate the thermal evolution from the mesh evolution within the code and, therefore, to select possibly distinct time-advance mechanisms appropriate to each.

The thermal subcycle presently implemented derives from the observation that a spacetime p.d.e. which has the derivative operations represented by discrete differences on one domain of dependence appears as a coupled set of o.d.e. on the orthogonal domain. The particular case here is that of spatial differential operators, corresponding to finite difference operators (derived through conventional or smooth interpolants), producing a set of coupled equations on the time domain. In all such cases the coupling of the time derivatives is expressed as a Jacobian matrix, implicitly dependent on the discrete mesh used for the derivative operators. Here the time derivatives are the material derivatives $\{\dot{T}_e, \dot{\theta}_e, \dot{E}_z\}$ and the underlying mesh motion $\{r, \dot{r}, \ddot{r}, \dddot{r}\}$ is completely transparent to the subcycling algorithm. The three fundamental codes for this process are described below.

(i) GEARBOX, the subcycle integrator

Based on a multi-time-level variable order scheme, the subroutine GEARBOX performs the thermal subcycle on a time domain [Tlo, Thi] defined externally. The input consists of ${}^{i}Y_j = \{\ldots, {}^{i}_{c}T_{I,j}, {}^{i}_{c}\theta_{ej}, {}^{i}_{c}T_{I,j+1}, {}^{i}\theta_{e,j+1}\ldots\}$, initial data for a total of 2·J coupled equations of motion. The time-dependent mesh (with which the thermal fluxes, compressional heating and radiative cooling are computed) is specified implicitly over this interval for Tlo < τ^* < Thi. The forward integration of $\{Y_j\}$is done by the IMSL subroutine DGEAR, using the material derivative subroutine FLUIDOTS and the coupling matrix (Jacobian) produced by JACOB. In the environment seen by DGEAR,the problem is specified completely by the input $\{{}^{i}Y_j\}$, the interior material derivatives $\{{}^{*}\dot{Y}_j\}$, the coupling matrix $^{*}(\partial\dot{Y}_i/\partial Y_j)$, and the intermediate time points τ^*. Values of τ^* are chosen by the integrator, in accordance with the rates of change produced by FLUIDOTS, and these values are thus the sole input arguments available for the specification of the Lagrangian mesh $^{*}r_j$.

To summarize the operation, GEARBOX begins by checking for an upper limit in the explicit timestep supplied as input whenever the integrator is being initialized. Next the explicit time limits set as arguments are stored in the common block <TIMEBASE> for use by FLUIDOTS and JACOB in implicitly advancing the fluid mesh. If desired, a short report on the model thermodynamic variables is written. The actual integration is effected by a call to DGEAR, and upon completion various error flags are checked and the severity of any integrator errors is assessed. If desired another short report on the evaluation is produced; and, if a terminal integrator error is found a completed dump of all pertinent information is triggered. The subroutine returns with a new (provisional) vector $\{{}^{i+1}Y_j\}$ overwriting

the original input. The advanced temperatures are either accepted or iterated upon in the calling routine, depending on the chosen option for the fluid mesh evolution.

The integrator DGEAR offers several control parameters and considerable flexibility in the integration scheme. The most important options are the use of explicit Jacobians or estimated Jacobians, and the control of the local truncation error through a variable TOL. The explicit Jacobian is apparently the most favorable option in this hydrodynamic application, but if the application of more detailed physics demands a very expensive evaluation for this matrix, an estimate based only on finite differences is adequate. The variable TOL provides a natural and consistent means of specifying time integration parameters for the mesh evolution as well, cf. Sections (III.H.i & ii). The integration scheme can be selected as either an implicit Adams method of up to twelfth order or a backward differentiation method of up to fifth order (Gear's stiff method). Both methods are of the implicit linear multilevel type and require the solution of an algebraic system at each interior (subcycle) timestep. The basis for an optimal choice between the two lies in the performance of several benchmark calculations discussed in Section III.I below.

(ii) FLUIDOTS, the material derivatives

The physical content of the problem is defined entirely in the subroutine FLUIDOTS, which incorporates an explicit time advance of the data $\{*Z_k, *\partial_\tau Z_k, *r_j\}$ in order to define the mesh and electromagnetic fields required to compute $*\dot{Y}_j$. In summary, FLUIDOTS receives as input $\{*Y_j\}$ and $\tau*$, which is any interior point selected by DGEAR on the <TIMEBASE> for the calculation of a derivative. First FLUIDOTS maps $*Y_j$ to separate variables $*T_{Ij}$, $*\theta_{ej}$. Next the mesh is obtained by a forward Taylor series using

whatever information resides in the common block MESHADVDATA:

$$^{*}r_j = {}^{i}r_j + (\tau^* - T1o)\,{}^{i}\dot{r}_j + \frac{1}{2}(\tau^* - T1o)^2\,{}^{i}\ddot{r}_j + \frac{1}{6}(\tau^* - T1o)^3\,{}^{i}\dddot{r}_j$$

$$^{*}\dot{r}_j = {}^{i}\dot{r}_j + (\tau^* - T1o)\,{}^{i}\ddot{r}_j + \frac{1}{2}(\tau^* - T1o)^2\,{}^{i}\dddot{r}_j \quad .$$

This mesh is checked for inversion $\{^{*}r_j > {}^{*}r_{j+1}\}$ and if such inversion occurs a terminal error condition is generated and a dump of all pertinent information is obtained. The mesh is also checked for closure of the annulus ($^{*}r_1 \leq 0$) and, if closure is sensed during a subcycle, the motion of the innermost cell boundary is corrected, $^{*}r_1 \rightarrow 0$ & $^{*}\dot{r}_1 \rightarrow 0$. Once these operations are complete an array of cell centers is generated by equal areal partition and a central velocity $^{*}_{c}\dot{r}_j$ is assigned by area weighting. A cell ion density is also calculated from the (conserved) particles resident in the cell.

The switches EOSMODE and JSET-ONLY are then set for a complete TDOT calculation. If electromagnetic effects are turned off this single TDOT calculation proceeds. If, as is usually the case, the E_z and B_θ are needed then JSET-ONLY is set .TRUE. and ZPUSH advances the Hertz potential from its last data base $\{^{i}Z_k, {}^{i}\partial_\tau Z_k\}$ to its values at τ^*. Given the results of ZPUSH, the subroutines EFIELD and BFIELD are called to establish E_z completely and B_θ partially, using the current densities of the previous τ^* point. Using the provisional $^{*}_{c}B_{\theta j}$, a call to TDOT with EOSMODE and JSET-ONLY set .TRUE. recalculates the central current densities at fixed E_z and estimated Σ. A second call to BFIELD uses the new current densities and thus implicitly refines $\Sigma(B)$. After one refinement pass (TDOT; BFIELD), EOSMODE is set .FALSE. and the initial calculation of $^{*}_{c}\epsilon_{Ij}$,

${}^{*}_{c}\mathcal{J}_j$, and ${}^{*}_{c}T_{ej}$ remains in place. The gradient calculation remains undisturbed as well so that the E_z thermoelectric field component will change only through the refinement of its B_θ-dependent coefficient. The refinement passes continue until $\{{}^{*}_{c}B_{\theta j}\}$ convergence is achieved to a specified tolerance or until a maximum number of passes has been made. Then JSET-ONLY is set .FALSE. and a final call to TDOT calculates the full complement of material derivatives and transport coefficients using the self-consistent magnetic field values. In this final pass the gradients are recomputed because the thermal conduction has changed with the magnetic field refinement. If magnetic field convergence is not achieved to the desired accuracy, e.g. $o(10^{-4})$, the vector of fractional changes at the last iteration is output as a diagnostic, and if desired a full report of the calculation is output as well.

The last operations performed by FLUIDOTS are to map ${}^{*}_{c}\dot{\theta}_{ej}$, ${}^{*}_{c}\dot{T}_{I,j}$ to the single variable ${}^{*}\dot{Y}_j$ and to update the acceleration and jerk arrays ${}^{i}\mu_j$, ${}^{i}\dot{\mu}_j$ if required. An update of these variables, contained in <MESHADVDATA>, is triggered within the subcycle whenever τ^* is larger than any previous time argument requested $\tau^* > \tau_{last}$, or if $\tau^* = Thi$. As discussed in the following Section III.H, this enables one either to correct the mesh evolution smoothly as temperatures evolve or to set up for a complete iterative refinement of the mesh evolution. The mesh update occurs with a call to MESHTRESS and, if DBLE-DOT-MODE is .FALSE. as is the case during a subcycle, it involves only the explicit recalculation of the acceleration array ${}^{*}\mu_j$. The jerk array is continually updated as

$$ {}^{*}\dot{\mu}_j = \left(\frac{{}^{*}\mu_j - {}^{i}\mu_j}{\tau^* - Tlo} \right) \quad , \quad \tau_{last} = \tau^* $$

for $\tau^* > \tau_{last}$. When FLUIDOTS is called outside a subcycle, DBLE-DOT-MODE

is set .TRUE. and an explicit evaluation of the jerk array ${}^{i}\dddot{r}_j$ is obtained as well.

(iii) JACOB, the link to the original p.d.e.

The coupling matrix calculated by JACOB is a simple if lengthy exercise in the use of partial derivative chain rules. In the case of a simple three point difference formulae this matrix is slightly sparse and pentadiagonal. The compression and exchange heating create a block-tridiagonal band dependent on $(\nabla \cdot V)$ and the local T_e, while the thermal conduction operators add two more bands and some diagonal contributions. In the case of a smooth interpolant (G or E-spline) the matrix contains more nontrivial bands of monotonically decreasing importance away from the diagonal. The rate of decay is determined by the coupling parameter choice in the underlying interpolant.

However one defines the differentiation process, the general form of the Jacobian is easily derived from the structure of the subcycled array $\{Y_j\}$. For all entries involving the material derivative of θ_e the local value of the branching ratio is the element of central interest, i.e.

$$\frac{\partial \dot{\theta}_\ell}{\partial \theta_m} = \frac{1}{b_m}\left(1 - \frac{\partial\, \ell n\, b}{\partial\, \ell n\, \theta}\right)_m \cdot \frac{\partial \dot{\theta}_\ell}{\partial T_{e,m}} \quad ,$$

while only on the first subdiagonal does $\dot{\theta}$ couple to T_I, since

$$\frac{\partial \dot{\theta}_j}{\partial T_{Ij}} = \frac{2m_e}{m_I \tau_{e,j}} \quad ,$$

apart from some radiative contributions due to the cell's self-opacity.

In a similar manner the entries involving the material derivative of T_I couple to θ only on the first super-diagonal, i.e.

$$\frac{\partial \dot{T}_j}{\partial \theta_j} = \frac{1}{b_j} \cdot \left(1 - \frac{\partial \ln b}{\partial \ln \theta}\right)_j \cdot \frac{2\gamma m_e}{m_I \tau_{e,j}} \quad .$$

The coupling on the main diagonal consists of several contributions. The primary one is compressional heating ($-\frac{2}{3}\nabla\cdot V$), with exchange losses and the term from the central coefficient of whatever gradient formula is employed in computing the heat flux also important. In the case of $\dot{\theta}_e$ a radiative diagonal contribution also arises.

The general Jacobian structure obtained for a three-point difference scheme is illustrated in Fig. 6. There compressional heating is denoted by θ on the diagonal; the ion-electron exchange terms are shown by x; and the contributions from the spatial differencing in thermal conduction are denoted by Δ.

Figure 6 - A simple Jacobian matrix

	T_{Ij}	θ_{ej}	T_{Ij+1}	θ_{ej+1}	T_{Ij+2}	θ_{ej+2}
$\dot{Y}_{2j-1}(\dot{T}_{Ij})$	$\Delta\theta X$	X	Δ			
$\dot{Y}_{2j}\ (\dot{\theta}_{ej})$	X	$\Delta\theta X$		Δ		
$\dot{Y}_{2j+1}(\dot{T}_{Ij+1})$	Δ		$\Delta\theta X$	X	Δ	
$\dot{Y}_{2(j+1)}(\dot{\theta}_{ej+1})$		Δ	X	$\Delta\theta X$		Δ
$\dot{Y}_{2j+3}(\dot{T}_{Ij+2})$			Δ		$\Delta\theta X$	X
$\dot{Y}_{2(j+2)}(\dot{\theta}_{ej+2})$				Δ	X	$\Delta\theta X$

The inclusion of radiation terms is dependent on the coupling matrix summations discussed above in computing Q_{photo}. It is a straightforward but more complex calculation along these same lines.

(iv) Modifications for E_z diffusion

If the option of diffusive evolution for E_z is desired, an alternate trio of subroutines is employed. The vector Y_j now becomes $\{\ldots {}^{*}_{c}T_{Ij}, {}^{*}_{c}E_{zj}, {}^{*}_{c}\theta_{ej}\ldots\}$ in analogy to the structure used in GEARBOX. This implies a Jacobian which is block pentadiagonal with further bands added depending on the spatial differencing. The routine GEARBOX is replaced by TETGEAR; FLUIDOTS, by TETDOTS; and JACOB, by TETJAC. The routine TETDOTS contains most of the principal differences; it must call an additional subroutine EDOT which evaluates equation (III.2). The iterative refinement of B_θ appearing in FLUIDOTS persists, but the external circuit equation requires an iterative refinement of ${}^{*}_{c}\dot{J}_{zj}$ and ${}^{*}_{c}\dot{E}_{zj}$ together in order to achieve a self-consistent prescription of the boundary conditions for equation (III.2). Once the values of ${}^{*}_{c}\dot{T}_{ej}$, ${}^{*}_{c}\dot{\mu}_{j}$ are established by TDOT and MESHTRESS, however, all iterative refinement can be made internal to EDOT. The flow of TETDOTS is thus quite similar to that of FLUIDOTS with an additional call to EDOT prior to the final return.

H. The Complete Fluid Evolution Package - HYDROPUSH

All of the foregoing algorithms are combined in the subroutine HYDROPUSH, which forms the nucleus of Program ZDIPR—apart from startup, diagnostic and graphics modules. HYDROPUSH is concerned exclusively with the

advance of the <FLUID-STATE> over the variable major time step which it selects as appropriate. The only required inputs are the common blocks <FLUID-STATE> and <FIELDADVDATA> (containing the previous time slice of fluid variables and field variables) and the appropriate mesh dimension, scale factors and physical constants, obtained from a variety of sources. The output is an update of <FLUIDSTATE>, and an increment of the variable MSI (Main Step Index) by 1. If a restart file is desired, it is created for later use and assigned a record number equal to the MSI. If any severe errors occur in the advance, dump files are created to allow examination of intermediate results, and a variety of reports at intermediate phases of the calculation are available as general diagnostics. The architecture of HYDROPUSH assumes that, after the fluid advance is successful, the main program will call ZPUSH separately to advance the fields ${}^{i}Z_k$ and ${}^{i}\partial_\tau Z_k$ - unless the electrodiffusive approximation is being used. In that case HYDROPUSH will also advance the E_z field by means of TETGEAR.

The sequence of processing begins by reading elements of the <FLUID-STATE> into <MESHADVDATA> and the subcycle vector $\{{}^{i}Y_j\}$. Once the appropriate data base is inferred a full complement of material derivatives is computed by FLUIDOTS/HERTZDOTS or TETDOTS. These material derivatives, including the field ${}^{i}(\partial_\tau^n J_z)_j$ or ${}^{i}_{c}\dot{E}_{zj}$, are then used by STEPPER to select a major time step. The environment of either GEARBOX or TETGEAR is held if one wishes to iterate the mesh evolution (GRIDITER=.TRUE.). Otherwise a single subcycle is done (advancing ${}^{i+1}Y_j$, ${}^{i+1}r_j$ and ${}^{i+1}\dot{r}_j$) and the new FLUID-STATE is output.

The iterative refinement of the mesh evolution is accomplished by first making a predictor step with the jerk continuously updated over the

subcycle. At the end of the first pass the GEARBOX or TETGEAR environment is reset to its content at the beginning of the subcycle. Then the full compliment of material derivatives is explicitly extracted from the (provisional) advanced fluid data, and the jerk coefficient is overwritten by a linear combination of its values at the beginning and end of the step. Further passes over the subcycle use progressively better estimates of the jerk until convergence in the linear combination coefficients is achieved or the maximum allowed number of corrective passes is reached. A final pass with the fully self consistent MESHADVDATA is then executed to obtain the most accurate thermal evolution. The resulting, time-advanced GEARBOX/TETGEAR environment is left in place and the new <FLUID-STATE> is output.

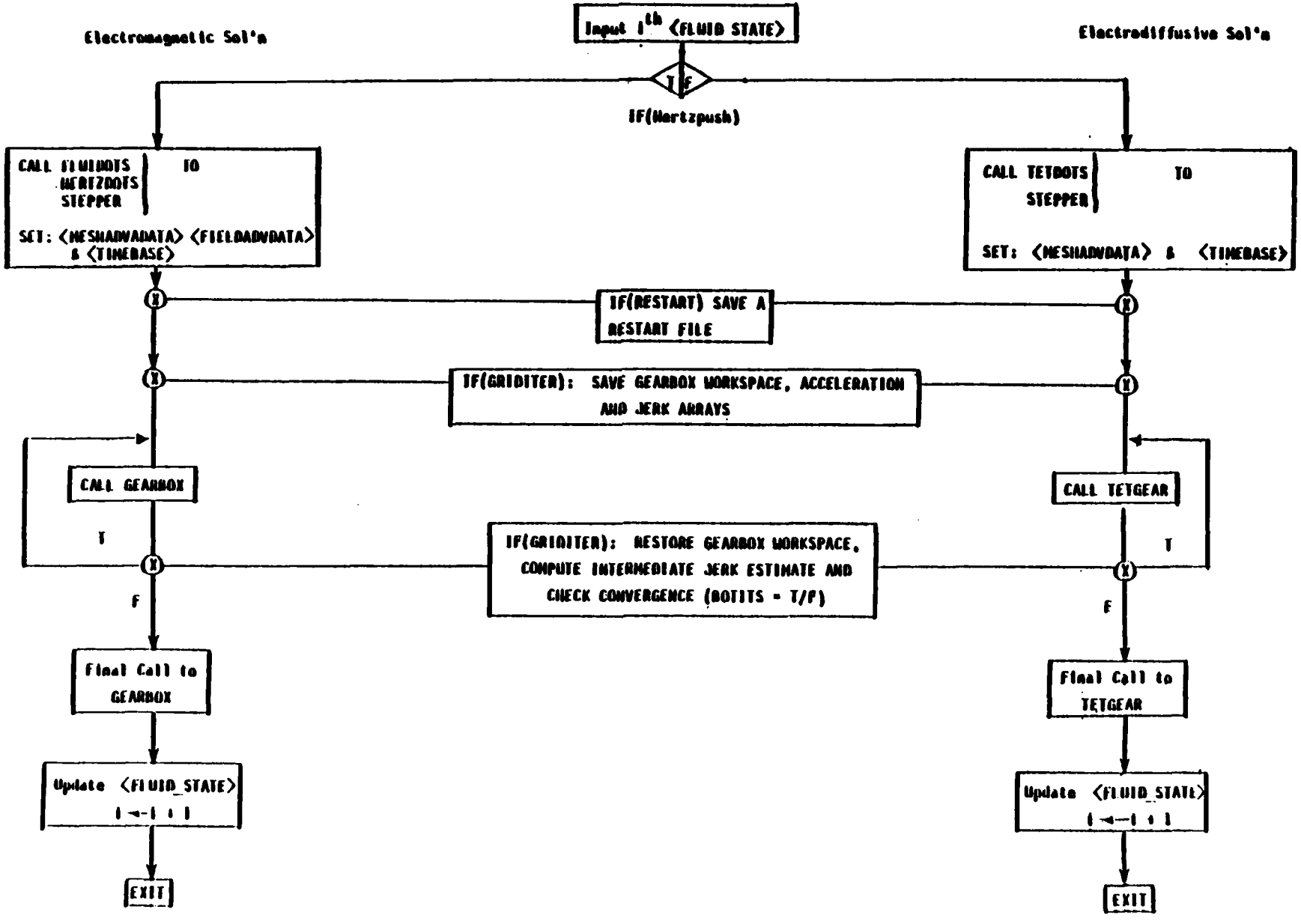
The general structure of HYDROPUSH is illustrated in Fig. 7; the details of the time step algorithms and the iterative refinement scheme are discussed in the subsections below.

(i) Time step considerations

Over a subcycle the steps are controlled by DGEAR, but the assignment of the proper <TIMEBASE> must be based on those rates not accessible to the subcycle integrator, viz. $\{{}^{i}\dot{r}_j, {}^{i}\ddot{r}_j, {}^{i}\dddot{r}_j, {}^{i}_{c}(\partial^{n}_{\tau}J_z)_j$ or ${}^{i}_{c}\dot{E}_z\}$. The comparison among these rates, and those time steps derivable from them, is accomplished by the subroutine STEPPER.

The velocities, accelerations, and jerks of the mesh can be used to formulate several distinct time step estimates. First is the magnetosonic Courant-Fredrich-Lewy (CFL) condition, which seeks to insure that the spatial domain of influence on future values is contained in the domain of dependence established by the finite differencing algorithms which provide the acceleration. The domain of influence is specified by the most

Figure 7. FLOW OF SUBROUTINE HYDROPUSH



rapid signal velocity physically significant in the fluid, which for the present application is the magnetosound speed

$$v_m = \left(\frac{\frac{3}{2}(\gamma T_e + T_I)}{m_I} + \frac{B_\theta^2}{4\pi\rho} \right)^{\frac{1}{2}} .$$

The domain of dependence for the pressure gradient stresses is at least two complete cells, so the distance of propagation is one cell width. Since the cell is generally either compressing or expanding, the disturbance propagation is superposed on the relative boundary motion, and

$$^{i}\Delta\tau_j\big|_{CFL} = \delta_1 ({}^{i}r_{j+1} - {}^{i}r_j)/({}_{c}v_{mj} - {}^{i}\dot{r}_{j+1} + {}^{i}\dot{r}_j)$$

with δ_1 in the range $\frac{1}{3} \rightarrow \frac{1}{2}$ for explicit mesh integration and perhaps as large as $\frac{2}{3} \rightarrow \frac{7}{8}$ for implicit evolution.

A second criterion is the Local Truncation Condition (LTC) based upon the highest time derivative of the forward Taylor series used in advancing the mesh ${}^{i}r_j$. Since area weighed differencing is the basis of the gradient calculations the highest time derivative's contribution to the relative cell area change should be (approximately) the truncation error δ_{GEAR} of the integrator (the GEARBOX variable TOL) multiplied by the typical number of subcycles expected, i.e.

$$^{i}\Delta\tau_j\big|_{LTC} = \left\{ \delta_2 \cdot ({}^{i}a_{j+1} - {}^{i}a_j) / \tfrac{1}{6} | {}^{i}\dddot{a}_{j+1} - {}^{i}\dddot{a}_j | \right\}^{\frac{1}{3}}$$

with $\delta_2 \sim (5\text{-}10) \cdot \delta_{GEAR}$.

A third related criterion is the relative cell area change itself, calculated through second order in the Taylor series - the Quadratic Cell

Area change (QCA). With $\Delta a_j = a_{j+1} - a_j$ and similar notation for the material derivatives, an expanding cell will grow by a fraction δ in

$$ {}^{i}\Delta\tau_j\Big|_{QCA} = \frac{{}^{i}\Delta\dot{a}_j}{{}^{i}\Delta\ddot{a}_j}\left\{\sqrt{1 + \frac{2\,\delta\,{}^{i}\Delta a_j\;{}^{i}\Delta\ddot{a}_j}{({}^{i}\Delta\dot{a}_j)^2}} - 1\right\} $$

where δ may be set in the range $0.08 \rightarrow 0.20$. A similar expression exists for the fractional shrinkage of a contracting cell, and the $\{{}^{i}\Delta\tau_j|_{QCA}\}$ is just the collection of such estimates. A cell destined to change from contraction to expansion (or vice versa) over the time step is excluded from the selection algorithm.

A fourth mesh-based criterion is the time for collapse on axis. If the innermost zones (r_1 and r_2) are projected ahead using the Taylor series coefficients, then one seeks a time which allows the first to intercept the axis, but not the second. After the first has collapsed, only the time for axial interception by the second is relevant. The collapse time ${}^{i}\Delta\tau_c$ is then appropriately set in between these two.

Finally a step $\Delta\tau|_{\mathcal{E}m}$ can be set by insisting that the relative changes in wave sources or diffusion sources, i.e. $\partial_\tau J$, $\dot{E}$ be bounded by a fraction similar to that limiting cell area changes.

The subroutine STEPPER calculates these various time steps $\{\Delta\tau|_{CFL}, \Delta\tau|_{LTC}, \Delta\tau|_{QCA}, \Delta\tau|_{C}, \Delta\tau_{\mathcal{E}m}\}$ and then assigns a final $\Delta\tau$ as

$$ {}^{i}\Delta\tau = \min\left\{\inf_j {}^{i}\Delta\tau_j\Big|_{CFL}, \inf_j {}^{i}\Delta\tau_j\Big|_{LTC}, \inf_j {}^{i}\Delta\tau_j\Big|_{QCA}, {}^{i}\Delta\tau_c, \inf_j {}^{i}\Delta\tau_j\Big|_{\mathcal{E}m}\right\} . $$

This provisional $\Delta\tau$ is then checked for any danger of mesh inversion by advancing the mesh to the end of the interval. If a mesh inversion is implied the step size is cut by a factor of 0.9 and the test is made again. This procedure continues until either the step cuts prevent the inversion or the original step has been cut to a tenth of its original size. In the later event the calculation proceeds with a warning of possible inversion. STEPPER then outputs the new end point (Thi) for the subcycle integrator and an estimate (delTmin) to be used as an explicit step if the time stack internal to the DGEAR integrator is to be reinitialized.

(ii) Explicit and implicit mesh advance with HYDROPUSH

The simplest mesh evolution available is an explicit advance using the Taylor series coefficients at a single time slice,

$$ {}^{*}r_j = {}^{i}r_j + {}^{*}\Delta\tau\, {}^{i}\dot{r}_j + \frac{{}^{*}\Delta\tau^2}{2}\, {}^{i}\ddot{r}_j + \frac{{}^{*}\Delta\tau^3}{6}\, {}^{i}\dddot{r}_j $$

$$ {}^{*}\dot{r}_j = {}^{i}\dot{r}_j + {}^{*}\Delta\tau\, {}^{i}\ddot{r}_j + \frac{{}^{*}\Delta\tau^2}{2}\, {}^{i}\dddot{r}_j \quad . $$

The local truncation condition in STEPPER insures that this Taylor series receives a small contribution from the highest order (and potentially most noisy) term ${}^{i}\dddot{r}_j$. In order to minimize the global accretion of whatever error might be added in any one such time step, however, it is necessary to insist on an evolutionary self-consistency in the advance of the acceleration fields and the jerk fields. A consistency determined at the future <FLUID-STATE> by the position, velocity and temperature fields and external stresses.

One way to help provide self-consistency is to alter $\ddot{r}_j$ during the subcycle by examining the acceleration values obtained ($^*\ddot{r}_j$) and revising by the rule,

$$\overleftarrow{^*\ddot{r}_j} = \frac{^*\ddot{r}_j - {}^i\ddot{r}_j}{^*\Delta\tau}$$

continuously as the subcycle progresses, the superscript $\leftarrow$ denotes a forward time ordering. This is tantamount to choosing a (variable interval) time averaged $\ddot{r}_j$ for the mesh advance, and clearly has a cumulative but distinct impact over the subcycle on the position and velocity fields. For the grid $^*r_j$ this version implies a local <u>effective</u> acceleration

$$^*\overline{\ddot{r}_j} = {}^i\ddot{r}_j + \frac{1}{3}\,(^*\ddot{r}_j - {}^i\ddot{r}_j),$$

while for the velocity, the local effective acceleration is

$$\overline{^*\ddot{r}} = {}^i\ddot{r}_j + \frac{1}{2}\,(^*\ddot{r}_j - {}^i\ddot{r}_j)\ .$$

These effective accelerations appear to be available only for future subcycle times — $^*\ddot{r}_j$ is the acceleration derived from the fluid <u>upon arrival</u> at τ^* by means of the Taylor series coefficients <u>previously</u> computed. The GEARBOX integrator is not always committed to a monotone increasing march in τ^* however, so that any mesh required when it "backs up" should reflect the forwardmost Taylor series information available. For this reason the material derivative codes FLUIDOTS and TETDOTS keep track of the times τ^* input to them and perform the revision of $\ddot{r}_j$ (and hence the calculation of the effective accelerations) if and only if

the given τ^* is the <u>largest</u> time argument seen thus far in the subcycle.

This approach therefore continuously updates the Taylor series on an ever broader temporal data base in order to follow the general trend of $\ddot{r}_j$ instead of just a single value. The explicit (or predictor) phase of HYDROPUSH is just the implementation of this algorithm.

On the other hand, nothing guarantees that the smooth changing of $\ddot{r}_j$ is sufficient to provide true self-consistency in the motion. At the end of a subcycle there exist two versions of ${}^{i+1}\ddot{r}_j$. The first is available from a time slice of advanced fluid variables $\ddot{r}_{jF}({}^{i+1}r_j, {}^{i+1}\dot{r}_j, {}^{i+1}_{c}T_{Ij}, {}^{i+1}_{c}\theta_{ej}, {}^{i+1}E_z)$, the second is that derived over the time step using the jerk coefficient ${}^{i+1}\ddot{r}_j = {}^{i}\ddot{r}_j + {}^{i}_{\Delta\tau}\overleftarrow{\ddot{r}}_j$.

The mean value theorem guarantees however that at least one value of $\ddot{r}_j$ (on $T_{lo} \leq \tau^* \leq T_{hi}$) exists which makes the second version of ${}^{i+1}\ddot{r}_j$ exact to all orders. The essential thrust of the implicit method described here is to find that best value ${}^{\Delta}\ddot{r}_j$ for each cell which provides a solution to the nonlinear difference equation

$$
{}^{i+1}\ddot{r}_{jF}({}^{i+1}r_j, {}^{i+1}\dot{r}_j, {}^{i+1}_{c}T_{Ij}, {}^{i+1}_{c}\theta_{ej}, {}^{i+1}_{c}E_{zj}, {}^{i+1}_{c}B_{\theta j}, {}^{i+1}_{c}E_{rj}) =
$$

$$
{}^{i}\ddot{r}_{jF} + {}^{i}_{\Delta\tau} \cdot {}^{\Delta}\ddot{r}_j
$$

to some specified precision, e.g. the TOL in GEARBOX. Here the acceleration at the new time level depends on the entire time history of the fluid variables computed over the subcycle. The most general method of solution is to iterate across the subcycles. This is accomplished by representing the unknown best value ${}^{\Delta}\ddot{r}_j$ as a weighted sum of the (explicitly calculable) end point values, viz.

$$^{\Delta}\ddot{r}_j = \delta_j \, {}^{i+1}\ddot{r}_{jF} + (1-\delta_j)\, {}^{i}\ddot{r}_{jF}$$

and iteratively computing δ_j by several passes across the subcycle until convergence (in δ_j) is achieved. At any iteration (p) the new δ_j is computed from the implied new fluid state (using the last δ_j vector) and the original fluid state by the required solution of the difference equation, i.e.

$$^{p}\delta_j = \frac{\left[\dfrac{\left({}^{i+1}_{p-1}\ddot{r}_{jF} - {}^{i}\ddot{r}_{jF}\right)}{{}^{i}\Delta\tau} - {}^{i}\ddot{r}_{jF}\right]}{{}^{i+1}_{p-1}\ddot{r}_{jF} - {}^{i}\ddot{r}_{jF}} \quad .$$

The jerk coefficient (embedded in MESHADVDATA) is then updated to be the current estimate of $^{\Delta}\ddot{r}_{jF}$ given above and the entire subcycle is recalculated, completing the loop. Convergence to a particular vector δ_j is thus equivalent to a solution of the difference equation and, in the limit of small time steps, any solution of the difference equation closely approximates a solution of the original partial differential equations. In the implicit scheme of course the revision of $\ddot{r}_j$ <u>during</u> the subcycle is suppressed.

The present implementation of this scheme in HYDROPUSH relies on the simple replacement algorithm discussed above in order to find δ_j. It is probable that convergence can be improved by using more sophisticated estimates based on the iterative sequence $\{_p\delta_j\}$ p=0,1,2,3.., but the best such choice will come from examining the algorithm's performance and thus cannot be selected a priori with any certainty. It is also probable that the inclusion of previous time levels (i-1, i-2...) in an estimate of $^{\Delta}\ddot{r}$ would be a means of filtering the noise that might develop in this variable, and general methods of effecting this extension are under study.

I. The Planned Sequential Benchmark of This Method

a) Checks on internal energy conservation

On a stationary mesh (no compressional heating or expansion cooling) with radiative losses disabled, an initial separation of T_I and T_e (θ_e) will relax inhomogeneously when the (time invariant) ion density varies in space. As a test of the subcycling algorithm,the time asymptotic final state ($T_I = T_e$ (θ_e) $= T_\infty$ for all values of r) must contain the same internal energy as the original. Preliminary tests with uniform ion density (and uniform relaxation) have demonstrated this energy conservation to $O(10^{-3})$.

b) Checks on the conservation of kinetic and internal energy

There exists a class of self-similar Gaussian implosions and explosions for which the time-dependent density scale length (or any other Lagrangian marker radius) varies as

$$\frac{r(t)}{r(o)} = \sqrt{\frac{\gamma_E + (1 - (1 + \gamma_E)\tau)^2}{\gamma_E + 1}}$$

with γ_E = initial thermal energy/initial flow kinetic energy. These flows remain isothermal in space at all times (removing the effects of thermal conduction) and heat/cool θ_e and T_I equally (removing the ion/electron thermal exchange. It is thus a necessary check to insist that the algorithm preserve this flow locally on the mesh and also conserve the sum of flow and internal energy globally in time.

Once these benchmarks are obtained the numerical performance of the code can be properly optimized with respect to mesh density, time step, and convergence criteria.

c) Checks on the conservation of kinetic, internal and magnetic field energy

For a strict MHD model plasma with a fixed (constant) current, the self similar solutions due to Felber[16] provide oscillatory trajectory families for Gaussian implosions. One makes contact with these in the present model by legislating a particular current profile, and by neglecting ohmic heating, ambipolar E_r stresses, ionization dynamics and radiation. The resulting motion can then be compared to the analytical results over several oscillations, and the energy interchange among these three reservoirs monitored with regard to its global conservation in time.

d) Examining the process of collection

A final basic study involves the filling of the central cavity by an initially annular load. In cylindrical geometry this process is quite similar to the diaphragm problem studied by A. Lapidus[17] and a detailed comparison of numerical results will be useful in assessing the shock and rarefaction resolution capabilities of HYDROPUSH. In the limit of large radii the annular diaphragm evolution is very close to the corresponding planar problem for which analytic flow solutions are known.

References

1. R. E. Terry and J. Guillory, Development and Exploration of the Core/ Corona Model of Imploding Plasma Loads, JAYCOR report number TPD-200-80-001-DFR, 1980.
2. D. G. Colombant, M. Lampe and H. W. Bloomberg WHYRAC, A New Modular One-Dimensional Exploding Wire Code, NRL Memo Report 3726, 1978.
3. R. E. Terry and J. Guillory, Development and Exploration of the Core/ Corona Model of Imploding Plasma Loads, JAYCOR report number TPD-200-80-001-FR, p.92.
4. ibid p. 44 and 45, pp 95-98.
5. R. E. Terry and J. Guillory, Annual Report on Modeling of Imploding Annular Loads, JAYCOR report number J207-81-004, 1981, p. 20.
6. ibid, pp 26-58.
7. ibid, Appendix D.
8. R. E. Terry, J. Guillory and D. Duston, Generalized Hertz Vector Potentials and Their Application to Diode Imploded Plasmas, Bull. APS, 26, No. 7, Sept. 1981.
9. A Talmi and G. Gilat, Journal of Computational Physics 23, p. 93, 1977.
10. D. Duston and J. Davis, Phys. Rev. A 21, May 1980.
11. Y. T. Lee (private communication).
12. R. E. Terry and J. Guillory, Development and Exploration of the Core/ Corona Model of Imploding Plasma Loads, JAYCOR report number TPD-200-80-001-FR, 1980, p. 70.
13. S. I. Braginskii in Reviews of Plasma Physics, ed. M. A. Leontovich (Consultants Bureau, NY 1965).

14. J. A. Pruzese, J. Quant. Spectrosc. Radiat. Transfer 25, p. 419, 1981.

15. R. Clark (private communication).

16. F. S. Felber, Self-similar Oscillations of a z-pinch (to appear in Physics of Fluids).

17. A. Lapidus, J. Computational Phys. 8, p. 106, 1971.

APPENDIX I: VAX INTERACTIVE INTERPOLATION PACKAGE

As an illustration of the commands REPRESENT and EVALUATE, the following calculation produced the G-spline coefficients of the branching ratio in Aluminum, cf. Chapter II.

The file of input data resides on {CHEMNODE}, I.E.

```
FILES CHEMNODE

Directory _DRA1:[IPR1.HYDRO.CHEM]

CRE15.DAT;1              7/11    12-AUG-1981 18:42 (RE,RWE,,)
CRE17.DAT;1              7/11     7-AUG-1981 09:30 (RE,RWE,,)
CRE19.DAT;1              7/11    19-AUG-1981 13:12 (RE,RWE,,)
CRE19BR.DAT;1            8/11    20-AUG-1981 16:41 (RE,RWE,,)
CRE19LTEB.DAT;1          7/11    26-MAR-1982 16:48 (RE,RWE,,)
CRE19LZTB.DAT;1          7/11    20-AUG-1981 16:37 (RE,RWE,,)
CRE19ZE.DAT;1            7/11    20-AUG-1981 14:17 (RE,RWE,,)
CRE21.DAT;1              7/11     7-AUG-1981 09:32 (RE,RWE,,)
CRECLAM.DAT;1            4/11    23-MAR-1982 20:48 (RE,RWE,,)
CREKLAM.DAT;1            3/11    19-MAR-1982 10:32 (RE,RWE,,)
CRELLAM.DAT;1            4/11    23-MAR-1982 13:39 (RE,RWE,,)
CRERLAM.DAT;1            4/11    26-MAR-1982 13:26 (RE,RWE,,)
EMITALC.DAT;1            3/11    23-MAR-1982 20:46 (RE,RWE,,)
EMITALK.DAT;1            3/11    19-MAR-1982 10:21 (RE,RWE,,)
EMITALL.DAT;1            3/11    23-MAR-1982 13:29 (RE,RWE,,)
EMITFILE.EXE;1          19/22    17-MAR-1982 14:19 (RE,RWE,,)
EOSFILE.EXE;1            8/11    19-AUG-1981 11:19 (RE,RWE,,)
EOSFILEXP.EXE;1         10/11    19-AUG-1981 11:20 (RE,RWE,,)
RCRSECT.DAT;1            3/11    17-MAR-1982 14:20 (RE,RWE,,)

Total of 19 files, 121/220 blocks.
```

and is named CRE19LZTB.DAT;1. The command REPRESENT IS

```
SHOW SYMBOL REPRESENT
REPRESENT = @COMMANDS:REPIT.COM

TYPE COMMANDS:REPIT.COM


<<<<     REPRESENT FILENAME      >>>>
                        P1
ASSIGN 'P1'.DAT INFILE
INQUIRE P2 'WHAT NAME DO YOU WISH FOR THE OUTPUT FILE? ENTER IT,PLEASE.'
ASSIGN 'P2'.DAT OUTFILE
ASSIGN 'P2'H.DAT HEADER
ASSIGN/USER_MODE SYS$COMMAND: SYS$INPUT:
RUN UTILITY:REPIT.EXE
!<<<< PROMPTS WILL BE GENERATED FOR FURTHER INFORMATION.  >>>>
DEASSIGN INFILE
DEASSIGN OUTFILE
DEASSIGN HEADER
FILES %.DAT
```

and its use is indicated by the above command file.

```
  REPRESENT CHEMNODE:CRE19LZTB
WHAT NAME DO YOU WISH FOR THE OUTPUT FILE? ENTER IT,PLEASE.: LAMZTB19
     WHAT IS YOUR SPLINE CHOICE?
   ENTER 'GSPLINES.','ESPLINES.',OR 'EI2SPLINES.'
'GSPLINES: '
     DO YOU WISH IMSL ACCURACY CHECKS?
'YES'
     ENTER THE DIGITS OF PRECISION FOR YOUR INPUT FILE.
5
     DO YOU WISH IMSL ITERATIVE REFINEMENT?
'NO '
 ARE THERE BOUNDARY DERIVATIVES IN THE INPUT FILE?
'NO'
     ENTER THE NUMBER OF X-VALUES IN YOUR INPUTFILE.
31
     ENTER THE NUMBER OF FUNCTIONS ( F(X)) TO BE FIT.
3
     THE ASSUMED FORMAT OF YOUR INPUT FILE IS:                (1X,1P4D25.15)
 ARRANGED AS X(I) F1(X(I)) F2(X(I))...IN EACH RECORD,
 WITH ONE RECORD FOR EACH X-VALUE AND B.C.
 DO YOU WISH TO CHOOSE THE WIDTH PARAMETER?
'YES'
     ENTER A VARIABLE MESH SCALE FACTOR (0.2D0, 5.0D0)
0.4425D0
 THE CHOSEN AND DEFAULTED CONTROL PARAMETERS ARE:
     BASE POINTS,+B.C . EPS_CONV , IDGT , ITMAX
         31          31   1.0000000000000000E-02          5           0
     B.C. DESCRIPTORS
         0           0           0           0
     BEGIN SMOOTH INTERPOLATION WITH GSPLINES:
    MILD IMSL ERRORS, DO YOU WANT THE DETAILS? (`YES` or `NO `)
'NO'

 PROCEEDING TO INTERPOLATION ANYWAY WITH IDGT=    5 AND IER=   34
 THE FORMAT OF YOUR OUTPUT FILE IS:                       (1X,1P5D25.15)
 ARRANGED AS X(I) , D(I) , L1(I) , L2(I)...  IN EACH RECORD
 WITH A HEADER FILE: D1 TITLE BDM DIM...ADDED
FILES CLOSED.

Directory _DRA1:[IPR1.NETNODE]

LAMZTB19.DAT;1          8/11    26-APR-1982 14:42 (RWED,RWED,RWE,R)
LAMZTB19H.DAT;1         1/11    26-APR-1982 14:42 (RWED,RWED,RWE,R)

Total of 2 files, 9/22 blocks.
      $;

        The two output files contain the base points, width parameters, and
        fit coefficients --

  type LAMZTB19.DAT;1
```

```
-3.565338761448581D+00   5.787921333314708D-02   7.686466045104355D-01   2.006698621063578D+00  -2.264636322246773D+00
-3.063203806969015D+00   5.787921333314708D-02   7.643502660357868D-01   2.349367141389346D+00   6.090413451934824D+01
-2.636144269074969D+00   5.787921333314708D-02   5.809530171818802D-01   1.306556374545496D+00  -5.456160516853623D+01
-2.411728298490525D+00   2.985753655867177D-02   8.537692377155075D-01   3.268382550492575D+00   1.968543982839974D+02
-2.287750073191981D+00   1.322979227397374D-02   1.018506598350696D-02  -4.800632379561615D-01  -1.031972843890942D+02
-2.145596772179739D+00   8.882736194546402D-03   3.226152226101544D-01   1.022673322835043D+00   3.183813256928196D+01
-2.025334097407630D+00   9.231741985704654D-03   2.671590540391557D-01   5.766126730902237D-01  -2.433064475962044D+01
-1.911229395174805D+00   9.766951688861868D-03   3.457589430854888D-01   9.915588370243677D-01   1.836933708472814D+01
-1.777513509488090D+00   1.047922253637821D-02   3.830841400484622D-01   1.000664669658616D+00  -3.326200241529888D+00
-1.620778112473290D+00   1.079311367311028D-02   4.090639809242934D-01   1.143257704665164D+00   1.133720702365725D+01
-1.481715532683225D+00   1.043980526753098D-02   3.076072430546788D-01   8.136798483178342D-01  -8.310246285855078D-01
-1.359529217820313D+00   9.595844238090964D-03   2.148609310745069D-01   6.328446000051369D-01   4.932138762323801D+00
-1.234918310150672D+00   9.864193917160699D-03   2.083292316920725D-01   6.147711236305317D-01   5.089001726894737D-01
-1.100576920577420D+00   1.133271281154021D-02   1.877785160841748D-01   6.120908398526308D-01   2.968301530969380D+00
-9.542907304550876D-01   1.349509619591027D-02   1.587029149764459D-01   5.590378960877922D-01   8.635997819158854D-01
-7.942061726822921D-01   1.616139200208668D-02   1.331636219519763D-01   5.356413259540969D-01   2.219044600258176D+00
-6.180140911974972D-01   1.907524092328200D-02   1.123757834033265D-01   5.086936554928983D-01   9.093491484739751D-01
-4.254753315173112D-01   2.196692030618370D-02   1.024329546225339D-01   5.035915960769449D-01   1.854473133597289D+00
-2.208504393867342D-01   2.438117567965083D-02   9.097820691455150D-02   4.924987702845158D-01   1.041806999745014D+00
-7.134403629543253D-03   2.569425987047498D-02   8.586142768162164D-02   4.834870756284850D-01   1.595837896391857D+00
 2.111816635080864D-01   2.587492059936546D-02   8.329916776996381D-02   4.883360671718932D-01   1.188883644530502D+00
 4.257563992688952D-01   2.530574655874976D-02   7.583817575667472D-02   4.474498190113425D-01   1.358115973748145D+00
 6.333686273111865D-01   2.451942071365663D-02   6.386138825479059D-02   4.557182759438769D-01   1.182899265267289D+00
 8.376372353645590D-01   2.412868288835708D-02   5.232917608929117D-02   3.889233538909116D-01   1.115420219813865D+00
 1.042724929267001D+00   2.434622141499233D-02   4.563795492591683D-02   4.690082118524350D-01   1.253852244646192D+00
 1.250647381518816D+00   2.495601072813559D-02   3.450929585228046D-02   3.271426402138240D-01   9.132611944682018D-01
 1.461969777311134D+00   2.569710458206345D-02   3.283984350484740D-02   5.426641213204757D-01   1.466113526070920D+00
 1.676551018612627D+00   2.642430949328068D-02   1.869996055765073D-02   1.822039689446685D-01   5.053223573505368D-01
 1.894003780339242D+00   2.707532699593058D-02   2.702001239422811D-02   7.293947220282713D-01   1.979270591722756D+00
 2.113887412616442D+00   2.707532699593058D-02   4.405841369402964D-03  -6.809513308372457D-02  -1.818997090695698D-01
 2.335778957844778D+00   2.707532699593058D-02   2.561421399560279D-02   9.533402199166938D-01   2.591387347019640D+00
```

for the three functions given in the input file, as well as a header file --

```
type LANZTB19H.DAT;1

 5.787921333314708D-02BSPLINES:    0000    31 5==========================================================================
```

used by the companion command EVALUATE to decode the first data file in order to calculate the interpolant on any desired domain.

The command EVALUATE IS

```
SHOW SYMBOL EVALUATE
EVALUATE = @COMMANDS:CHEKIT.COM
```

```
 TYPE COMMANDS:CHEKIT.COM

 <<<<      EVALUATE FILENAME   >>>>
                        P1
ASSIGN 'P1'.DAT INFILE
ASSIGN 'P1'H.DAT HEADER
INQUIRE P2 'WHAT NAME DO YOU WISH FOR THE OUTPUT FILE? ENTER IT,PLEASE.'
ASSIGN 'P2'.DAT OUTFILE
ASSIGN/USER_MODE SYS$COMMAND: SYS$INPUT:
RUN UTILITY:CHEKIT.EXE
!<<<< PROMPTS WILL BE GENERATED FOR FURTHER INFORMATION.  >>>>
DEASSIGN INFILE
DEASSIGN OUTFILE
DEASSIGN HEADER
FILES *.DAT
```

and its use is again indicated in the command file.

```
  EVALUATE LAMZTB19
WHAT NAME DO YOU WISH FOR THE OUTPUT FILE? ENTER IT,PLEASE.: BRANCH.
     THE INTERPOLATION FILE FORMAT APPEARS TO BE
(1X,1PSD25.15)
     THE DEFAULT SPLINE CHOICE IS
GSPLINES:
     ENTER XMIN,XMAX,NUMBER OF INTERIOR CHECK POINTS:
-3.565338761448581D0 2.335778957844778D0 49
     DO YOU WISH TO CHANGE THE SPLINE CHOICE?
'NO'
     ENTER THE INDEX OF THE INTERPOLANT, NUMBERS OF DERIVATIVES (0-4):
1 2
     ACTIVE CONTROL & DOMAIN PARAMETERS:
  XMIN XMAX JORD OUTPUT FORMAT
  -3.565338761448581        2.335778957844778                1 (1X,1P4E18.6)
     DO YOU WISH TO CALCULATE OTHER INTERPOLANTS IN THIS FILE?
'NO':
FILES CLOSED.

Directory _DRA1:[IPR1.NETNODE]

BRANCH.DAT;1              8/11    26-APR-1982 15:07 (RWED,RWED,RWE,R)
LAMZTB19.DAT;1            8/11    26-APR-1982 15:07 (RWED,RWED,RWE,R)
LAMZTB19H.DAT;1           1/11    26-APR-1982 15:07 (RWED,RWED,RWE,R)

Total of 3 files, 17/33 blocks.
```

Here the output file contains the evaluation grid, the interpolant, and its first two derivatives which can be used to check the quality of the fit against the original data.

```
  type BRANCH.DAT;1
```

-3.565339E+00	1.039832E+00	1.227985E+00	-3.239513E+00
-3.447316E+00	1.161874E+00	8.422902E-01	-3.140310E+00
-3.329294E+00	1.240859E+00	5.099783E-01	-2.448775E+00
-3.211272E+00	1.285800E+00	2.670826E-01	-1.657441E+00
-3.093249E+00	1.307987E+00	1.302959E-01	-5.579908E-01
-2.975227E+00	1.323375E+00	1.674333E-01	1.307838E+00
-2.857205E+00	1.356955E+00	4.363951E-01	3.021442E+00
-2.739182E+00	1.429004E+00	7.586584E-01	1.667650E+00
-2.621160E+00	1.518562E+00	6.416368E-01	-4.197363E+00
-2.503138E+00	1.551030E+00	-1.798344E-01	-8.467281E+00
-2.385115E+00	1.479900E+00	-8.899082E-01	-1.815282E+00
-2.267093E+00	1.381906E+00	-6.668783E-01	3.148222E+00
-2.149071E+00	1.310677E+00	-6.836972E-01	-3.541784E+00
-2.031048E+00	1.204371E+00	-1.062440E+00	-1.043433E+00
-1.913026E+00	1.083757E+00	-9.075226E-01	2.678957E+00
-1.795003E+00	9.956105E-01	-5.930998E-01	2.358012E+00
-1.676981E+00	9.377671E-01	-4.370949E-01	-1.565059E-01
-1.558959E+00	8.774831E-01	-6.351986E-01	-2.630297E+00
-1.440936E+00	7.860593E-01	-8.806479E-01	-1.031771E+00
-1.322914E+00	6.802963E-01	-8.742708E-01	8.375162E-01
-1.204892E+00	5.840681E-01	-7.526054E-01	1.065729E+00
-1.086869E+00	5.024869E-01	-6.311619E-01	9.972327E-01
-9.688470E-01	4.347243E-01	-5.192443E-01	8.939088E-01
-8.508246E-01	3.794732E-01	-4.184092E-01	8.224108E-01
-7.328023E-01	3.356715E-01	-3.252771E-01	7.490377E-01
-6.147799E-01	3.022271E-01	-2.441380E-01	6.136411E-01
-4.967575E-01	2.772742E-01	-1.822774E-01	4.341325E-01
-3.787352E-01	2.584212E-01	-1.399093E-01	2.974040E-01
-2.607128E-01	2.438119E-01	-1.087352E-01	2.413996E-01
-1.426905E-01	2.325889E-01	-8.211865E-02	2.062576E-01
-2.466813E-02	2.241874E-01	-6.174208E-02	1.306780E-01
9.335422E-02	2.175600E-01	-5.281279E-02	1.707617E-02
2.113766E-01	2.111704E-01	-5.771640E-02	-9.672065E-02
3.293989E-01	2.034791E-01	-7.411116E-02	-1.720560E-01
4.474213E-01	1.934654E-01	-9.582754E-02	-1.846407E-01
5.654436E-01	1.809539E-01	-1.152028E-01	-1.349504E-01
6.834660E-01	1.666010E-01	-1.263985E-01	-5.330437E-02
8.014884E-01	1.514884E-01	-1.283493E-01	1.498907E-02
9.195107E-01	1.365440E-01	-1.241977E-01	5.096940E-02
1.037533E+00	1.222962E-01	-1.167856E-01	7.449613E-02
1.155555E+00	1.090829E-01	-1.067505E-01	9.304567E-02
1.273578E+00	9.713288E-02	-9.590212E-02	8.582338E-02
1.391600E+00	8.636455E-02	-8.692821E-02	6.892371E-02
1.509622E+00	7.659666E-02	-7.833485E-02	8.157033E-02
1.627645E+00	6.795973E-02	-6.787143E-02	8.826947E-02
1.745667E+00	6.048220E-02	-5.975888E-02	4.331164E-02
1.863690E+00	5.363881E-02	-5.660488E-02	2.469528E-02
1.981712E+00	4.725497E-02	-5.020054E-02	9.163668E-02
2.099734E+00	4.207892E-02	-3.729026E-02	1.003034E-01
2.217757E+00	3.807632E-02	-3.384146E-02	-6.394240E-02
2.335779E+00	3.319387E-02	-5.198019E-02	-2.161074E-01

****

The original data for this check is the second column of
CRE19LZTB I.E. the natural logarithm of the branching ratio.

```
.wpe CHEMNODE:CRE19LZTB.DAT
```

-3.565338761448581D+00	1.039831424539511D+00	2.828740117988182D+00	1.692412498640776D+01
-3.063203806969015D+00	1.311707869719672D+00	3.712508782266432D+00	4.095642850659761D+01
-2.636144269074969D+00	1.508508898314314D+00	4.519986908226843D+00	9.183431304429138D+01
-2.411728298490525D+00	1.502666359599353D+00	4.493654810405845D+00	8.944776381830432D+01
-2.287750073191981D+00	1.396386075598492D+00	4.040571230227099D+00	5.685881300327130D+01
-2.145596772179739D+00	1.308280867311329D+00	3.699807781339405D+00	4.043953038057452D+01
-2.025334097407630D+00	1.198285032784034D+00	3.314427910689791D+00	2.750665322016935D+01
-1.911229395174805D+00	1.082131225717454D+00	2.950962020156286D+00	1.912434288477465D+01
-1.777513509488090D+00	9.855886021047646D-01	2.679388514090636D+00	1.457617744308059D+01
-1.620778112473290D+00	9.120654898201603D-01	2.489459179306264D+00	1.205475489629135D+01
-1.481715532683225D+00	8.208695603108210D-01	2.272475032728364D+00	9.703387319323800D+00
-1.359529217820313D+00	7.127973658743286D-01	2.039689042059260D+00	7.688218114662057D+00
-1.234918310150672D+00	6.071497642445650D-01	1.835193203994632D+00	6.266344711318834D+00
-1.100576920577420D+00	5.112326445184116D-01	1.667345172661011D+00	5.298083612695162D+00
-9.542907304550876D-01	4.272603253413399D-01	1.533051701989231D+00	4.632291647861490D+00
-7.942061726822921D-01	3.570863738147308D-01	1.429159306464112D+00	4.175187662790916D+00
-6.180140911974972D-01	3.030199460001211D-01	1.353941469889685D+00	3.872659459592118D+00
-4.254753315173112D-01	2.653001963809026D-01	1.303822319777141D+00	3.683348731472336D+00
-2.208504393867342D-01	2.396665792120749D-01	1.270825360082219D+00	3.563792761113591D+00
-7.134403629543253D-03	2.231241056698613D-01	1.249975693180896D+00	3.490258119358040D+00
2.111816635080864D-01	2.111816635080864D-01	1.235136713950289D+00	3.438848627149600D+00
4.257563992688952D-01	1.954978899694906D-01	1.215916228299126D+00	3.373383438189475D+00
6.333686273111865D-01	1.728516087123774D-01	1.188689700800641D+00	3.282776968045358D+00
8.376372353645590D-01	1.468617074663453D-01	1.158193782401587D+00	3.184176762817926D+00
1.042724929267001D+00	1.216908920693823D-01	1.129404939794960D+00	3.093814946105111D+00
1.250647381518816D+00	9.935483502179337D-02	1.104458130461900D+00	3.017588886214447D+00
1.461969777311134D+00	8.041872151470686D-02	1.083740758255173D+00	2.955715513970581D+00
1.676551018612627D+00	6.474145351679537D-02	1.066883149836152D+00	2.906306845302302D+00
1.894003780339242D+00	5.193570594400500D-02	1.053308018891528D+00	2.867119934824716D+00
2.113887412616442D+00	4.156082892180062D-02	1.042436570215534D+00	2.836119005440282D+00
2.335778957844778D+00	3.319386485073257D-02	1.033750927785840D+00	2.811592159769375D+00

The general features of these two interpolation commands which have not been illustrated explicitly revolve around the additional interpolants available:

ESPLINES (the error function)
EI2SPLINES(its second integral)

and the error reporting paths available in REPRESENT when the spline matrix is ill-conditioned or when interative refinement is invoked.

APPENDIX II. Derivation of the Electrodiffusive Limit

The general solution of the Hertz vector wave equation is a superposition of incoming and outgoing cylindrical waves. The linear or nonlinear response of the medium, specified by the convolution involving $\partial_\tau J$, acts to cut off these freely propagating components. One may view this process as the production of reflected power of nearly opposite intensity to that incident, leaving a residual amount deposited in the load as Joule heating. Because of this near cancelation of Z field components it is possible to neglect the second order time derivative in the Z wave equation in a first approximation. In particular the requirement of a strict detailed balance in the free wave component $E_z^W(\chi,\tau)$ demands that $\nabla^2 E_z^W = 0$, because one is interested in the limit for which $\partial_\tau^2 E_z^W \ll \partial_\tau J$ is true to arbitrary precision. Physically the common situation is one of nearly balanced waves in which $\varepsilon \equiv \sup_{\{\chi\}} \frac{\partial_\tau^2 E_z^W}{\partial_\tau J} \ll 1$, but the free wave components are solutions of $\Box^2 E_z^W = 0$. The constraint that $\nabla^2 E_z^W = 0$ is simply a way to select the limit $\varepsilon \to 0$. Since any solution of Laplaces equation valid inside some radius, and vanishing on that radius, must vanish everywhere within, then in the limit of truly detailed balance inside a radius χ_0 and on that radius, $\chi^{-1}\partial_\chi(\chi\partial_\chi E_z^W) = 0$ is a necessary condition. But using Faraday's law one has $\chi^{-1}\partial_\chi(\chi\partial_\tau B_\theta^W) = 0$ as well. In terms of the Hertz vector Z, $B_\theta^W = -\partial_\tau\partial_\chi Z$, so that the original condition of detailed balance in the wave components becomes $-\chi^{-1}\partial_\chi(\chi\partial_\chi\partial_\tau^2 Z) = 0$ for some region $\chi < \chi_0$. The equation governing the electodynamics in the limit of strict detailed balance is thus obtained by operating on the original wave equation with ∇^2 and removing the contribution arising from $\partial_\tau^2 Z$. The application of ∇^2 on the convolution with $\partial_\tau J$ produces just the integrand

$$4\pi\chi^{-1}\partial_\chi\left\{\chi\,\partial_\chi\left[\left(\int_0^\chi d\tilde{\chi}\tilde{\chi}\partial_\tau J\right)\ln\frac{\chi}{\chi_N}+\int_\chi^\infty d\tilde{\chi}\tilde{\chi}\ln\frac{\tilde{\chi}}{\chi_N}\partial_\tau J\right]\right\}$$

$$= 4\pi\partial_\tau\, J(\chi,\tau).$$

The resulting equation is

$$\chi^{-1}\partial_\chi\,\chi\,\partial_\chi E = 4\pi\partial_\tau J\ , \tag{A.II.1}$$

because $-\nabla^2 Z = E$ in the original relationship. It is this relationship which can be transformed into a diffusion equation for $E_z(\chi,\tau)$.

Using the usual Ohm's law, one finds

$$\partial_\tau J = \hat{E}\,\partial_\tau\Sigma + \Sigma\,\partial_\tau\,(E_z + \beta_r B_\theta + E_{th})$$

$$= \hat{E}\,\partial_\tau\Sigma + \Sigma(\partial_\tau E_z + \beta_r\partial_\tau B_\theta + B_\theta\partial_\tau\beta_r + \partial_\tau E_{th})$$

and applying Faraday's law once again, to eliminate $\partial_\tau B_\theta$ and form the (dimensionless) material derivative,

$$\partial_\tau J = \hat{E}\,\partial_\tau\,\Sigma + \Sigma\,\left(\frac{D}{D\tau}E_z + B_\theta\partial_\tau\beta_r + \partial_\tau E_{th}\right). \tag{A.II.2}$$

Here $\frac{D}{D\tau} = \partial_\tau + \beta_r\,\partial_\chi$; $\hat{E} \equiv E_z + \beta_r B_\theta + E_{th}$, the total electric field in the plasma load; and $\Sigma = \sigma\,\frac{r_o}{c}$ is the dimensionless conductivity. Finally, in this case of a linear response relating J and E, the equation determining the diffusion of E_z into the plasma becomes

$$\frac{D}{D\tau} E_z = \frac{1}{4\pi\Sigma} \chi^{-1}\partial_\chi(\chi\, \partial_\chi E) - \hat{E}\partial_\tau \ln\Sigma - B_\theta\, \partial_\tau\beta_r - \partial_\tau E_{th}. \qquad \text{(A.II.3)}$$

In the nonlinear case, reflecting a marginally stable drift speed limit, this derivation produces the same result, insofar as $J = \Sigma(E)\, E$ is formally admissible as a constitutive relation. However, one is forced either to neglect, or to model from first principles, the term $\partial_\tau \ln\Sigma$ in the relationship just given. The reason stems from the implicit definition of $\Sigma(E) = \frac{en_e c_s}{E \cdot J_{scale}}$. If an explicit time derivative is to be calculated for Σ, then it must arise through some model of the turbulent fields providing the enhanced drag on one's proverbial test particle. This model would necessarily involve various functionals of E and its time history and would generally not hold Σ to the exact, marginally stable value at all times. Such a model is beyond the present scope of this work.

It is a better choice to assume that $\Sigma(E)$ has no well-defined time derivative, that it respresents a local, time averaged estimate of the truly physical transport coefficient. A strict application of the marginal stability constraint decouples J and E once E becomes sufficiently large. When this decoupling is imposed on a diffusion equation there is no mechanism left to evolve J and E, cf. Eqn(A.II.1). The more accurate first principles model would not force a complete decoupling, but would provide a smooth transition between linear response and drift speed limitation -- the appropriate asymptotic conditions. In contrast, for the complete electromagnetic description, a drift speed limit is fully admissible -- it simply specifies a wave source term $\partial_\tau J = e\partial_\tau(n_e c_s)$. The use of the full wave equation is preferable whenever strict adherence to the drift speed limitation is desired, and the wave components generated in the drift speed limited

regions of the load are probably important contributions to the dynamics of the transition front between the two asymptotic domains of resistivity. On the other hand, if the diffusive approximation is adopted, then the drift speed limitation must be applied with the caveat that its time dependence $\partial_\tau \ln\Sigma(E)$ cannot be specified from the local drift speed constraint alone and is better neglected than modeled in any ad hoc fasion.

APPENDIX III. Area Weighted Differences

In the Lagrangian mesh shown in Figure 4 the only fundamental positions are the radial cell boundary locations $^*r_j$, and their associated velocities $^*\dot{r}_j$. The scalar quantities assigned to the cell centers have implied spatial locations, $^*_c r_j$, derived from the boundary locations. In order to avoid formal singularities in various derivative expressions the independent variable is chosen here as $a_j = (^*r_j)^2$, an areal coordinate. The two primary types of differential operators to be estimated are (i) derivatives of first and second order at a boundary based on cell-cenetered or cell boundary data, and (ii) derivatives at the cell centers based on cell-centered data.

Turning first to those operators based on or producing cell boundary data, the radial gradient for any function $_cF_j$ defined on $^*_c r_j$ is estimated by

$$(\partial_r F)_j \approx 2\ ^*r_j \left(\frac{_cF_j - {}_cF_{j-1}}{_ca_j - {}_ca_{j-1}} \right) \quad , \qquad \text{(A.III.1)}$$

with the second derivative estimate, based on F_j, given by

$$(\partial_r^2 F)_j \approx 2\ ^*r_j \left[\frac{2\,^*_c r_j \left(\frac{F_{j+1} - F_j}{a_{j+1} - a_j} \right) - 2\,^*_c r_{j-1} \left(\frac{F_j - F_{j-1}}{a_j - a_{j-1}} \right)}{_ca_j - {}_ca_{j-1}} \right] \quad , \qquad \text{(A.III.2)}$$

and the averaging operator to the boundary defined as

$$\overline{F}_j = \left(\frac{_ca_j - a_j}{_ca_j - {}_ca_{j-1}} \right) {}_cF_{j-1} + \left(\frac{a_j - {}_ca_{j-1}}{_ca_j - {}_ca_{j-1}} \right) {}_cF_j \quad . \qquad \text{(A.III.3)}$$

These difference operators are then sufficient to cover all gradients and averages needed in computing the evolution of the fundamental boundary variables $[V_r(t)]_j = {}^*\dot{r}_j$ and its integral, the Lagrangian position.

The corresponding operators estimating first and second order gradients at the cell centers can be obtained similarly. Here the use of ghost points located at ${}_c r_o$ and ${}_c r_{j+1}$ is the natural means of enforcing the boundary conditions appropriate to the diffusive processes, such as thermal condition. The difference operators given below assume that some cell-centered quantity ${}_c F_j$ has been assigned ghost point values ${}_c F_{J+1} = {}_c F_j$, ${}_c F_o = {}_c F_1$ in order to make the flux estimate, proportional to $\partial_r F$, vanish at the first and last boundary. First one may define several (nonuniform) mesh intervals,

$$
\begin{aligned}
h_j^< &= {}_c a_j - {}_c a_{j-1} \\
h_j^> &= {}_c a_{j+1} - {}_c a_j \\
h_j &= {}_c a_{j+1} - {}_c a_{j-1} \\
h_j^\Delta &= \left(\frac{h_j^>}{h_j^<} - \frac{h_j^<}{h_j^>} \right)
\end{aligned}
\tag{A.III.4}
$$

and the ghost point locations

$$
\begin{aligned}
h_o^> &= 2({}_c a_1 - a_1) \\
h_J^> &= 2(a_{J+1} - {}_c a_J) \quad .
\end{aligned}
$$

The first derivative $\partial_a F$ is then estimated by

$$
{}_c(\partial_a F)_1 \approx \frac{\left[\frac{h_j^<}{h_j^>} \; {}_c F_{j+1} + h_j^\Delta \, {}_c F_j - \frac{h_j^>}{h_j^<} \; {}_c F_{j-1} \right]}{h_j} \tag{A.III.5}
$$

with boundary conditions,

$$c(\partial_a F)_1 \approx \frac{h_o^>}{h_o^> + h_1^>}\left(\frac{c F_2 - c F_1}{h_1^>}\right), \quad c(\partial_a F)_N \approx \frac{h_J^>}{h_J^> + h_{J-1}^>} \cdot \left(\frac{c F_J - c F_{J-1}}{h_{J-1}^>}\right) .$$

The second derivatives are similarly obtained using

$$c(\partial_a^2 F)_j \approx \frac{2}{h_j^< \, h_j^>}\left[\frac{h_j^<}{h_j} \, c F_{j+1} - c F_j + \frac{h_j^>}{h_j} \, c F_{j-1}\right] , \qquad \text{(A.III.6)}$$

while the boundary conditions become

$$c(\partial_a^2 F)_1 \approx \frac{2}{h_1^>}\left(\frac{c F_2 - c F_1}{h_o^> + h_1^>}\right) , \quad c(\partial_a^2 F)_J \approx - \frac{2}{h_{J-1}^>}\left(\frac{c F_J - c F_{J-1}}{h_{J-1}^> + h_J^>}\right)$$

All the expressions reduce to well known second order accurate differencing schemes when the mesh is presumed uniform.

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

Assistant to the Secretary of Defense
Atomic Energy
ATTN: Executive Assistant
ATTN: Military Applications

Defense Intelligence Agency
ATTN: DT-1B, R. Rubenstein

Defense Nuclear Agency
ATTN: RAAE
ATTN: RAEV
ATTN: STNA
ATTN: RAEE
4 cy ATTN: TITL

Defense Technical Information Center
12 cy ATTN: DD

Field Command
Defense Nuclear Agency, Det 1
Lawrence Livermore Lab
ATTN: FC-1

Field Command
Defense Nuclear Agency
ATTN: FCPR
ATTN: FCTT, W. Summa
ATTN: FCTT
ATTN: FCTXE
ATTN: FCTT, G. Ganong

Under Secy of Def for Rsch & Engrg
ATTN: Strategic & Space Sys, OS

DEPARTMENT OF THE ARMY

Harry Diamond Laboratories
ATTN: DELHD-NW-P
ATTN: DELHD-NW-RA
ATTN: DELHD-NW-RI, Kervis
ATTN: DELHD-TA-L

US Army Nuclear & Chemical Agency
ATTN: Library

US Army Test and Evaluation Comd
ATTN: DRSTE-EL

USA Missile Command
ATTN: Documents Section

DEPARTMENT OF THE NAVY

Naval Research Laboratory
ATTN: Code 2000, J. Brown
ATTN: Code 4770, I. Vitokovitsky
ATTN: Code 4720, J. Davis
ATTN: Code 4780, S. Ossakow
ATTN: Code 4773, G. Cooperstein

Naval Weapons Center
ATTN: Code 343, FKA6A2, Tech Svcs

DEPARTMENT OF THE NAVY (Continued)

Naval Surface Weapons Center
ATTN: Code R40
ATTN: Code F31
ATTN: Code F34

DEPARTMENT OF THE AIR FORCE

Air Force Weapons Laboratory
ATTN: SUL
ATTN: CA
ATTN: NTYC
ATTN: NT
ATTN: NTYP

Ballistic Missile Office
ATTN: ENSN
ATTN: SYDT

Deputy Chief of Staff
Research, Development, & Acq
ATTN: AFRDQI

Space Division
ATTN: XR, Plans
ATTN: YEZ
ATTN: YGJ
ATTN: YKF
ATTN: YKS, P. Stadler
ATTN: YKM
ATTN: YNV

Strategic Air Command
ATTN: INT, E. Jacobsen

DEPARTMENT OF ENERGY

Office of Military Application
Attention Ofc of Inert Fusion for
ATTN: S. Kahalas
ATTN: T. Godlove
ATTN: C. Hilland

OTHER GOVERNMENT AGENCY

Central Intelligence Agency
ATTN: OSWR/NED

DEPARTMENT OF ENERGY CONTRACTORS

University of California
Lawrence Livermore National Lab
ATTN: Technical Info Dept Library
ATTN: L-545, J. Nuckolls, Class L-33
ATTN: L-47, L. Wouters
ATTN: L-153
ATTN: L-153, D. Meeker, Class L-477
ATTN: W. Pickles, L-401

Los Alamos National Laboratory
ATTN: MS222, J. Brownell

DEPARTMENT OF ENERGY CONTRACTORS (Continued)

Sandia National Lab
ATTN: M. Clauser, Org 5241
ATTN: G. Kuswa, Org 5240
ATTN: G. Yonas
ATTN: Tech Lib 3141
ATTN: J. Powell
ATTN: Org 9336, D. Allen

DEPARTMENT OF DEFENSE CONTRACTORS

Advanced Research & Applications Corp
ATTN: R. Armistead

Aerospace Corp
ATTN: V. Josephson
ATTN: S. Bower
ATTN: Technical Information Services

BDM Corp
ATTN: Corporate Library

BDM Corp
ATTN: L. Hoeft

Boeing Co
ATTN: Aerospace Library

Dikewood
ATTN: Tech Lib for L. Davis

EG&G Wash Analytical Svcs Ctr, Inc
ATTN: Library

General Electric Co
ATTN: J. Peden
ATTN: H. O'Donnell

IRT Corp
ATTN: R. Mertz

JAYCOR
ATTN: E. Wenaas

JAYCOR
ATTN: E. Alcaraz
ATTN: R. Sullivan
4 cy ATTN: R. Terry
4 cy ATTN: J. Guillory

Kaman Sciences Corp
ATTN: S. Face

Kaman Tempo
ATTN: DASIAC

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Lockheed Missiles & Space Co, Inc
ATTN: L. Chase

Lockheed Missiles & Space Co, Inc
ATTN: S. Taimuty, Dept 81-74/154

Maxwell Labs, Inc
ATTN: A. Kolb
ATTN: D. Tanimoto
ATTN: A. Miller
ATTN: O. Cole

McDonnell Douglas Corp
ATTN: S. Schneider

Mission Research Corp
ATTN: W. Hart
ATTN: C. Longmire

Mission Research Corp
ATTN: B. Godfrey

Mission Research Corp, San Diego
ATTN: B. Passenheim

Pacific-Sierra Research Corp
ATTN: H. Brode, Chairman SAGE
ATTN: L. Schlessinger

Physics International Co
ATTN: C. Gilman
ATTN: C. Stallings
ATTN: G. Frazier

R&D Associates
ATTN: P. Haas
ATTN: A. Latter

R&D Associates
ATTN: P. Turchi

S-CUBED
ATTN: A. Wilson

Science Applications, Inc
ATTN: K. Sites

TRW Electronics & Defense Sector
ATTN: Technical Information Center
ATTN: D. Clement

END

FILMED

9-83

DTIC