Particle Pushing with High-Order Elements

A. H. Glasser, PSI Center, University of Washington
V. S. Lukin, Naval Research Laboratory

Presented at the CEMM Meeting
Madison, Wisconsin, June 12-13, 2012

Key Concepts

- Add particles to HiFi. First step: particle tracing only, preliminary to closure.
- Full-orbit equations for now, guiding center equations later.
- Electromagnetic potentials (**A**, ϕ) and Cartesian coordinates **x** expressed as high-order spectral elements in logical coordinates **q**.
- Hamilton's equations of motion in logical coordinates:
  - Exploit full high-order representation.
  - Avoid mapping between logical and physical coordinates.
  - Quasi-continuous representation of metric tensor. No stopping at grid cell boundaries.
- High-order implicit and symplectic integrators, assuming implicit PIC formulation, *c.f.* Chen, Chacón, and Barnes, no particle CFL condition, separate time steps for electrons, ions, fluid.
- Compare speed, accuracy, conservation properties for different methods.

Lagrangian Formulation

Cartesian and Logical Coordinates

$$x_i = x_i(q_j), \quad \dot{x}_i = J_{ij}\dot{q}_j, \quad J_{ij} \equiv \frac{\partial x_i}{\partial q_j}$$

Lagrangian

$$\begin{aligned} L \equiv& \frac{1}{2}m\dot{x}_i\dot{x}_i + eA_i\dot{x}_i - e\varphi \\ =& \frac{1}{2}m(J_{ij}\dot{q}_j)(J_{ik}\dot{q}_k) + eA_iJ_{ij}\dot{q}_j - e\varphi \\ =& \frac{1}{2}mg_{ij}\dot{q}_i\dot{q}_j + e\bar{A}_i\dot{q}_i - e\varphi \end{aligned}$$

Metric Tensor and Logical Components of A

$$g_{jk} \equiv J_{ij}J_{ik}, \quad g_{ij}g_{jk}^{-1} \equiv \delta_{ik}, \quad \bar{A}_i \equiv A_jJ_{ji}$$

Conjugate Momenta

$$p_i \equiv \frac{\partial L}{\partial \dot{q}_i} = mg_{ij}\dot{q}_j + e\bar{A}_i$$

Lagrange's Equations of Motion

$$\dot{q}_i = \frac{1}{m}g_{ij}^{-1}\left(p_j - e\bar{A}_j\right), \quad \dot{p}_i = \frac{\partial L}{\partial q_i} = e\frac{\partial \bar{A}_j}{\partial q_i}\dot{q}_j - e\frac{\partial \varphi}{\partial q_i} + \frac{m}{2}\frac{\partial g_{jk}}{\partial q_i}\dot{q}_j\dot{q}_k$$

Hamiltonian Formulation

Legendre Transformation

$$
\begin{aligned}
H =& \dot{q}_i p_i - L \\
=& \frac{1}{m} p_i g_{ij}^{-1} \left(p_j - e\bar{A}_j\right) - \frac{1}{2m} g_{ij}^{-1} \left(p_i - e\bar{A}_i\right) \left(p_j - e\bar{A}_j\right) \\
& - \frac{e}{m} \bar{A}_i g_{ij}^{-1} \left(p_j - e\bar{A}_j\right) + e\varphi \\
=& \frac{1}{2m} g_{ij}^{-1} \left(p_i - e\bar{A}_i\right) \left(p_j - e\bar{A}_j\right) + e\varphi
\end{aligned}
$$

Hamilton's Equations of Motion

$$
\dot{q}_i = \frac{\partial H}{\partial p_i} = \frac{1}{m} g_{ij}^{-1} \left(p_j - e\bar{A}_j\right)
$$

$$
\dot{p}_i = -\frac{\partial H}{\partial q_i} = e\frac{\partial \bar{A}_j}{\partial q_i}\dot{q}_j - e\frac{\partial \varphi}{\partial q_i} + \frac{m}{2}\frac{\partial g_{jk}}{\partial q_i}\dot{q}_j\dot{q}_k
$$

Example: Cylindrical Coordinates

$$\mathbf{q} = \begin{pmatrix} r \\ \theta \\ z \end{pmatrix}, \quad \mathbf{x}(\mathbf{q}) = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r\cos\theta \\ r\sin\theta \\ z \end{pmatrix}$$

$$\mathbf{J} \equiv \frac{\partial \mathbf{x}}{\partial \mathbf{q}} = \begin{pmatrix} \cos\theta & -r\sin\theta & 0 \\ \sin\theta & r\cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{g} \equiv \mathbf{J}^T\mathbf{J} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & r^2 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{g}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/r^2 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{A} = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix}$$

$$\bar{\mathbf{A}} \equiv \mathbf{A} \cdot \mathbf{J} = \begin{pmatrix} A_x\cos\theta + A_y\sin\theta \\ rA_y\cos\theta - rA_x\sin\theta \\ A_z \end{pmatrix} = \begin{pmatrix} A_r \\ rA_\theta \\ A_z \end{pmatrix}$$

$$H = \frac{1}{2m}\left[(p_r - eA_r)^2 + (p_\theta - erA_\theta)^2/r^2 + (p_z - eA_z)^2\right] + e\varphi$$

Field Specification

```
TYPE :: field_type
      REAL(r8) :: phi
      REAL(r8), DIMENSION(3) :: gradphi,a
      REAL(r8), DIMENSION(3,3) :: grada,ginv
      REAL(r8), DIMENSION(3,3,3) :: gmat1
END TYPE field_type
```

```
SUBROUTINE field_eval(t,q,field)

     REAL(r8), INTENT(IN) :: t
     REAL(r8), DIMENSION(3), INTENT(IN) :: q
     TYPE(field_type), INTENT(OUT) :: field
```

Interface could be used for any method of discretization, e.g. NIMROD, M3D-C1

ODE Solvers

Reference

E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd Ed., Springer, 2006.

Ordinary Differential Equation

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(t_0) = \mathbf{y}_0$$

Runge-Kutta Methods

$$\mathbf{k}_i = \mathbf{f}\left(t_0 + c_i h, \mathbf{y}_0 + h \sum_{j=1}^{s} a_{ij} \mathbf{k}_j\right), \quad i = 1, \ldots, s$$

$$c_i = \sum_{j=1}^{s} a_{ij}, \quad \mathbf{y}_1 = \mathbf{y}_0 + h \sum_{i=1}^{s} b_i \mathbf{k}_i$$

- Specific method specified by s, **a, b, c.**
- Key property: order of accuracy.
- Special properties: explicit, implicit, diagonally implicit, symplectic.
- Implicit methods require Picard iteration.

Examples of Runge-Kutta Methods

Fourth-Order Explicit Methods, $s=4$

$$\mathbf{a} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1/6 \\ 2/6 \\ 2/6 \\ 1/6 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 0 \\ 1/2 \\ 1/2 \\ 1 \end{pmatrix}$$

$$\mathbf{a} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 \\ -1/3 & 1 & 0 & 0 \\ 1 & -1 & 1 & 0 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1/8 \\ 3/8 \\ 3/8 \\ 1/8 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 0 \\ 1/3 \\ 2/3 \\ 1 \end{pmatrix}$$

2nd-Order Implicit Trapezoidal Method, $s=2$

$$\mathbf{a} = \begin{pmatrix} 0 & 0 \\ 1/2 & 1/2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- Symplectic: preserves discretized phase space volume,
- Energy error is oscillatory, not secularly growing. Amplitude of oscillation depends on order and step size.
- Implicit: requires Picard iteration

Gauss Collocation Methods

4th-Order Gauss Collocation Method, $s=2$

$$\mathbf{a}=\begin{pmatrix}1/4 & 1/4-\sqrt{3}/6\\ 1/4+\sqrt{3}/6 & 1/4\end{pmatrix},\quad \mathbf{b}=\begin{pmatrix}1/2\\ 1/2\end{pmatrix},\quad \mathbf{c}=\begin{pmatrix}1/2-\sqrt{3}/6\\ 1/2+\sqrt{3}/6\end{pmatrix}$$

6th-Order Gauss Collocation Method, $s=3$

$$\mathbf{a}=\begin{pmatrix}5/36 & 2/9-\sqrt{15}/15 & 5/36-\sqrt{15}/30\\ 5/36+\sqrt{15}/24 & 2/9 & 5/36-\sqrt{15}/24\\ 5/36+\sqrt{15}/30 & 2/9+\sqrt{15}/15 & 5/36\end{pmatrix},\quad \mathbf{b}=\begin{pmatrix}1/18\\ 4/9\\ 5/18\end{pmatrix},\quad \mathbf{c}=\begin{pmatrix}1/2-\sqrt{15}/10\\ 1/2\\ 1/2+\sqrt{15}/10\end{pmatrix}$$

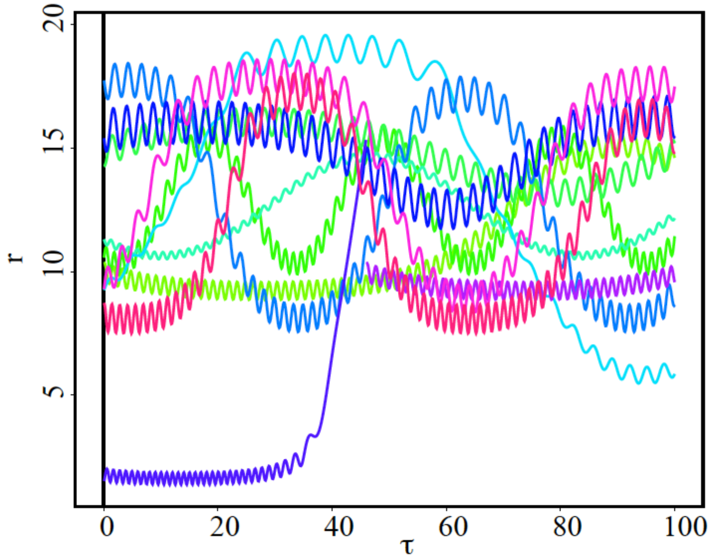
- Symplectic: preserves discretized phase space volume,
- Energy error is oscillatory, not secularly growing.
- Implicit: requires Picard iteration
- Order of accuracy is twice the number of function evaluations.

The PUSH Code

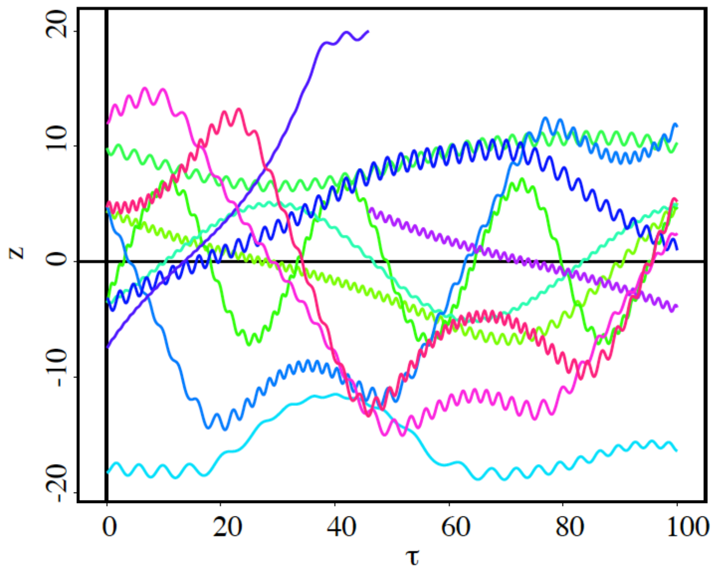
- **Fields**
  - Analytical FRC with vacuum RMF
  - Analytical cylindrical spheromak
  - HiFi fields, arbitrary nx, ny, nz, np
    Horner's method for fast polynomial evaluation
- **ODE solvers**
  - Explicit: LSODE, RK4
  - 2nd order implicit, symplectic: Crank-Nicholson, Midpoint
  - Higher-order implicit: 4th and 6th order Gauss Collocation
- **Initial conditions, n particles**
  - Maxwellian velocity distribution
  - Random initial positions
- **Diagnostics**
  - XDRAW
  - VisIt

XDRAW Graphics, Spheromak

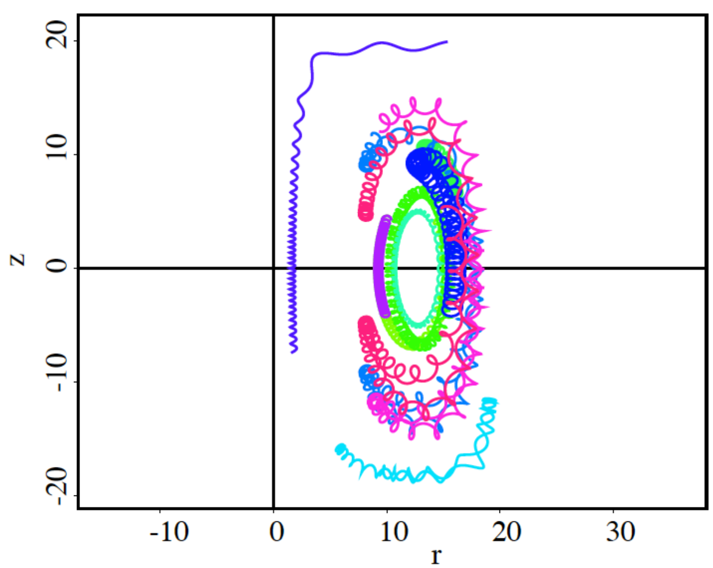
Radial Position vs. Scaled Time



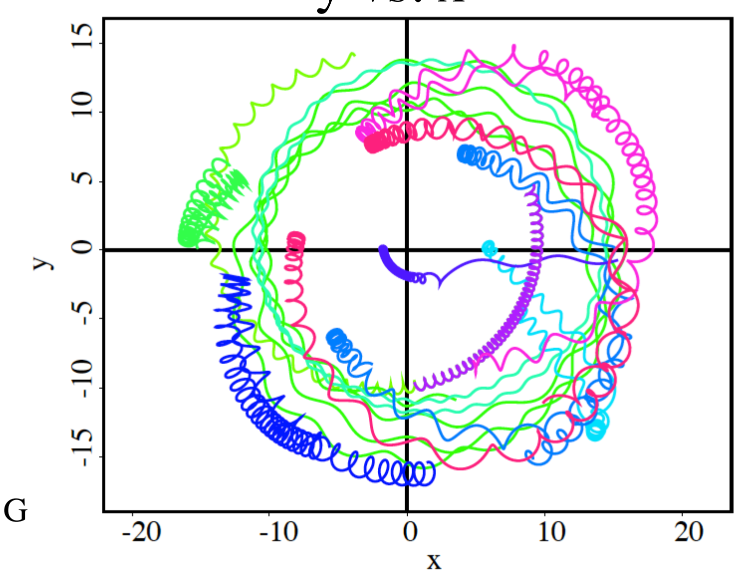
Axial Position vs. Scaled Time



z vs. r

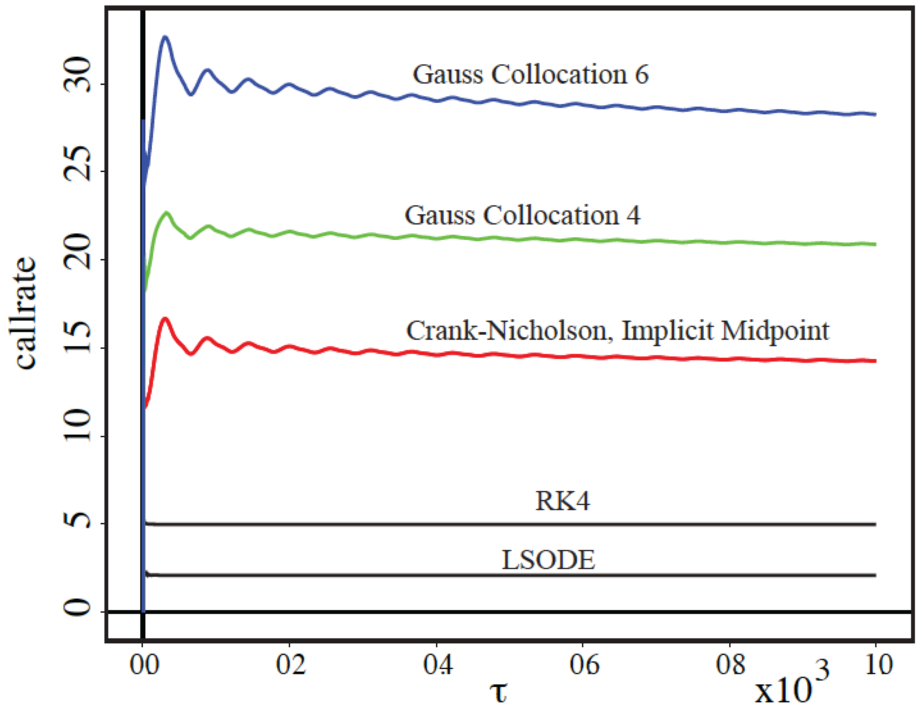


y vs. x

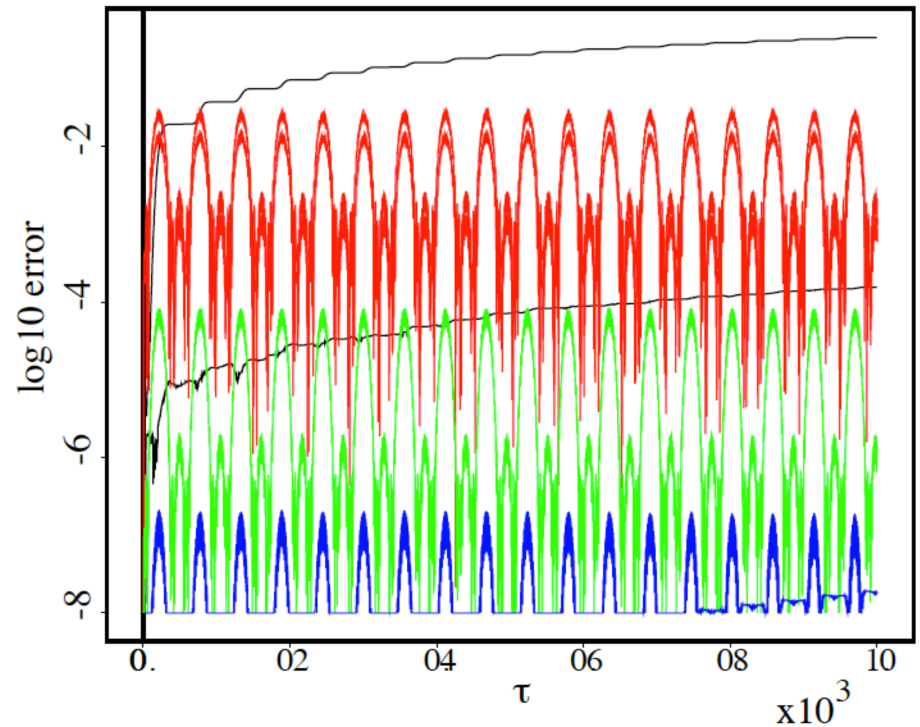


Speed and Accuracy

Function Calls per Time Step



Relative Error vs. Scaled Time



- Explicit methods are cheap, but error grows secularly.
- Implicit, symplectic methods are expensive, but error oscillates, bounded by step size and order
- How much effort is justified by improved error control?

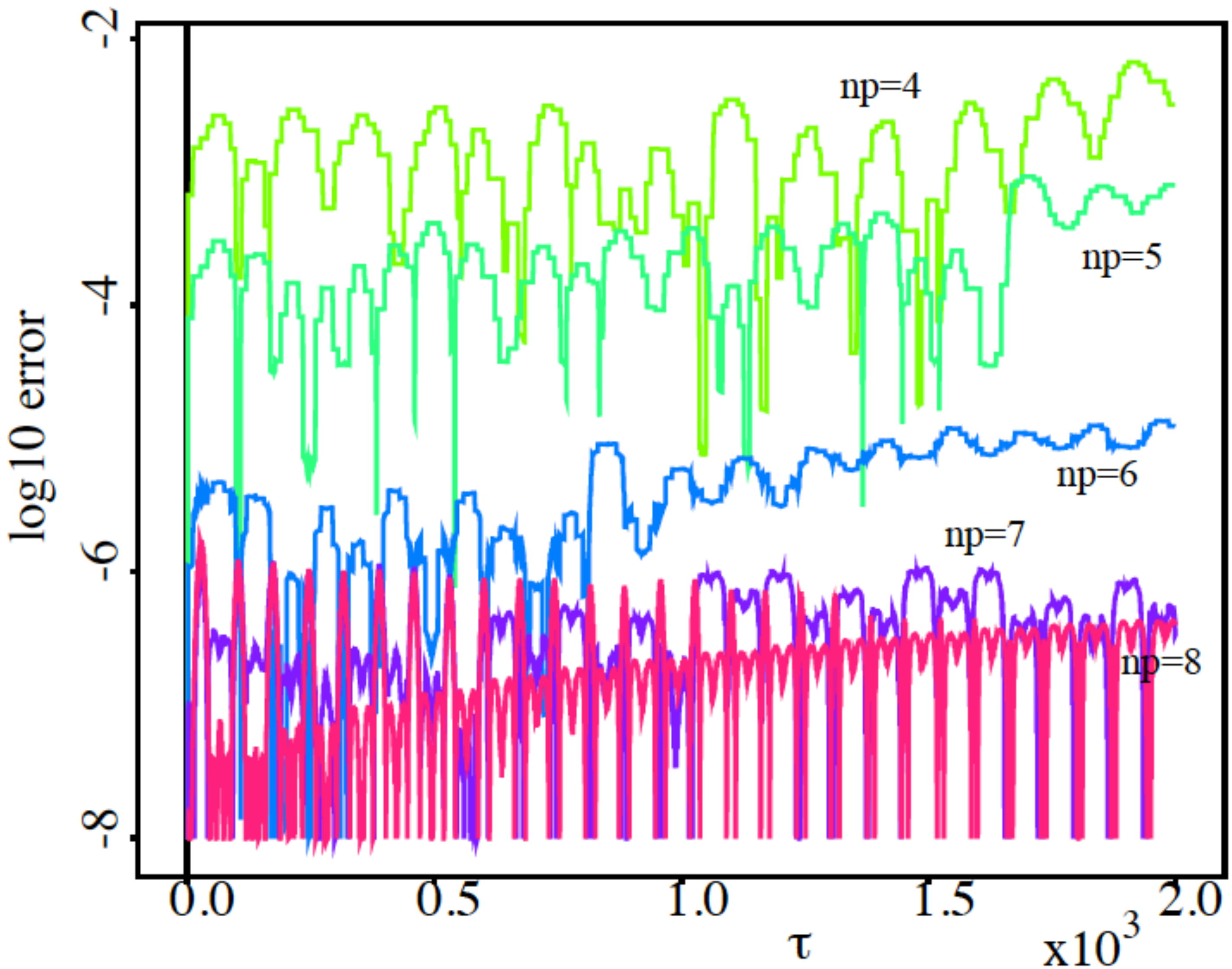
Catch-22

- Implicit, symplectic methods have higher order but require more function evaluations per time step.
- Can we win by exploiting higher order to take larger time steps?
- Catch-22: failure of Picard iteration.
- This is for full orbits. It may be possible to beat this for guiding center orbits. That remains to be seen.

HiFi Fields

- Horner’s method is used to optimize polynomial evaluation after input spectral element amplitudes are converted to polynomial coefficients.
- High-order elements require more cpu time for polynomial evaluation.
- Low-order elements cause deterioration of energy conservation because of discontinuities in metric tensor.
- High degree polynomial input can be truncated.

Effect of np on Error, GC4



Summary and Future Directions

- The PUSH code is designed to allow exploration of different methods for optimizing particle pushing in analytical and numerical fields.
- Hamiltonian formulation in logical coordinates allows full use of high-order representation and avoids the need to transform between logical and physical coordinates.
- Flexible field specification allows easy interface for any kind of spatial discretization.
- Advanced ODE solvers have been implemented and tested:
  implicit, high-order, symplectic.
- The results show that advanced methods can improve error control at a price. It is not yet clear when that price is worth paying.
- Implementation of Hamiltonian guiding-center equations is straightforward.
- Future directions: efficient parallelization, closure.

Plasma Science and Innovation PSI-Center

NAVAL RESEARCH LABORATORY WASHINGTON, DC