CONF-9409105--5

UCRL-JC-117800
PREPRINT

Unstructured 3-D Electromagnetic Calculations Using the Discrete Surface Integration Method

Niel Madsen, David Larson, Scott Brandon, Grant Cook
J. Brian Grant, David Steich, Dale E. Nielsen, Jr.

This paper was prepared for submittal to the
15th International Conference on the Numerical Simulation of Plasmas
Valley Forge, PA
September 7, 1994

September 1994

Lawrence Livermore National Laboratory

DISCLAIMER

DISCLAIMER

Unstructured 3-D Electromagnetic Calculations Using the Discrete Surface Integration Method

15th International Conference on the Numerical Simulation of Plasmas

September 7, 1994

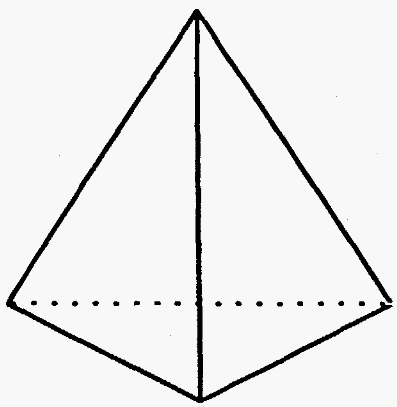
Niel Madsen, David J. Larson, Scott Brandon,

Grant Cook, J. Brian Grant, David Steich, and Dale E. Nielsen Jr.

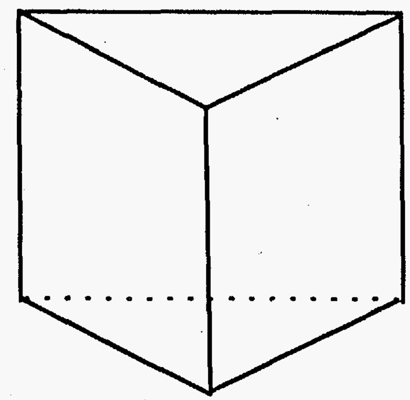
Lawrence Livermore National Laboratory

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

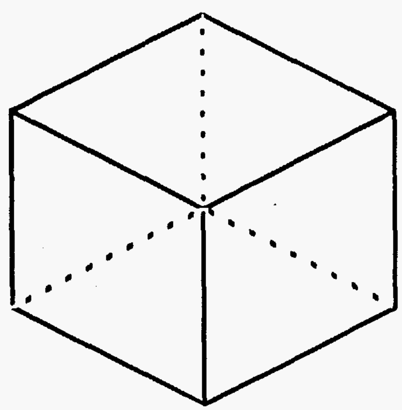
Supported Elements



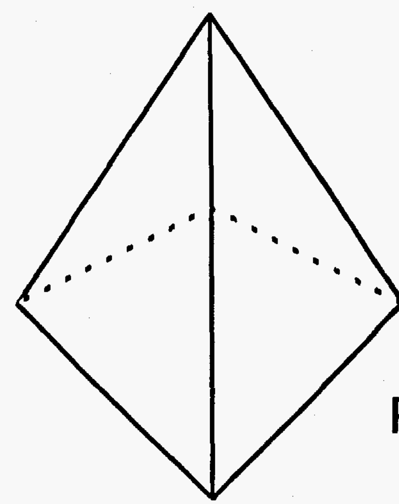
Tetrahedron



Triangular Prism

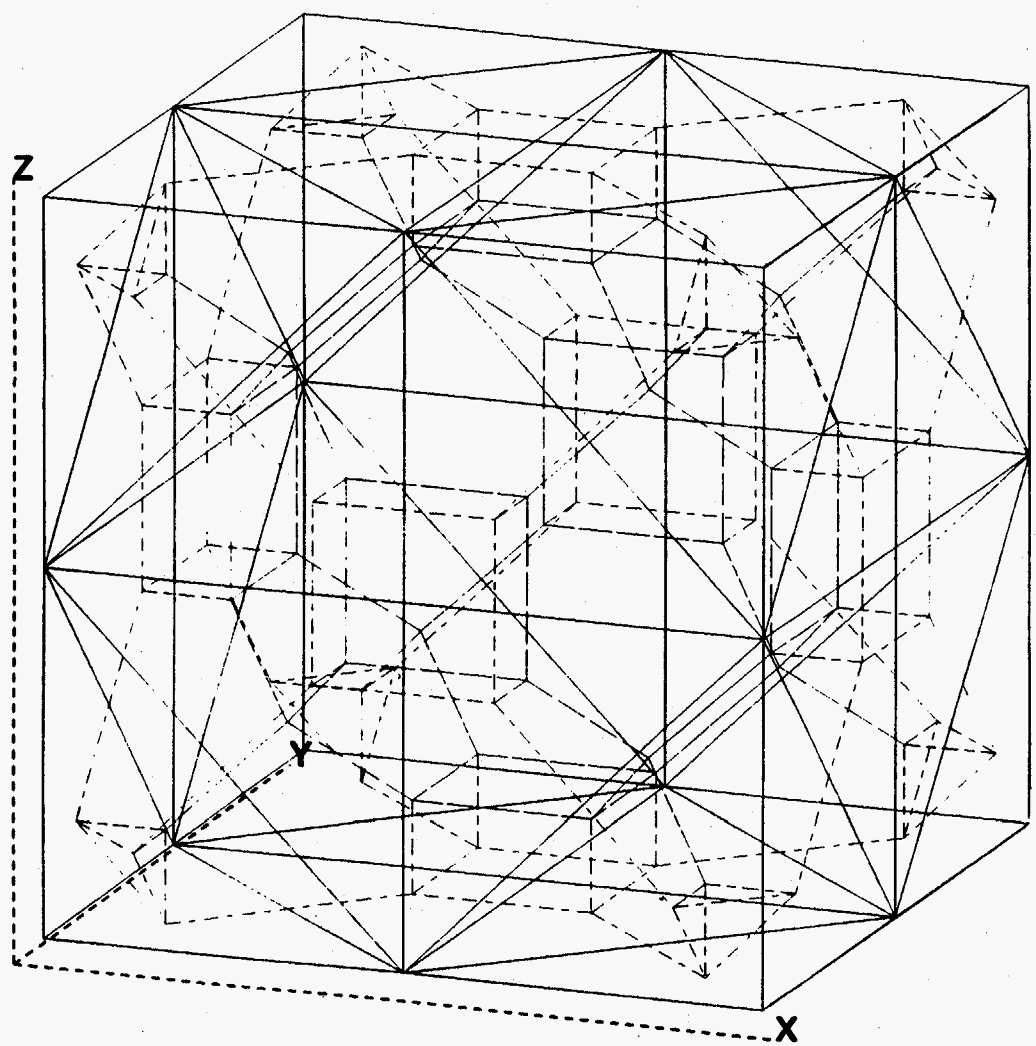


Hexahedron



Pyramid

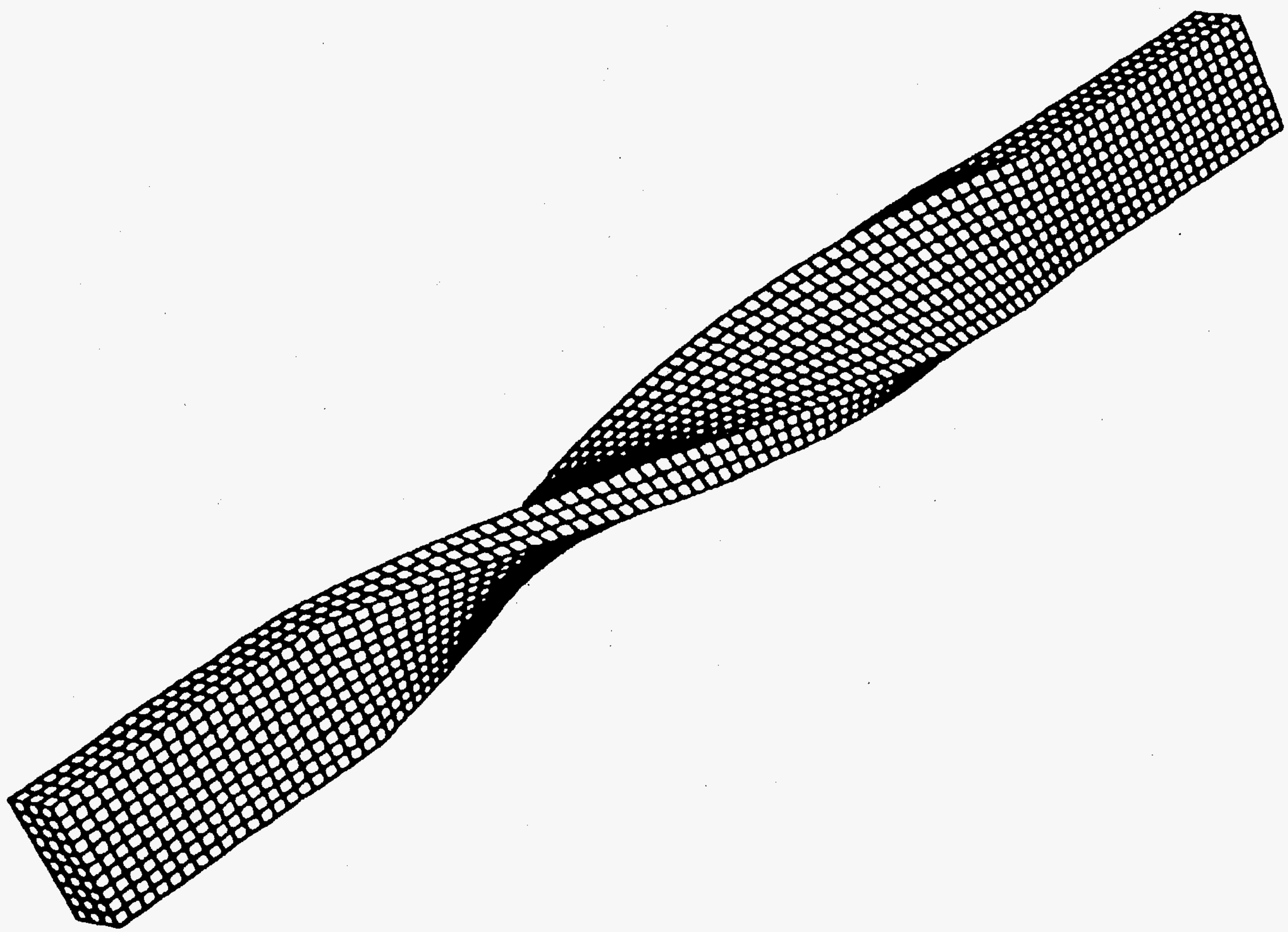
Primary and Dual Grids



View Angle (75.00, 20.00)

Distorted Regular Grid

Twisted waveguide discretized into hexahedral cells.

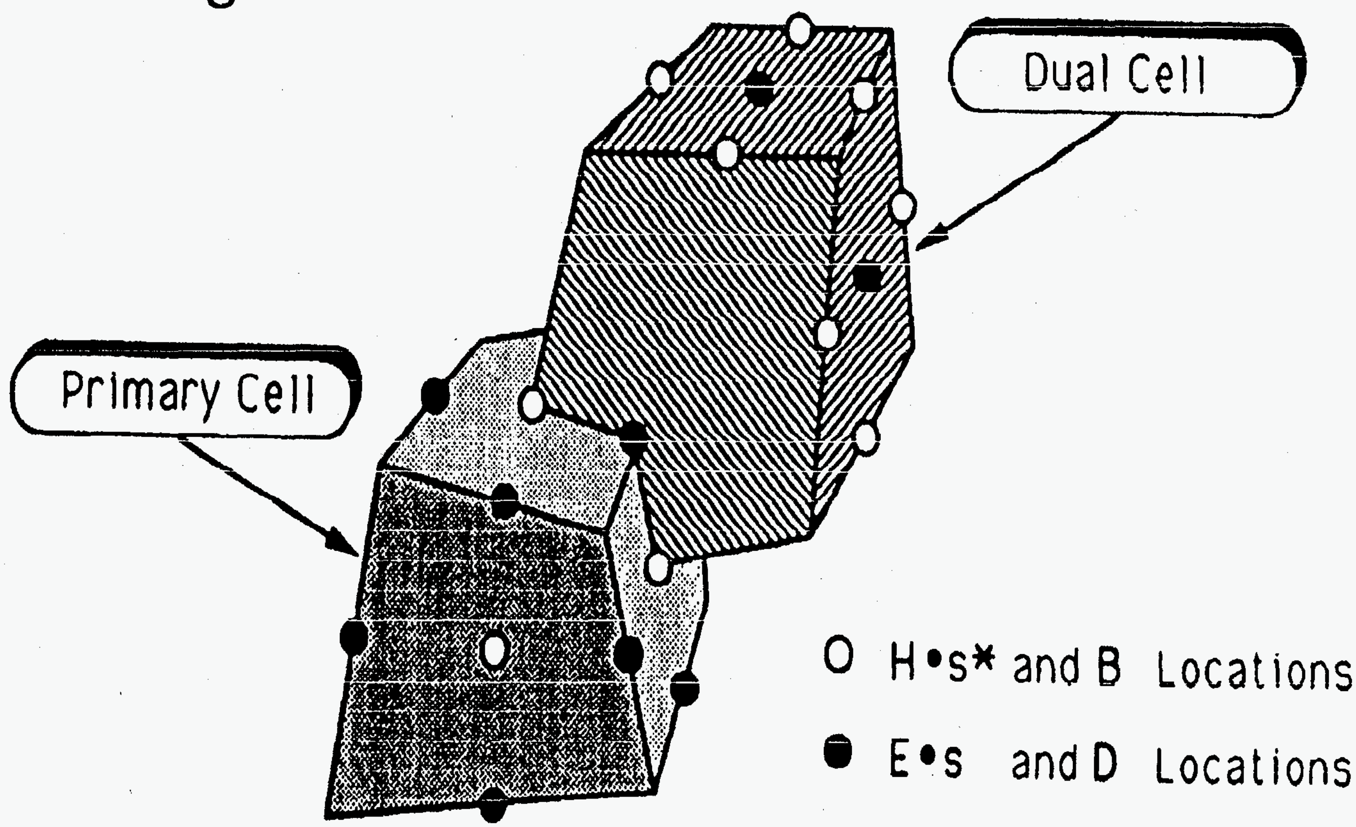


Criteria for a reliable/robust 3-D electromagnetic field solve

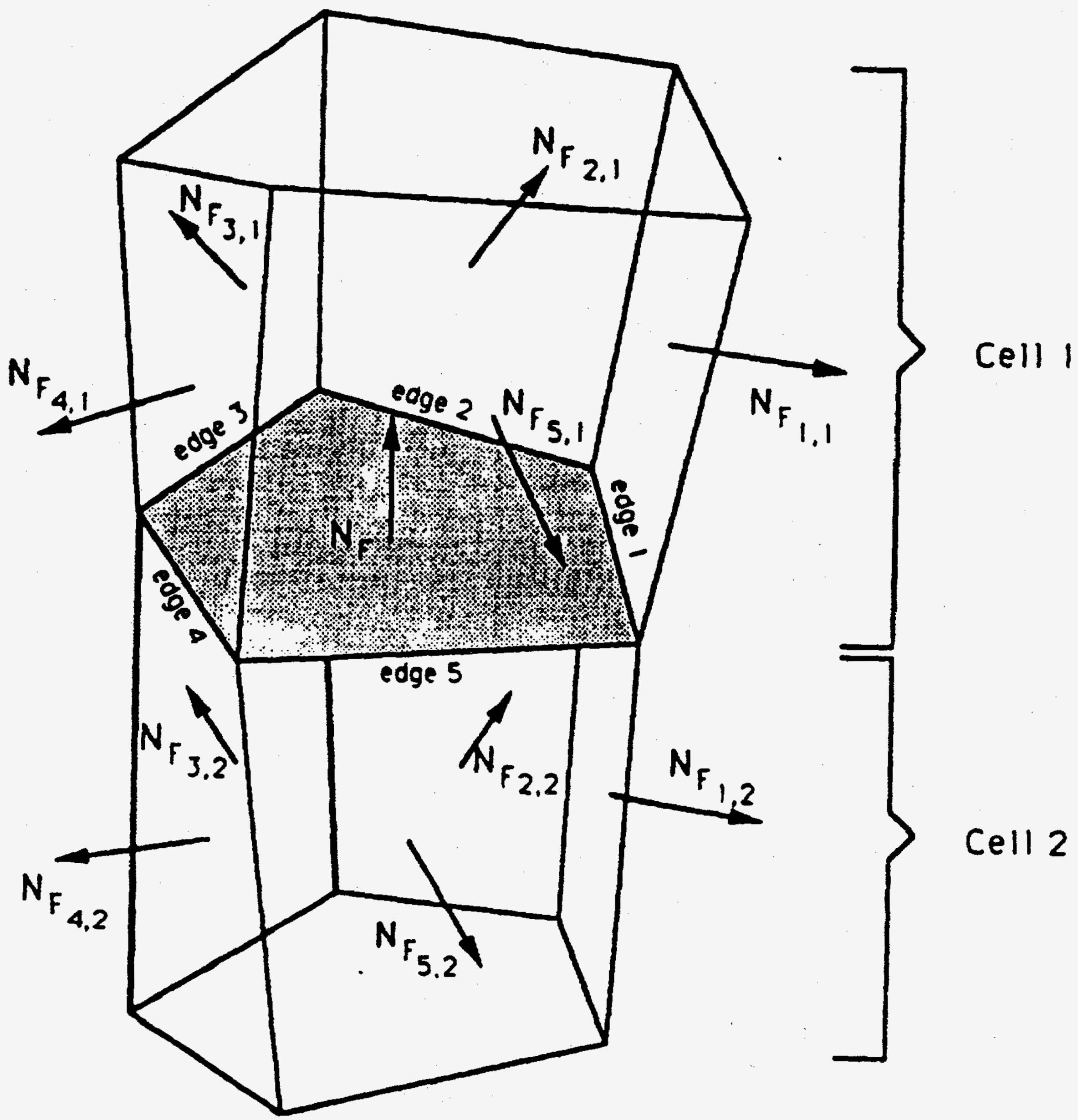
- allow use of unstructured non-orthogonal grids while retaining accuracy
- allow a variety of cell or element types
- reduce to standard FDTD method when orthogonal grids are used
- preserve charge/divergence locally (and globally)
- conditionally stable
- non-dissipative

Spatial Discretization

Electric and magnetic field locations relative to the primary and dual grid cells

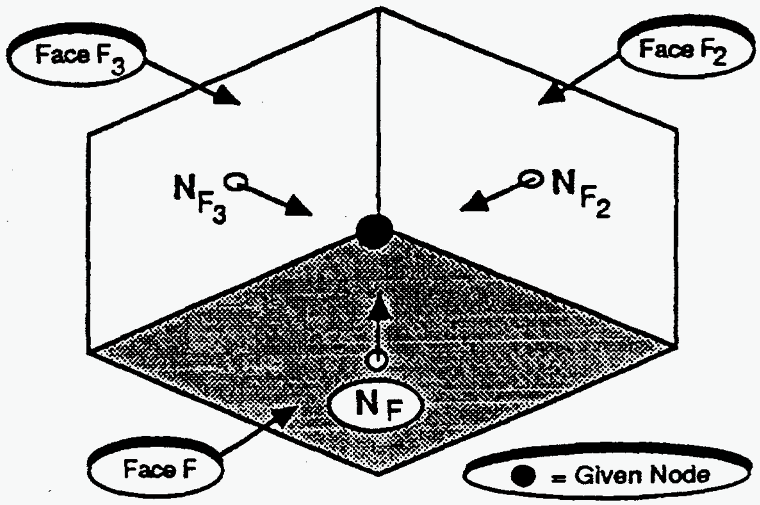


Face values for B advance



APS-11-93-7-DL&SB

Node values of B dot



$$\frac{dB_{ij}^{k}}{dt} \bullet N_F = -\oint_F E^k \bullet dl$$

$$\frac{dB_{ij}^{k}}{dt} \bullet N_{F_{i,j}} = -\oint_{F_{i,j}} E^k \bullet dl$$

$$\frac{dB_{ij}^{k}}{dt} \bullet N_{F_{m,j}} = -\oint_{F_{m,j}} E^k \bullet dl$$

APS-11-93-8-DL&SB

Face values of B dot

The node B dot values are averaged to obtain a single B dot value for the face

The fully volume weighted scheme

$$\frac{dB_F^k}{dt} = \frac{\sum_{j=1}^{N_c} \sum_{i=1}^{P} \left|\omega_{ij}\right| \frac{dB_{ij}^k}{dt}}{\sum_{j=1}^{N_c} \sum_{i=1}^{P} \left|\omega_{ij}\right|}$$

where the weight

$$\omega_{ij} = N_F \bullet \left(N_{F_{i,j}} \times N_{F_{i+1,j}}\right)$$

represents the volume of the jth coordinate system at node i of face F

works out well in our implementation.

DSI* algorithm for B

Face aligned B dot

$$\frac{\partial B^k}{\partial t} \bullet N_F \equiv \int_F \left(\frac{\partial B^k}{\partial t} \bullet n \right) dS = -\int_F \left(\nabla \times E^k \right) \bullet n \, dS$$

$$= -\oint_F E^k \bullet dl$$

Node centered B dot

$$\frac{dB^k_{ij}}{dt} \bullet N_F = -\oint_F E^k \bullet dl$$

$$\frac{dB^k_{ij}}{dt} \bullet N_{F_{i,j}} = -\oint_{F_{i,j}} E^k \bullet dl$$

$$\frac{dB^k_{ij}}{dt} \bullet N_{F_{m,j}} = -\oint_{F_{m,j}} E^k \bullet dl$$

Interpolate to face centers

$$\frac{dB^k_F}{dt} = \frac{\sum_{j=1}^{N_c} \sum_{i=1}^{P} \left| \omega_{ij} \right| \frac{dB^k_{ij}}{dt}}{\sum_{j=1}^{N_c} \sum_{i=1}^{P} \left| \omega_{ij} \right|} \qquad \omega_{ij} = N_F \bullet \left(N_{F_{i,j}} \times N_{F_{i+1,j}} \right)$$

Leapfrog advance

$$B^{k+\frac{1}{2}} = B^{k-\frac{1}{2}} + \Delta t \frac{dB^k}{dt}$$

Project to dual edge

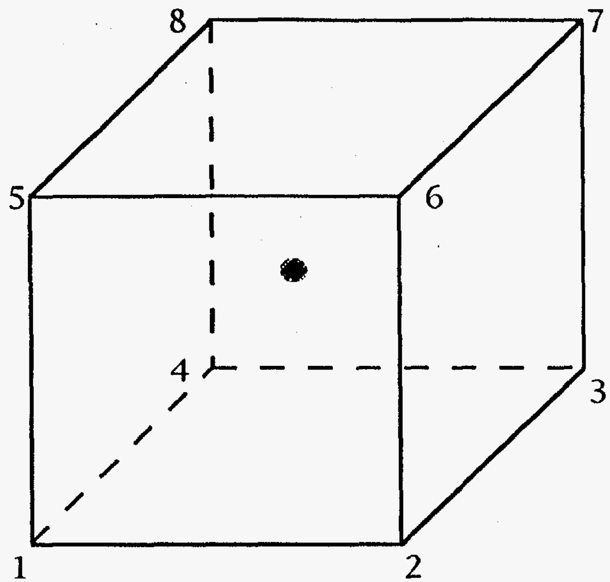
$$\left(H \bullet s^* \right)^{k+\frac{1}{2}} = \frac{B^{k+\frac{1}{2}} \bullet s^*}{\mu}$$

* Niel Madsen, February, 1992, UCRL-JC-109787

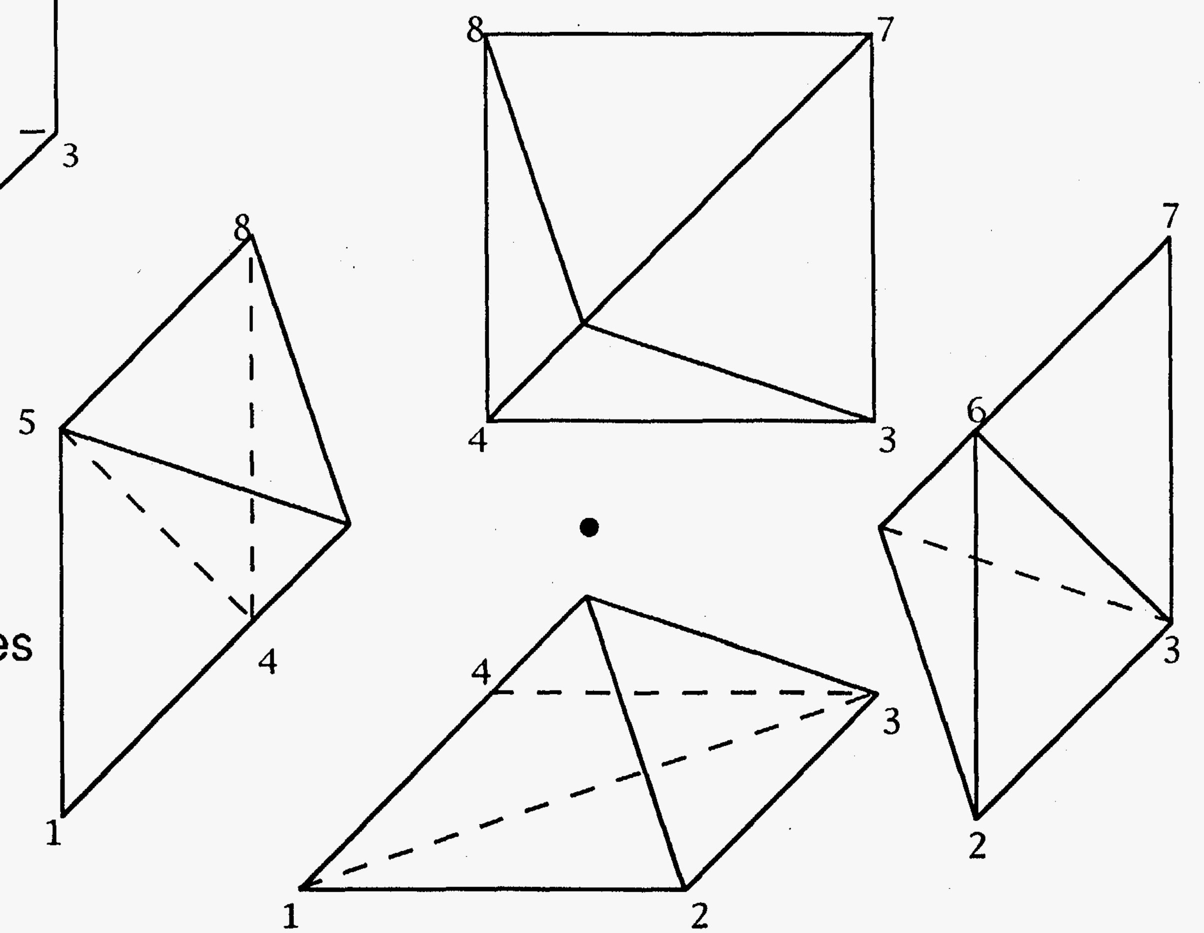
Particle Tracking

- Locate the element in which the particle resides. The first guess is the element associated with the particle's previous position.
- Calculate the interpolation weights required to deposit the particle current and to interpolate the E and B fields to the particle position.
- Mesh elements are decomposed into tetrahedral volumes defined by the particle location and the element faces.
- A negative weight indicates that the particle has left the current element and provides some clue as to where to look next.

Hexahedral Element



A simulation particle within a hexahedral mesh element.

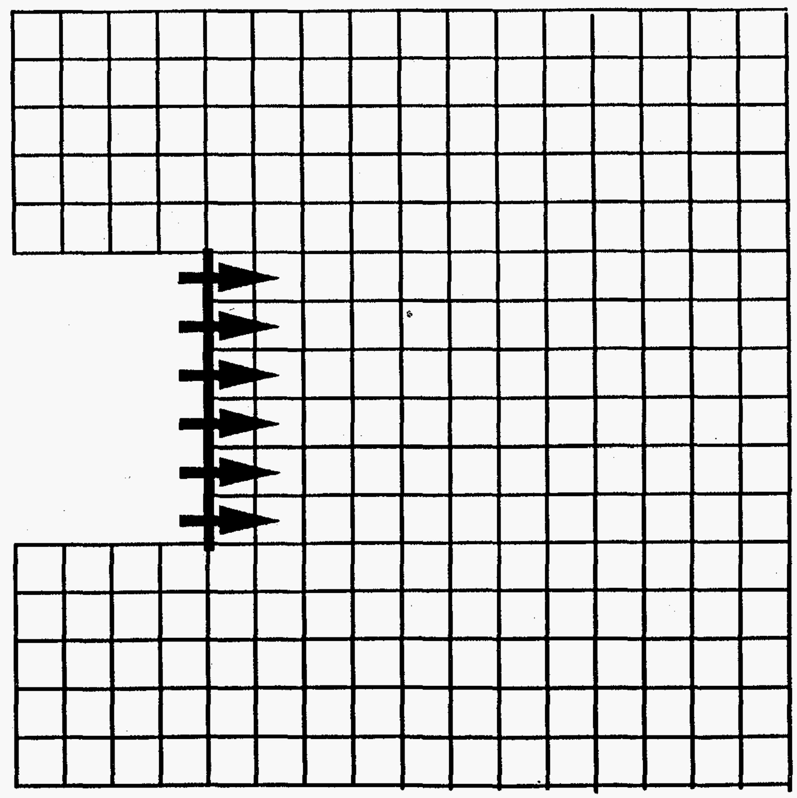


Eight of the twelve volumes required to calculate the interpolation weights.

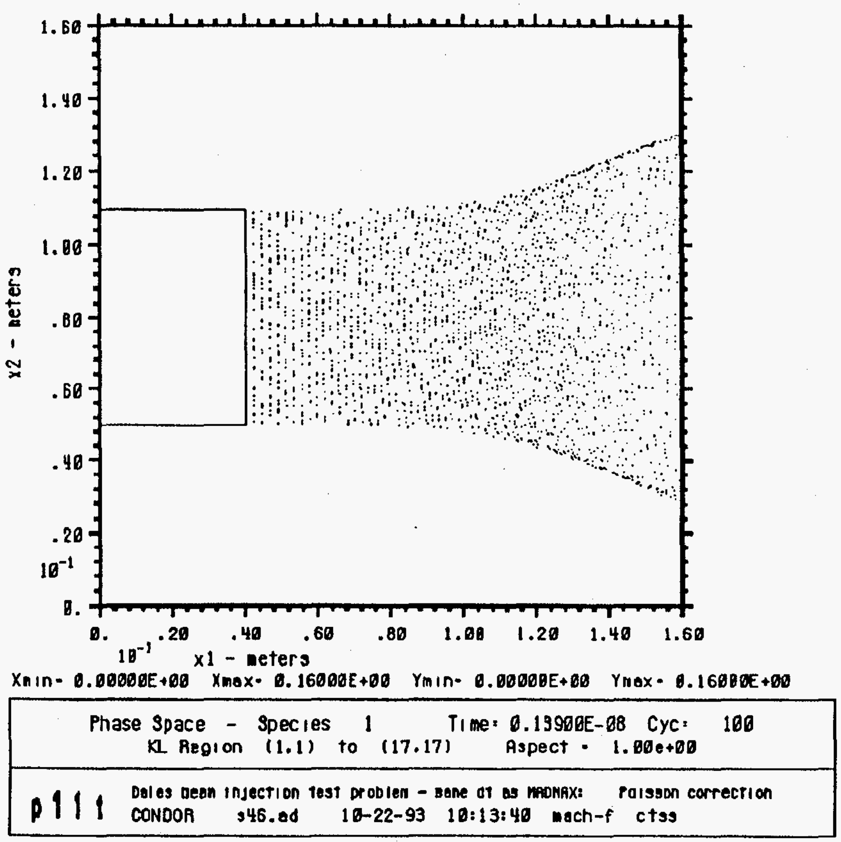
Test Problem

Test Problem: Space Charge "Almost" Limited Injection

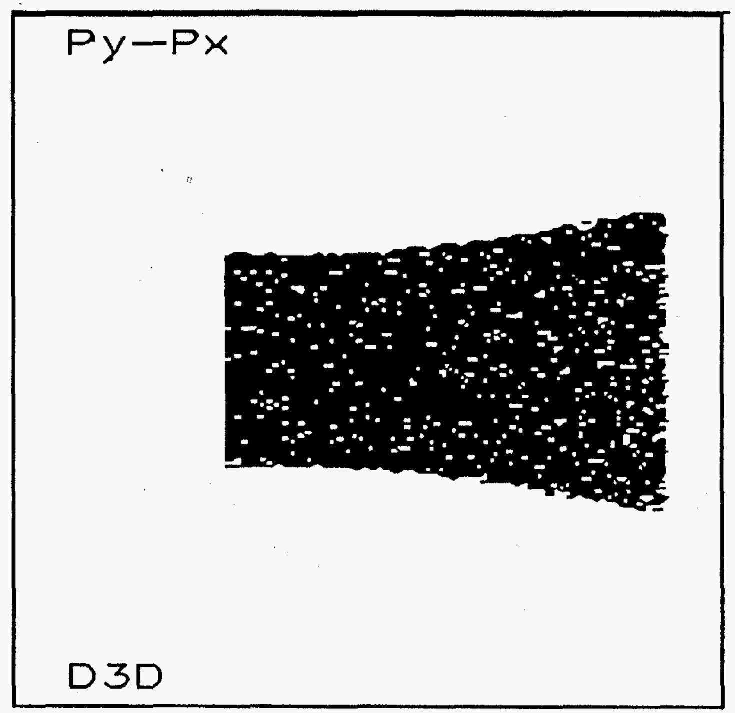
We inject 80 kA/m^2 of 100 keV electrons from all 36 zones on the face of the "pedastal" shown in the grid below. There are 16 zones in all three dimensions with $\Delta x = \Delta y = \Delta z = 0.01$ meters. A square mesh is chosen to allow comparison with the results from other well tested EM PIC codes. The "coarse" resolution accentuates the effects of interpolation.



Particle Postion

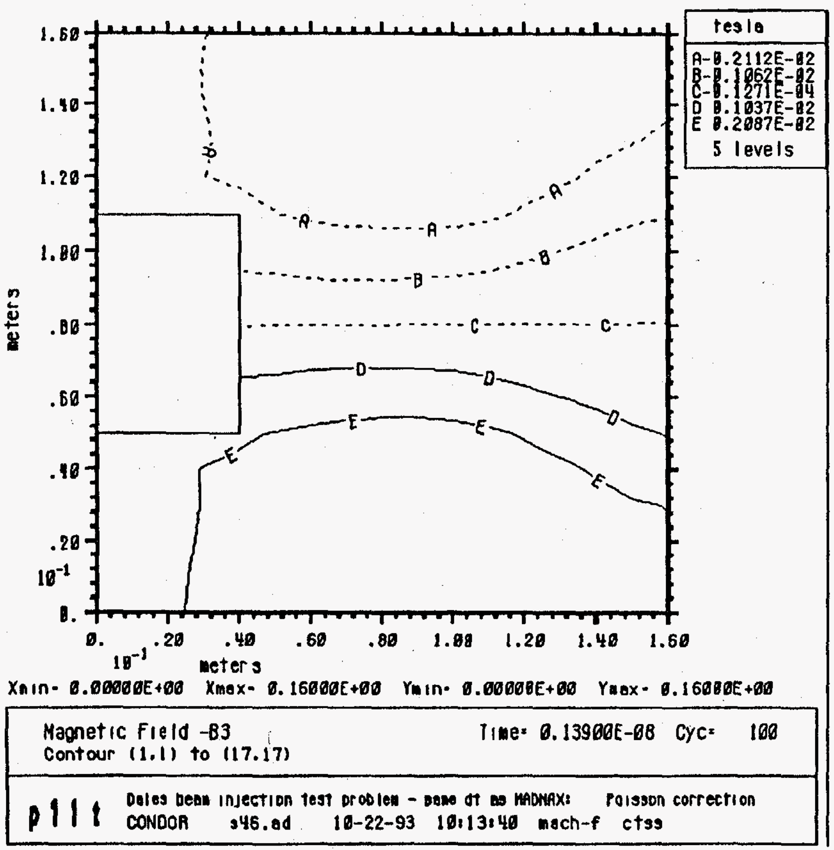


2-D Calculation

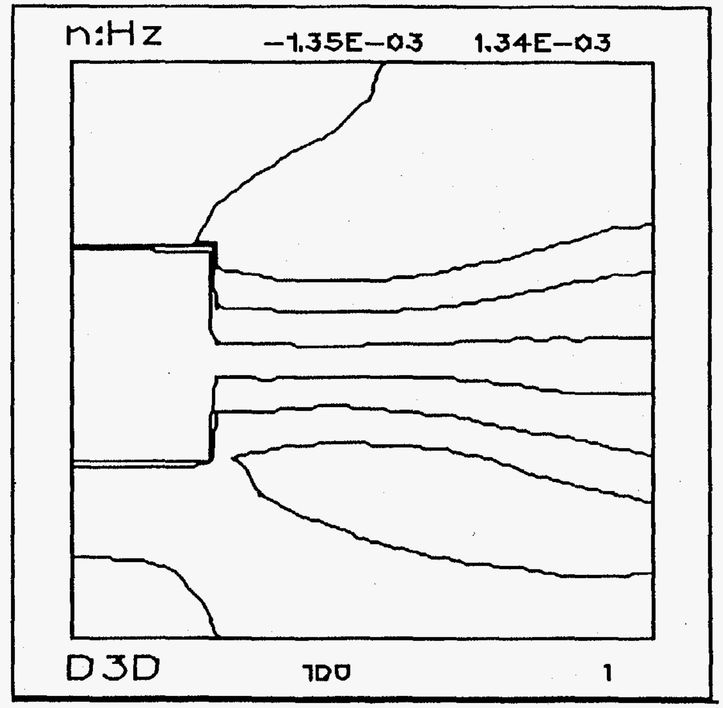


3-D Calculation

Bz Contour Plot

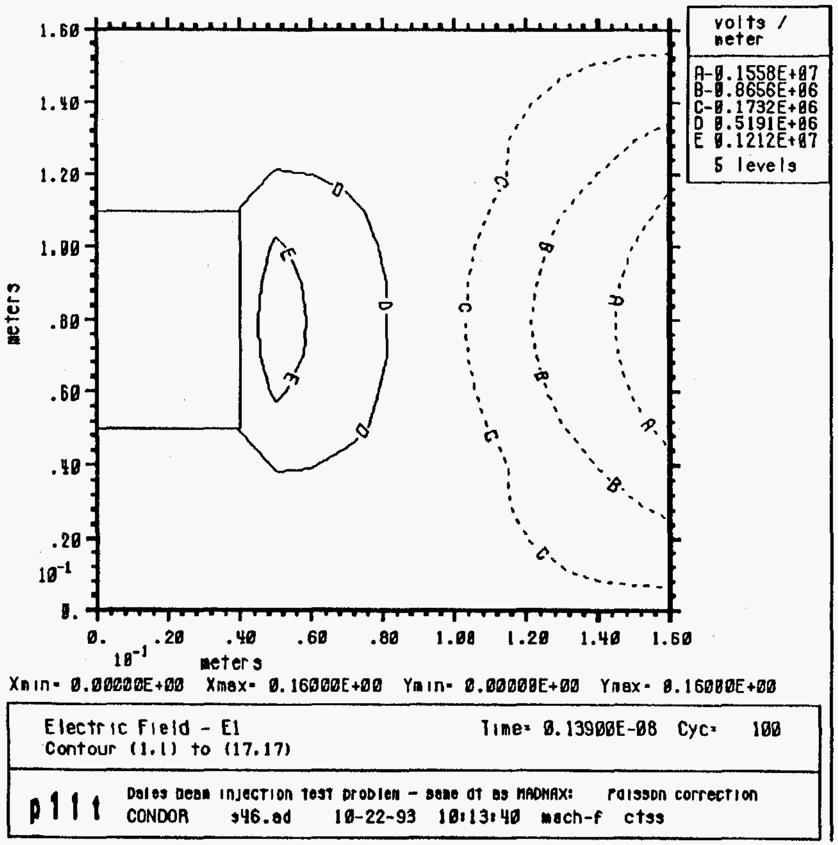


2-D Calculation

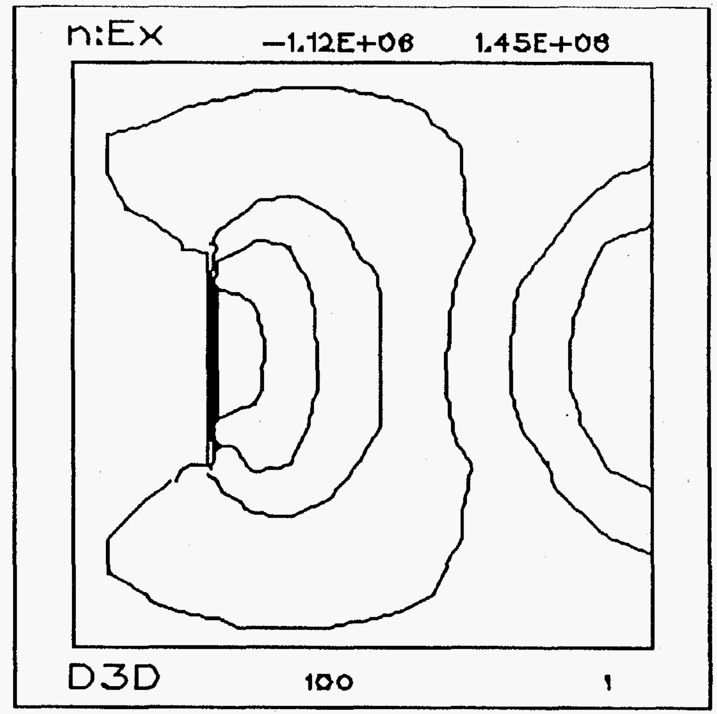


3-D Calculation

Ex Contour Plot



2-D Calculation



3-D Calculation

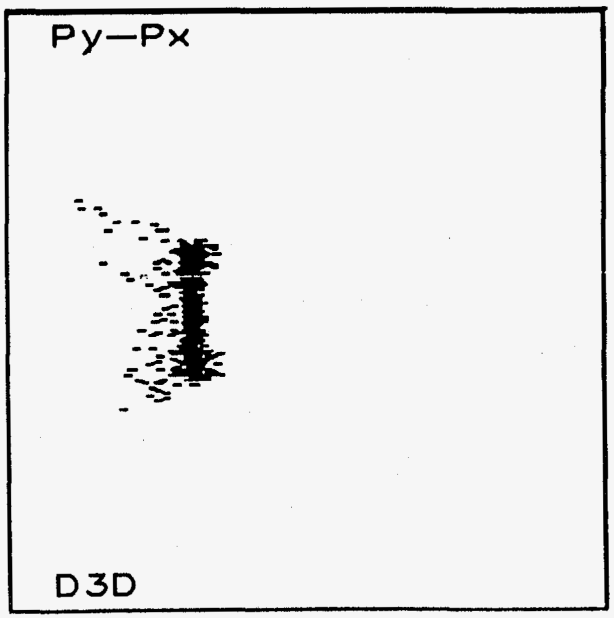
Marder current correction

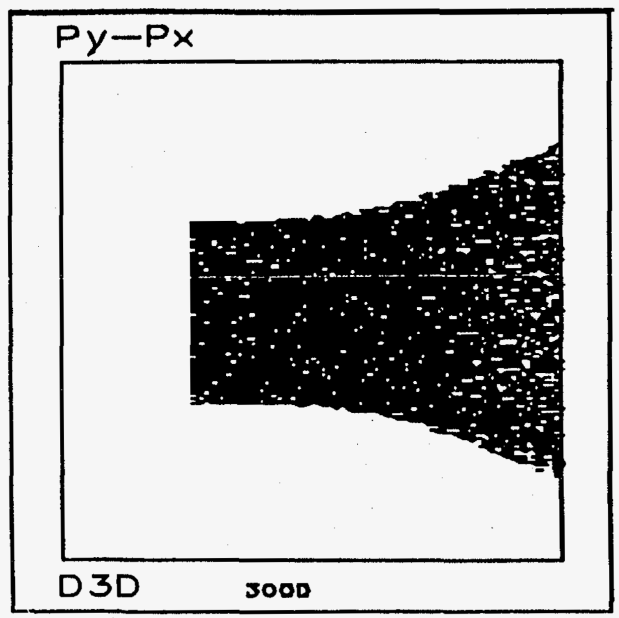
PROBLEM: Gauss' law $\nabla \bullet \mathbf{D} = \rho$ is not preserved due to numerical inaccuracies.

The Marder pseudo-current algorithm adds diffusion to Ampere's Law in order to bound the error in Gauss' Law.

Define: $F = \varepsilon_0 \nabla \bullet \mathbf{E} - \rho$ then $\varepsilon_0\, \partial \mathbf{E}/\partial t = \nabla \times \mathbf{B}/\mu_0 - \mathbf{J} + c_m \nabla F$

<u>3000 Δt without the Marder correction</u> <u>with the Marder correction</u>





Divergence Calculation

The conventional finite-difference method of calculating divergence on an orthogonal grid

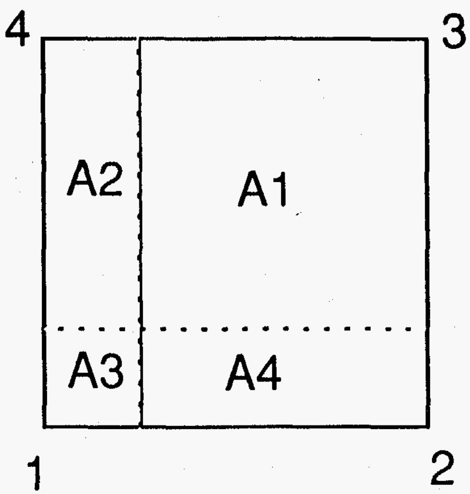
$$\nabla \bullet \mathbf{E} = (E_x^{i+1} - E_x^{i-1})/2\Delta x + \ldots$$

does not generalize to an unstructured grid. Instead we use the definition of divergence:

divergence = the outgoing flux per unit volume.

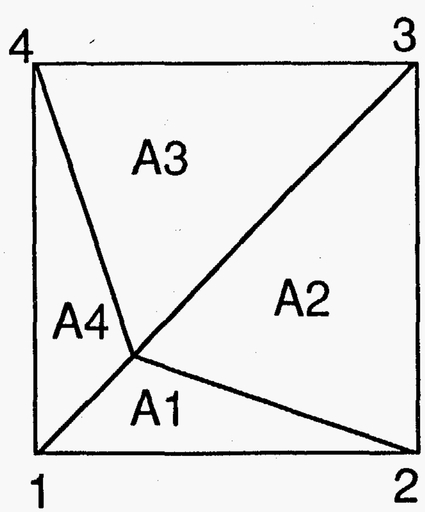
We obtain the divergence of **E** by taking the dot product of the primary edge **E** values with the associated dual-face area normals for all the primary edges connected to a primary node. This sum is then divided by the volume of the dual cell, yielding the divergence at the primary node.

Weighting Algorithms



Area weighting using conventional PIC technique on a 1 X 1 square:

W(n1) = Area 1 = 9/16
W(n2) = Area 2 = 3/16
W(n3) = Area 3 = 1/16
W(n4) = Area 4 = 3/16

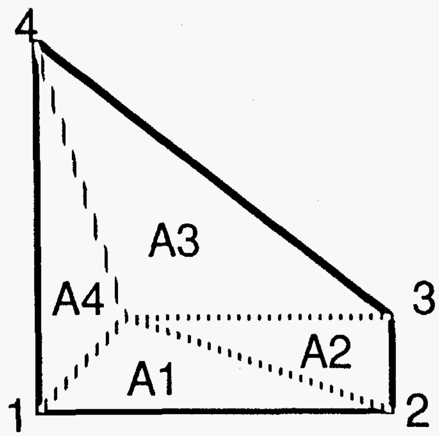


Weighting using our present side-based scheme on an "unstructured grid":

W(n1) = w2*w3 = 9/16
W(n2) = w3*w4 = 3/16
W(n3) = w1*w4 = 1/16
W(n4) = w1*w2 = 3/16

where A1 = 1/8 A2 = 3/8   A3 = 3/8   A4 = 1/8
and
w1 = A1/(A1+ A3) = 1/4   w2 = A2/(A2 + A4) = 3/4
w3 = A3/(A1+ A3) = 3/4   w4 = A4/(A2 + A4) = 1/4

Distorted Element

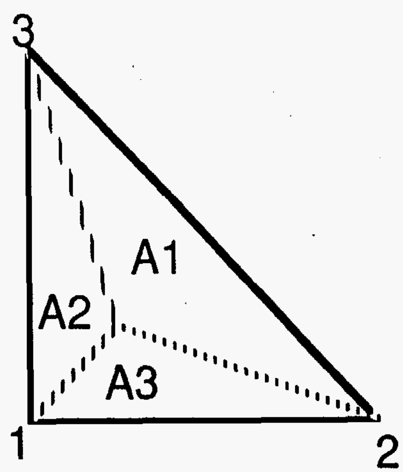


Our present hexahedral weighting algorithm produces erroneous results:

W(n1) = 27/91 W(n2) = 36/91
W(n3) = 16/91 W(n4) = 12/91

In addition we would like to have consistent weights calculated by our triangular and quadrilateral element algorithms. If side 2-3 of the above quad vanishes (the element becomes a triangle) the weights become

W(n1) = 0 W(n2) = 2/3
W(n3) = 1/3 W(n4) = 0



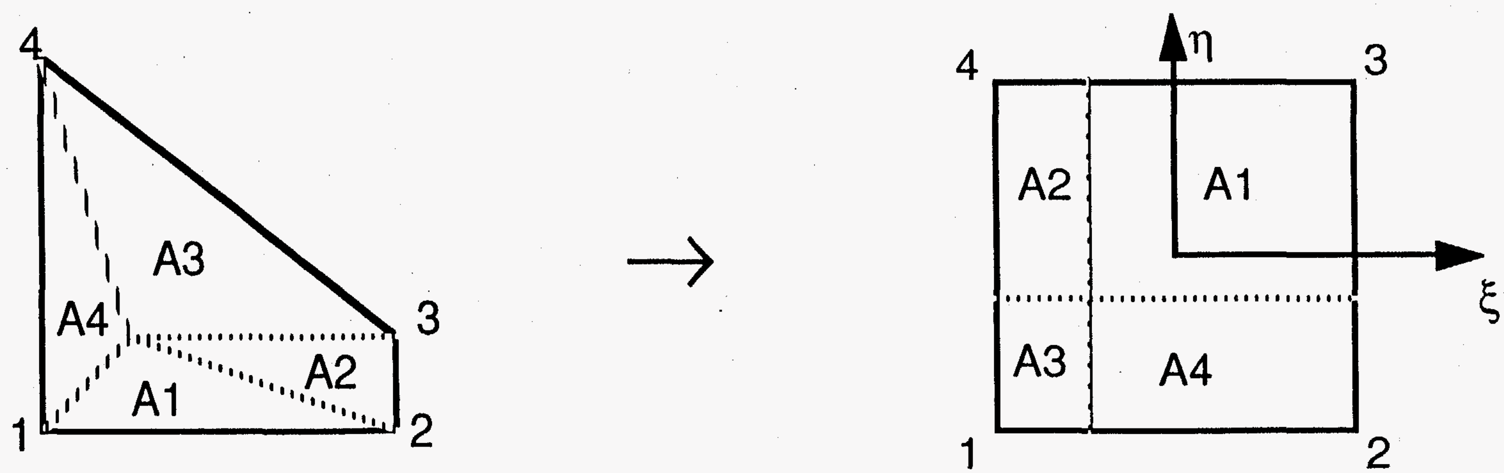
while our current triangular element weighting scheme yields

W(n1) = A1/(A1+A2+A3) = 1/2
W(n2) = A2/(A1+A2+A3) = 1/4
W(n3) = A3/(A1+A2+A3) = 1/4

Isoparametric Maps

We are considering the use of isoparametric maps in order to circumvent these difficulties. Mapping distorted elements to regular elements allows the use of standard bi or tri-linear weighting.

Consider a quadrilateral element → triangular element as side 2-3→ 0. The quadrilateral element maps to a 2x2 square in (ξ,η) space.



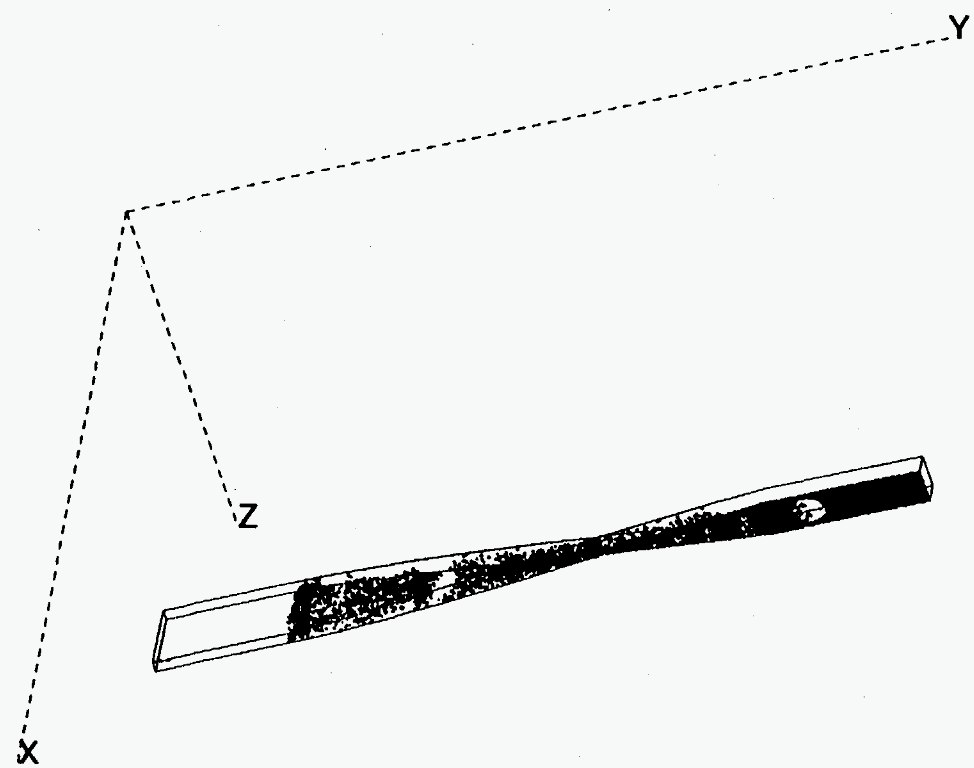
In the mapped coordinate space we use bilinear weighting

$W(n1) = (24+3\varepsilon)/(48+4\varepsilon) = 1/2$ $W(n2) = (8+\varepsilon)/(48+4\varepsilon) = 1/6$

$W(n3) = 1/(12+\varepsilon) = 1/12$ $W(n4) = 3/(12+\varepsilon) = 1/4$

As $\varepsilon \to 0$ the weights at nodes 2 and 3 combine to give a weight of 1/4. These results are consistent with the triangular element weighting scheme.

Field Emission in a Twisted Waveguide



Post Processing

- During a run the code dumps data that can be read by several different LLNL-developed graphics packages: GRIZ, PDBview, and Mesh-TV.
- GRIZ produced the twisted waveguide field plot.
- PDBview produced the primary and dual grid plots and the plot of particles in the twisted waveguide.

Parallel Particles

- The particle push will coexist nicely with the domain decomposed field blocks
- particles will be sorted in spatially distinct domains
- each particle domain will be completely contained in a single field domain
- every field block will keep at least 1 particle domain (may have zero particles)
- particle domains can be split or combined arbitrarily
- load balancing is accomplished by moving daughter particle blocks to idle processors (with a rather large communications penalty) or returning particle blocks to the original field processor

Computer Science

- Standard Fortran77 coding
- all variables explicitly declared
- COMMON blocks avoided
- no POINTER statements
- static memory management (for now)
- GRAPHICS based on GL or GKS
- source code control via CVS
- Parallel processing via domain decomposition
- Field regions split automatically via the Recursive Spectral Bisection algorithm (H. D. Simon - NASA Ames)
- About 4000 elements are required per node on the Intel iPSC/860 to amoritize communication time

Summary

- Self-consistent plasma simulation on grids composed of tetrahedrons, hexahedrons, triangular prisms, and pyramids.
- Successful simulation of beam injection and field emission in a twisted waveguide.
- Divergence errors controlled by the Marder pseudo-current algorithm.
- Simple particle and field boundary conditions.
- Future plans include more boundary condition options (periodic, etc.)
- Parallel implementation on the Meiko CS-2 is underway, with general procedures for handling parallel particles currently undergoing tests.