Recent Progress on the MaryLie/IMPACT Beam Dynamics Code

Robert D. Ryne
Lawrence Berkeley National Laboratory

in collaboration with:
D. Abell (Tech-X), A. Adelmann (PSI), J. Amundson (FNAL), E.W. Bethel (LBNL), A. Dragt (U. Md.), C.T. Mottershead (LANL, retired), F. Neri (LANL), I.V. Pogorelov (LBNL), J. Qiang (LBNL), R. Samulyak (BNL), J. Shalf (LBNL), C. Siegerist (LBNL), P. Spentzouris (FNAL), M. Venturini (LBNL), P. Walstrom (LANL)

ICAP2006
Chamonix, France
October 2-6, 2006

SciDAC Scientific Discovery through Advanced Computing

Office of Science U.S. Department of Energy

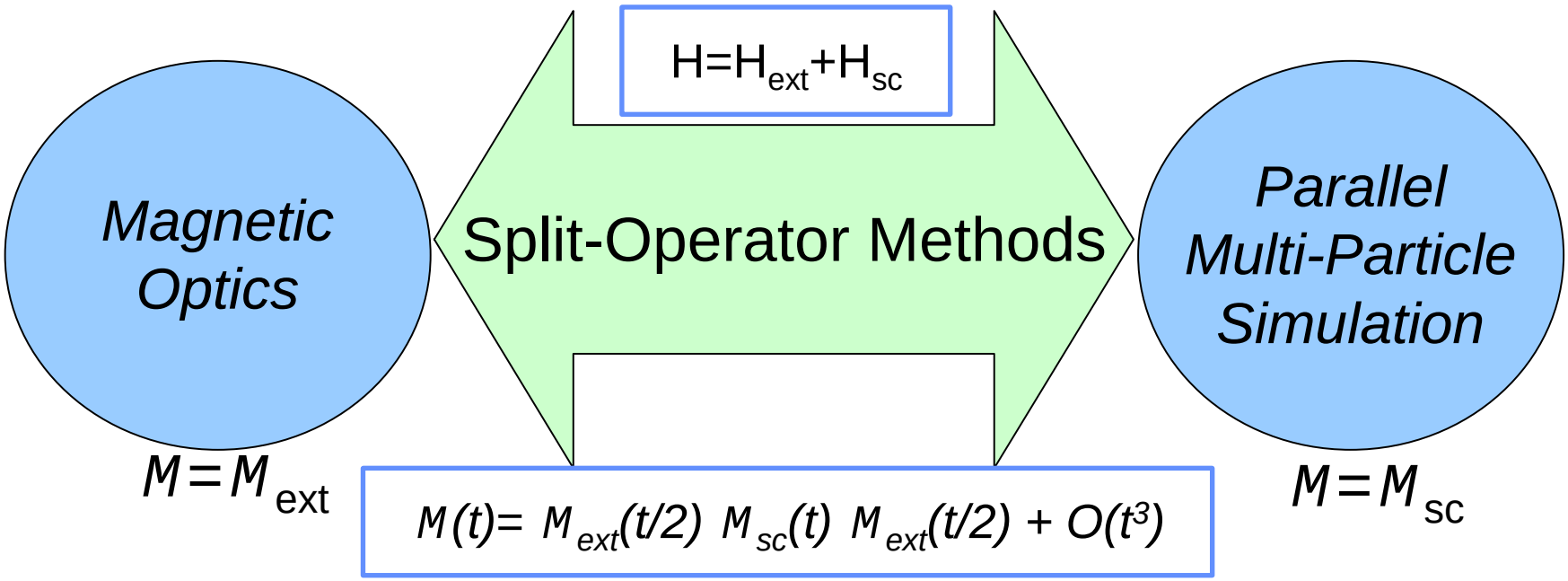
What is MaryLie/IMPACT (ML/I) ?

- **A 3D parallel PIC code that**
  - combines 5th order MaryLie beam optics with
  - IMPACT space charge & RF cavity modeling
  - and augments these with new capabilities
- **Multiple capabilities in a single environment**
  - Map generation and analysis, particle tracking, envelope tracking, fitting/optimization

Combines the functionality of high order optics code (MaryLie, others) and space-charge code (IMPACT, PARMILA) and rms envelope code (e.g. TRACE3D) in a single unified environment

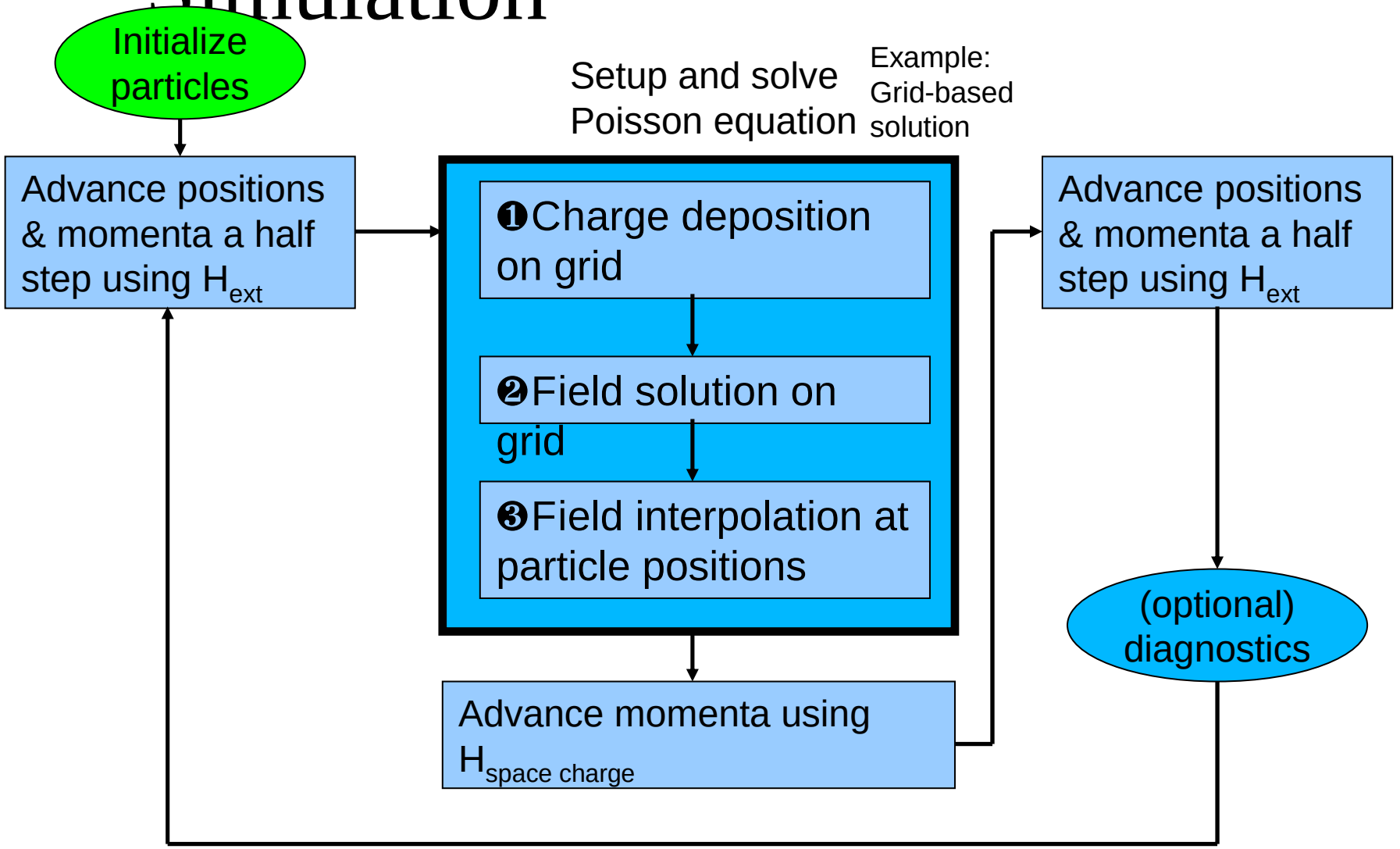
- **Other codes combine high order optics w/ space charge**
  - Synergia (talk by P. Spentzouris on Thursday)

Split Operator Approach



- **Note that the rapidly varying s-dependence of external fields is decoupled from slowly varying space charge fields**
- **Leads to extremely efficient particle advance:**
  - Do not take tiny steps to push ~100M particles
  - Do take tiny steps to compute maps; then push particles w/ maps

Particle-In-Cell (PIC) Simulation



What to do after a map is created?

- **Concatenate with previous maps**
- **Use it to track particles**
- **“Sandwich” with the previous map**
- **Use it to track envelopes**

Some ML/I features

- 5th order optics
- 3D space charge
- realistic RF fields
- acceleration
- ref traj computed “on the fly”
- 5th order rf cavity model (D. Abell)
- multi-reftraj for rf cavities
- envelope tracking
- soft-edged magnets
- methodical treatment of units
- MAD-style input compatibility
- “automatic” commands

MARYLIE/IMPACT

A Parallel Beam Dynamics Code with Space Charge
based on the MaryLie Lie Algebraic Beam Transport Code
and the IMPACT Parallel Particle-In-Cell Code *

Robert D. Ryne
Ji Qiang
Accelerator and Fusion Research Division
Lawrence Berkeley National Laboratory
Berkeley, California 94720

Alex J. Dragt
Department of Physics and Astronomy
University of Maryland
College Park, MD 20742

C. Thomas Mottershead
Filippo Neri
Peter Walstrom
LANSCE Division
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

Salman Habib
Theoretical Division
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

Roman Samulyak
Center for Data Intensive Computing
Brookhaven National Laboratory
Upton, New York 11973

Dan T. Abell
Tech-X Corporation
5621 Arapahoe Ave, Ste A
Boulder, CO 80303

Phillip Colella
Peter McCorquodale
David Serafini
Computational Research Division
Lawrence Berkeley National Laboratory
Berkeley, California 94720

Viktor Decyk
Physics Department
Univ. of California at Los Angeles
Los Angeles, CA 90095

August 2004

*Work supported in part by U.S. Department of Energy, Office of Science, Office of High Energy Physics and Office of Advanced Scientific Computing Research under the auspices of the Scientific Discovery through Advanced Computing (SciDAC) program and by grant DEFG02-96ER40949.

Treatment of Units in ML/I

- **ML/I allows arbitrary specification of `l`, ω, δ, where 6-vector = (x/`l`, p_x/δ, y/`l`, p_y/δ, ωt, p_t/ω`l`δ)**
  - Common choices are:
    - ✔Magnetostatic systems: δ=p_0, `l` =1, ω `l` /c=1
    - ✔Systems w/ acceleration: δ=m_0c, ω=ω_{bunch}, `l` =c/ω
- **New UNITS command:**

```
myunits, units: l=1.0, p=0.8, w=2.856e9
myunits, units: type=magnetic, l=1.0
myunits, units: type=dynamic, l=c/w
Etc.
```

New Commands (selected)

- `autoslice`: **automatic slicing of thick elements**
  - `SLICES =` *# of slices*`, L =` *distance between slices*`, CONTROL = local/global/none`
- `autoapply`: **automatic application of a commands**
  - `NAME=` *name of menu element or line*
- `autotrack`: **automatic tracking of particles**
  - `TYPE=<taylorN, symplecticN>, envelope=true/false`
- `autoconcat`: **automatic concatenation of maps**
- `poisson`: **select/set parameters of Poisson solver**
  - `NX=,NY=,NZ=,ngridpoints=` *fixed/variable*`, boundingbox=`*fixed/variable*...
- `units`: **specification of units**
  - `TYPE=<static, dynamic>, L =` *scale length*`, P =` *scale momentum* `, F =` *scale freq* `,W =` *scale angular freq*`, T =` *scale time*

ML/I Data Handling and I/O

- **Uses headers in particle data files (developed under SciDAC w/ FNAL) to ease macroparticle input from codes with different units**
  - Descriptive text, scale length, scale momentum, scale time
  - Frees the user from the headache of unit conversion
- **H5Part (implementation in ML/I under development)**
  - Parallel I/O crucial in very large scale simulations
  - See presentation by A. Adelmann, PSI
- **Working w/ Tech-X (D. Abell, K. Paul) to develop standard for map I/O**
  - Scale length, momentum, time
  - Initial reference trajectory
  - Final reference trajectory
  - Map coefficients

RMS Envelope Calculations

- **based on observation that envelope Hamiltonian is of the form $H=H_{ext}+H_{sc+emit}$**
- **envelope tracking can be done *using the same split-operator functionality in ML/I as is used for particle tracking***
  - `dotrack, autotrack: envelope=true`
- **Subroutine to update the envelopes due to H_{ext} is just a matrix-vector multiply**
- **Subroutine to compute "envelope kick" from $H_{sc+emit}$ involves calling math library routine to compute integrals in 3D envelope equations**
  - easily incorporated into other split-operator based space-charge simulation codes

Code for computing “envelope kick”

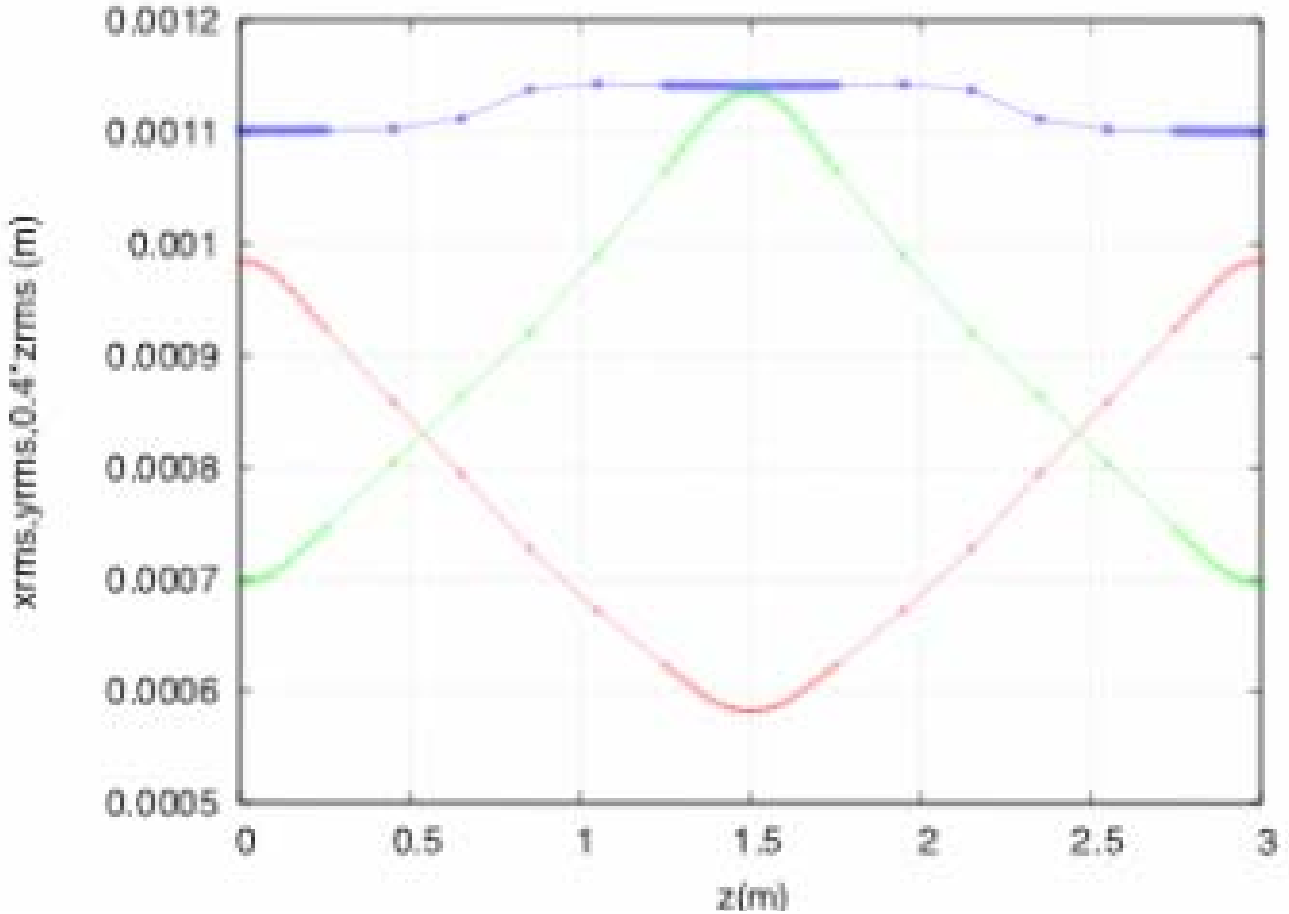
```
uu=env(1)**2
vv=env(3)**2
ww=(env(5)*gamma*beta*clite/(w*xl))**2

call scdrd(uu,vv,ww,g311,g131,g113)

qtot=bcurr/bfreq
xi0=xmc2*clite/(q*fpei)
xlam3=1.d0/(5.d0*sqrt(5.d0))
qcon=1.5d0*qtot*xlam3*clite/(xi0*(beta*gamma*xl)**2)*p0/xp
tcon=qcon*(gamma*beta*clite/(w*xl))**2

r11= qcon*g311*env(1) + emxn2*(xp/(p0*xl))/env(1)**3
r33= qcon*g131*env(3) + emyn2*(xp/(p0*xl))/env(3)**3
r55= tcon*g113*env(5) + emtn2/env(5)**3*(xl*xp*xw*2*xmp0**2/p0)

env(2)=env(2)+r11*tau
env(4)=env(4)+r33*tau
env(6)=env(6)+r55*tau
```

Test Suite

(developed in collaboration w/ A. Adelmann, J. Amundson, P. Spentzouris)

- KV beam in a FODO channel
- Free expansion of a cold, uniform density bunch
- Cold beam in a FODO channel with RF cavities
- Thermal beam in a constant focusing channel
- Bi-thermal beam in a constant focusing channel

Channel with RF



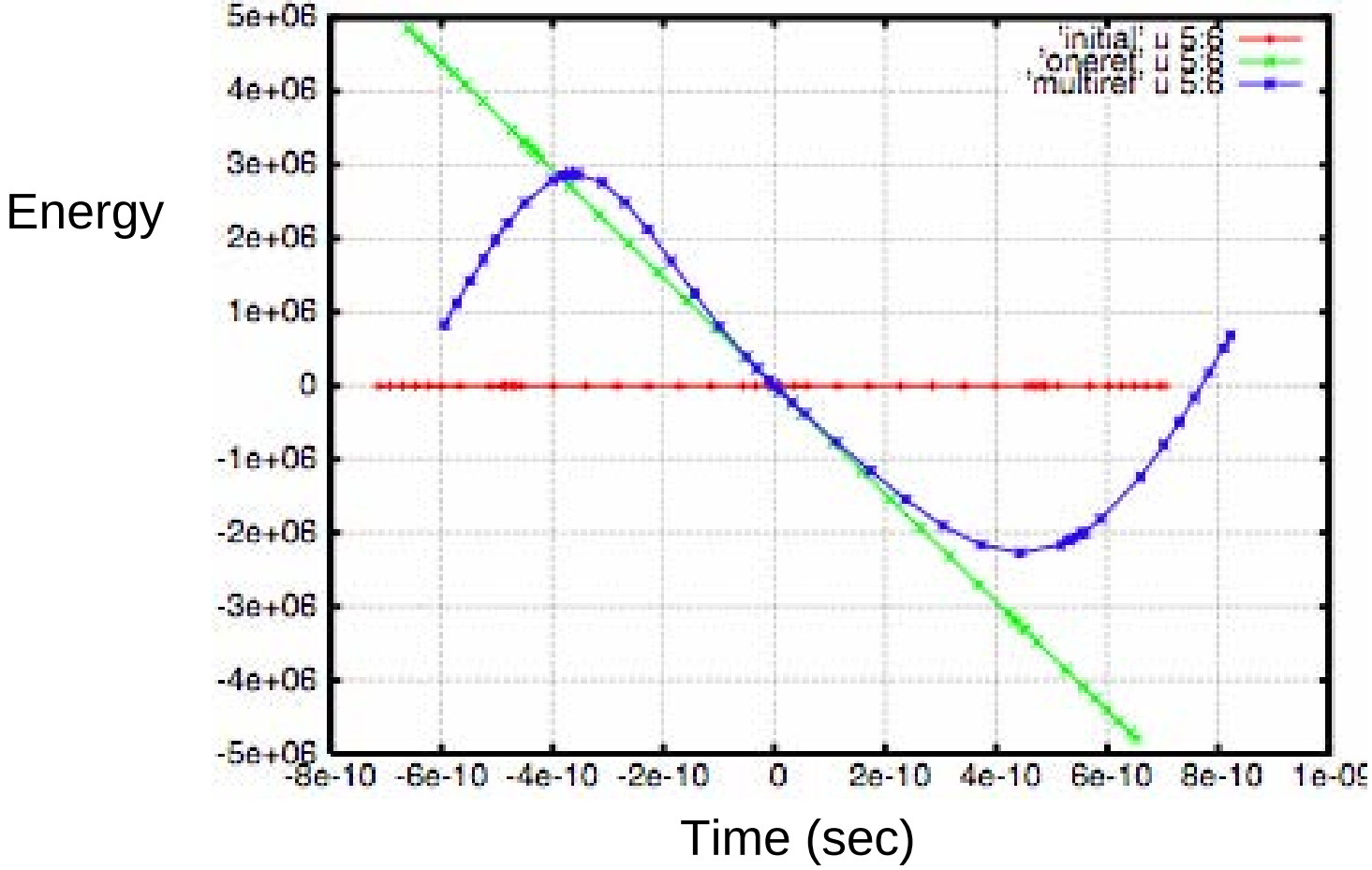
Plot of xrms, yrms, and 0.4*trms vs. distance in 1 period of a FODO channel with rf cavities. ML/I results (symbols) are on top of curves obtained from the rms equations

```
#comments
 Transport in a quad channel w/ rf cavities and space charge

#menu
 beam: beam,particle=proton,ekinetic=.250,bfreq=7.d8,bcurr=0.1d0
 units: units, type=dynamic, l=1.0d0, w=2.d0*pi*700.d6

dr: drift, l=0.10  slices=4
fquad: quadrupole, l=0.15 g1=6. lfrn=0. tfrn=0.  slices=6
dquad: quadrupole, l=0.30 g1=-6. lfrn=0. tfrn=0. slices=12

gapa1: rfgap,freq=7.d8,phasedeg=45.,file=f1,steps=100, slices=5
gapb1: rfgap,freq=7.d8,phasedeg=-1.,file=f2,steps=100, slices=5

pois: poisson, nx=64,ny=64,nz=128  !solver parameters

raysin: raytrace, type=readonly file1=partcl.data

dump: particledump,file=adump,sequencelength=100,precision=6

slice: autoslice, control=local
```

```
dotrack:autotrack, type=taylor1
post: autoapply, name=prntall
prntmoms: moments, precision=9, nunits=1
prntref: reftraj, precision=9, nunits=1
prntmax: maxsize, precision=9, nunits=1

cell,line=(fquad dr gapa1 dr dquad dr gapb1 dr fquad dump)
linac, line= 20*cell
prntall,  line=(prntmoms,prntref,prntmax)

#labor
slice    !slice elements
pois     !set poisson solver parameters
raysin   !read in some rays
prntall  !print moments and reference trajectory
dotrack  !tell the code to do autotracking
post     !after every slice, print moments
linac
end
```

Recent advances in RF cavity maps

- **5th order from field data on cylinder (D. Abell, Tech-X)**
- **Multiple reference particles in longitudinal phase space**
  - Used for tracking bunches with large phase & energy spread



Algorithmic Advances

- **Integrated Green function**
  - Solves the long-standing problem that has plagued certain codes (e.g. PARMELA, original IMPACT) when grid aspect ratios become large
- **Shifted Green function**
  - Originally developed for long-range beam-beam
  - Same algorithm works for cathode image effects
- **Wavelet-based methods**
- **Multi-level gridding, AMR**

Integrated Green Function (IGF) addresses a Critical Issue: high aspect ratios

- **Some Poisson solvers used in static electric and gravitational particle-in-cell simulations lose accuracy when the grid aspect ratio >> 1**
- **Some important problems involve extreme aspect ratios:**
  - Long beams in rf accelerators; pancake beams
  - Beams in induction linacs: L~ 10s of meters; R ~ cm
  - Galaxies
- **Standard grid-based approaches involve using a very large # of grid points in the long dimension, leading to prohibitively long run times**
  - As a result, it is ***extremely*** difficult model high aspect ratio systems accurately using standard grid-based approaches

IGF approach recognizes that certain physical quantities may vary on vastly different scales

- **The Green function, G, and source density, ρ, may change over different scales**
- **G is known apriori; ρ is not**

We should use all the information available regarding G so that the numerical solution is only limited by our approximate knowledge of ρ

- **Example: 2D Poisson equation in free space**

$$\phi(x,y) = \int G(x-x',y-y')\rho(x',y')dx'dy'$$

Standard Approach (Hockney and Eastwood)

$$\phi_{i,j} = \sum G_{i-i',j-j'} \rho_{i',j'}$$

- **This approach is equivalent to using the trapezoidal rule (modulo treatment of boundary terms) to approximate the convolution integral**
- **This approach makes use of only partial knowledge of G**
- **The error depends on how rapidly the integrand, ρG, varies over an elemental volume**
  - If ρ changes slowly we might try to use a large grid spacing; but this can introduce huge errors due to the change in G over a grid length

IGF Algorithm

- Assume the charge density, ρ, varies in a prescribed way in each cell
- Use the analytic form of the Green function to perform the convolution integral exactly in each cell, then sum over cells
- Example: linear basis functions to approximate ρ in a cell:

$$\phi(x_i,y_j)=\sum_{i',j'}\rho_{i,j}\int_0^{h_x}dx'\int_0^{h_y}dy'(h_x-x')(h_y-y')G(x_i-x_{i'}-x',y_j-y_{j'}-y')+$$

$$\sum_{i',j'}\rho_{i+1,j}\int_0^{h_x}dx'\int_0^{h_y}dy'\,x'(h_y-y')G(x_i-x_{i'}-x',y_j-y_{j'}-y')+$$

$$\sum_{i',j'}\rho_{i,j+1}\int_0^{h_x}dx'\int_0^{h_y}dy'(h_x-x')y'G(x_i-x_{i'}-x',y_j-y_{j'}-y')+$$

$$\sum_{i',j'}\rho_{i+1,j+1}\int_0^{h_x}dx'\int_0^{h_y}dy'x'y'G(x_i-x_{i'}-x',y_j-y_{j'}-y')$$

- Shifting the indices results in a single convolution involving an integrated effective Green function:

$$\phi_{i,j}=\sum G^{eff}_{i-i',j-j'}\rho_{i',j'}$$

Improvement over Standard Approach

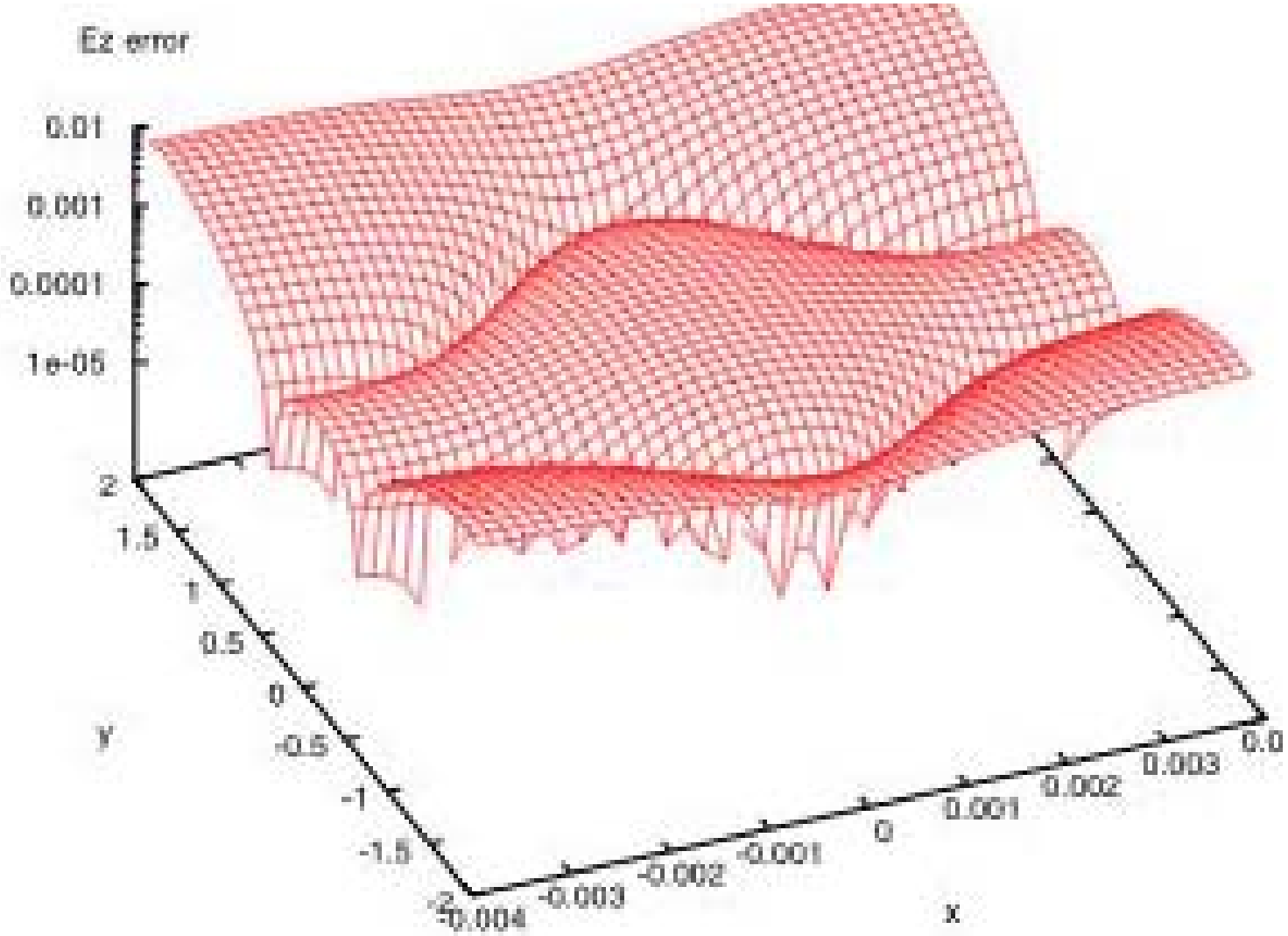
- **Cost: IGF elemental integrals can be done analytically; formulas are very lengthy**
  - Requires more FLOPS than simply using G_{ij} but…
  - In situations where the grid is fixed, this only needs to be done once at the start of a run. Amortized over many time steps, this does not significantly impact run time.
- **Accuracy: Method works as long as the elemental integrals are computed accurately and as long as the grid and # of macroparticles are sufficient to resolve variation in ρ**
- **IGF maintains accuracy even for extreme aspect ratios (>1000:1)**

As a result, IGF performs orders of magnitude better than the standard convolution algorithm for realistic problems involving large aspect ratios

Example: 2D gaussian ellipse

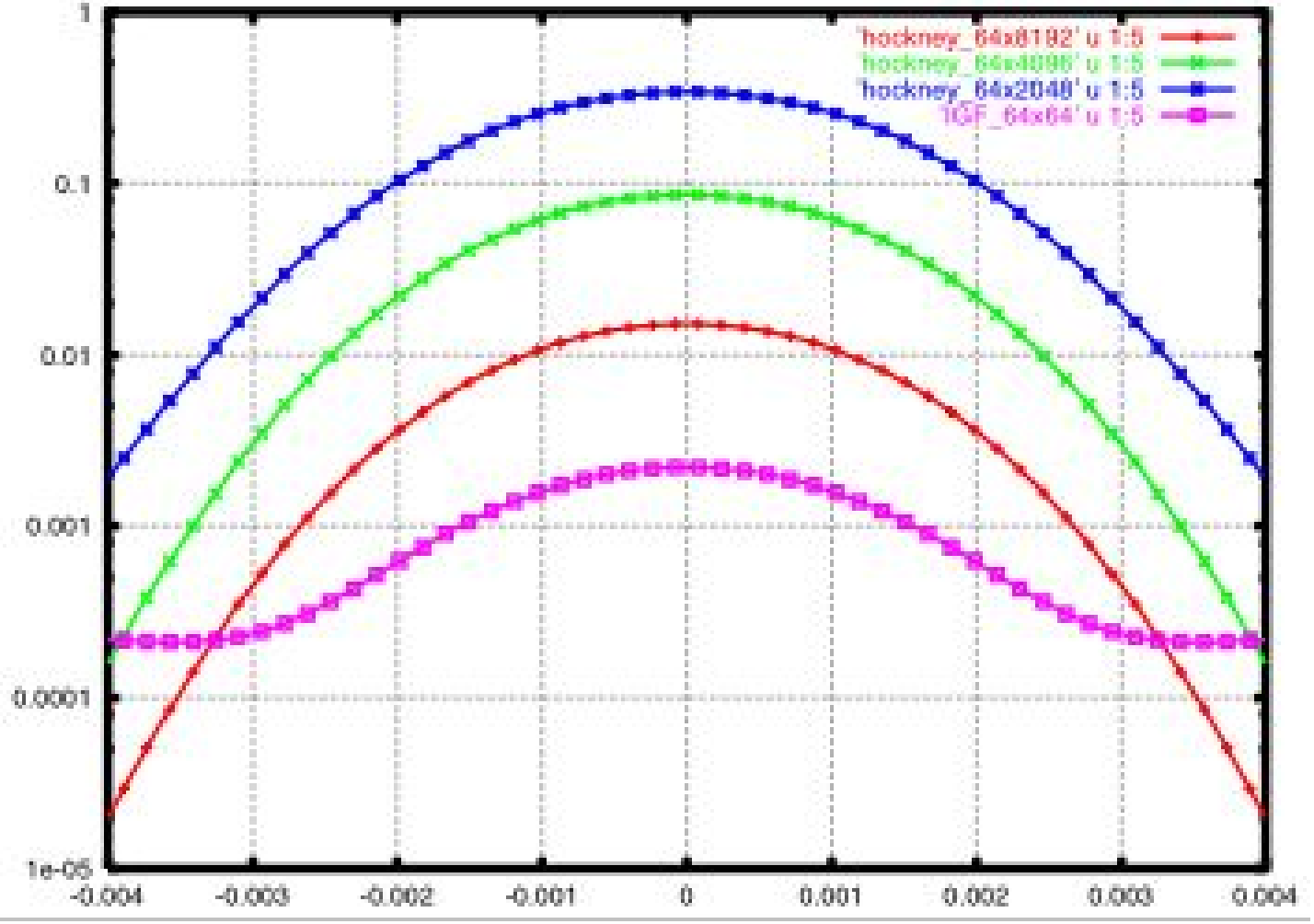
- **Aspect ratio is 1:500 -- x_{max}=0.002, y_{max}=1**
- **Calculation of fields using (1) standard Hockney algorithm and (2) IGF approach**
  - In both cases, performed convolutions for the fields directly (rather than calculating the potential and using finite differences to obtain fields)
- **Calculation performed using a mesh of size**
  - Hockney: 64x64, 64x128, 64x256,…, 64x16384
  - IGF: 64x64

IGF field error



Electric field error using IGF is below 1% using a 64x64 grid.

Comparison of IGF vs standard Hockney approach



Simulation of a high-aspect ratio bunch using an integrated Green Function (IGF) and a conventional algorithm (Hockney). IGF on a 64x64 grid (purple) is more accurate than a standard calculation using 64x2048 (blue), 64x4096 (green), and 64x8192 (red).

IGF summary

- **For the 2D Gaussian test problem, the standard Hockney algorithm would require ~500 times more computational effort to achieve the same worst-case accuracy as a simulation using the IGF approach.**
- **IGF works whether the aspect ratio is large, small, or near unity, i.e. it is generally applicable.**
- **2D implemented**
- **3D implemented w/ constant basis function**
- **3D w/ linear basis functions leads to messy formulae**
  - Collaborating w/ D. Abell (Tech-X) to produce improved implementation

Extension of IGF to Beams in Pipes

- IGF is especially useful when applied to beams in pipes, since the Green function falls off exponentially in z, though $\rho(z)$ may change slowly over meters
- Due to shielding in beampipe, sum can be truncated in the "long" direction:

$$\phi_{i,j} = \sum_{i'=1}^{N_x} \sum_{j'=j}^{j \pm j_{cutoff}} G^{eff}_{i-i',j-j'} \rho_{i',j'}$$

- If grid length in z is >> pipe radius, can truncate at nearest neighbors:

$$\phi_{i,j} = \sum_{i'=1}^{N_x} (G^{eff}_{i-i',j-1} \rho_{i',j-1} + G^{eff}_{i-i',j} \rho_{i',j} + G^{eff}_{i-i',j+1} \rho_{i',j+1})$$

- For a rectangular pipe, can rewrite Green function as a sum of convolutions and correlations; can still use FFT-based approach to sum over elements
- Applicability to circular pipes is still an open problem

Conclusion

- **Under SciDAC we have developed a multi-physics 3D parallel framework**
  - Space charge
  - Nonlinear optics
  - Wakes (SciDAC/BNL, tested in Synergia)
  - Improved algorithms
  - Test suite
- **Available for use through A. Dragt and R. Ryne**
- **Questions: rdryne@lbl.gov**