Predictive Community Computational Tools for Virtual Plasma Science Experiments

J.-L. Vay, E. Esarey, S. Bulanov, A. Koniges

Lawrence Berkeley National Laboratory

J. Barnard, A. Friedman, D. Grote

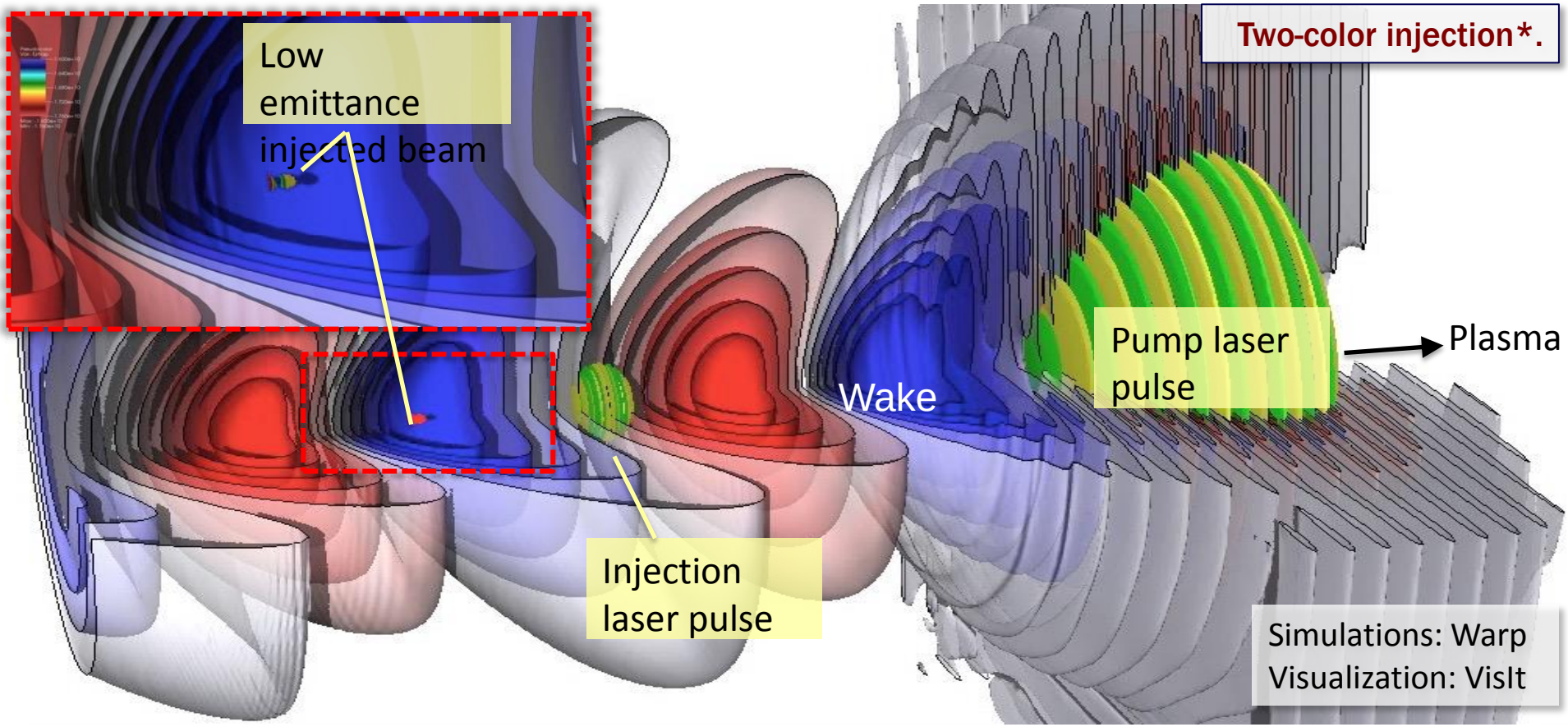
Lawrence Livermore National Laboratory

Frontiers of Plasma Science Workshops - Town Hall

June 30-July 1, 2015

Advanced simulations play an increasingly important role in the exploration of new concepts.

Large scale 3-D simulations with Warp validated a new concept of injection of ultra-low emittance beam in laser-plasma accelerator

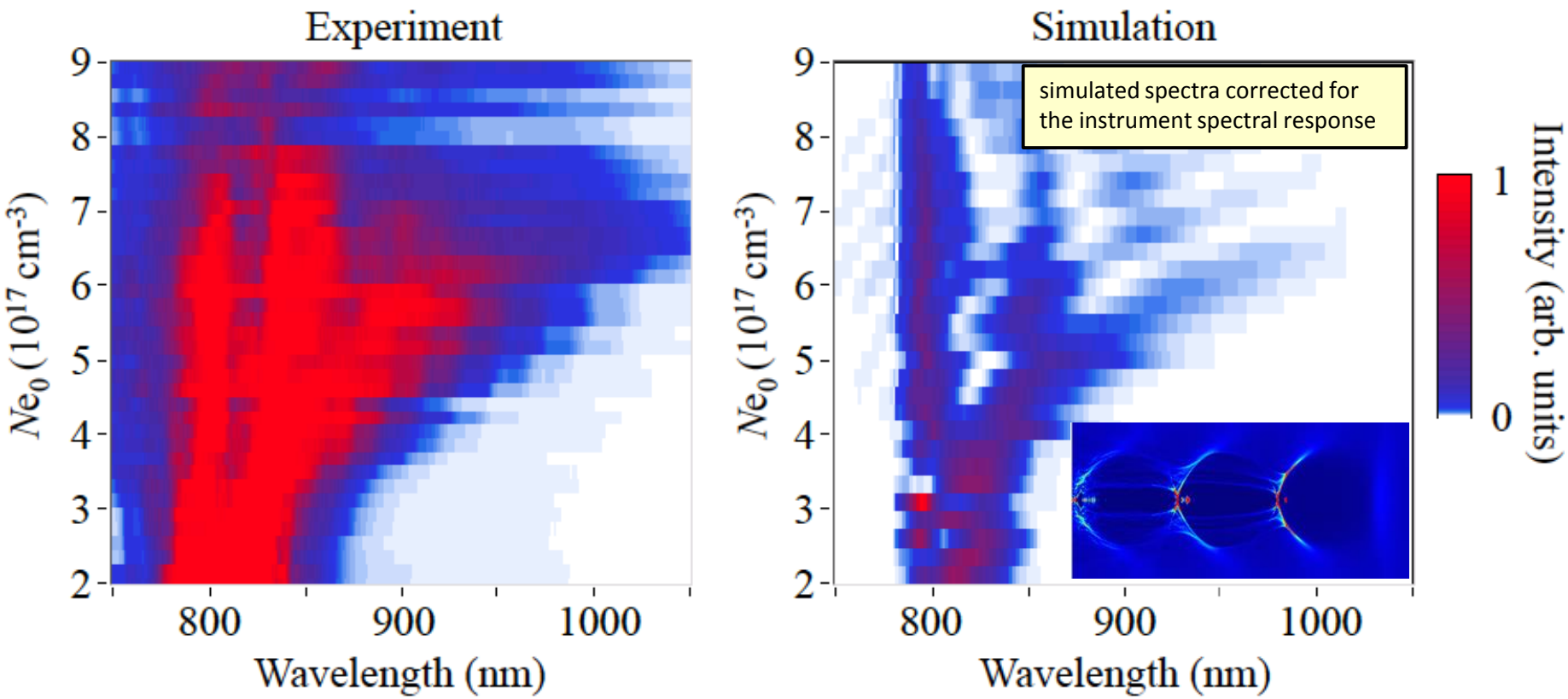


*L.-L. Yu, et al, *Phys. Rev. Lett.* **112**, 125001 (2014)

Advanced simulations play an increasingly important role in the design, operation and analysis of experiments.

Large scale simulations with INF&RNO* supported world record BELLA 4.25 GeV beam over 9 cm**

Laser optical spectra after 9 cm plasma (U_{laser}=7.5 J)



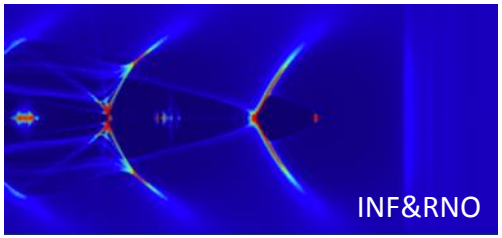
*2015 NERSC HPC award

**W. P. Leemans, et al., *Phys. Rev. Lett.* 113, 245002 (2014)

Large scale simulations (still) take too long! Real machines need fast turnaround for real-time tuning.

Now

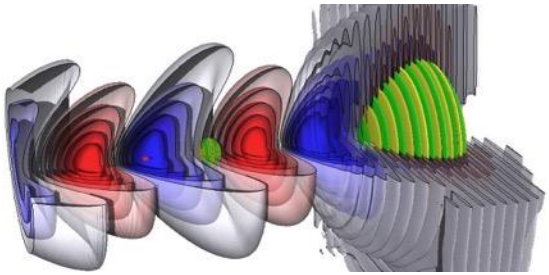
2D-RZ simulations of BELLA



0.5 M CPU-HRS (~1 week for 9 cm plasma)

Geometry	R, Z-CT
# grid cells	1,400x14,000
# grid cells/kp	15x200
L_{plasma}	~9 cm
# time steps	12,000,000
# cores	3,400
Runtime	150 hours
Cost	0.5 M CPU-HRs

3D simulation of novel injection scheme

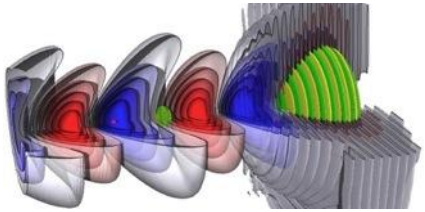


0.25 M CPU-HRS (~1.5 day for 1 mm plasma)

Geometry	X, Z-CT	3-D X, Y, Z-CT
# grid cells	1,200x3,300	600x600x1,650
# grid cells/kp	65x140	32x32x70
L_{plasma}	~1 mm	
# time steps	300,000	150,000
# cores	1,200	6,144
Runtime	1 hours	36 hours
Cost	1.2 k CPU-HRs	0.22 M CPU-HRs

Future computational needs are extremely high

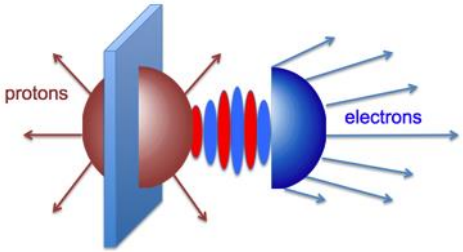
- **Laser plasma acceleration**
  - 3-D BELLA simulation: ~ 500 M CPU-HRs
  - 3-D k-BELLA simulation: ~ 50 M CPU-HRs
  - Parametric studies: × 100



➔ **10-100 Billions of CPU-HRs**

- **Ion acceleration (BELLA-i)**
  - $\Delta x \approx \lambda/10^3$ to resolve λ solid density plasma

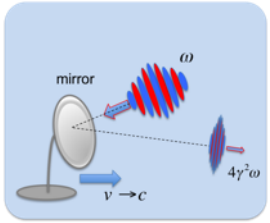
  ➔ small box of $(10\lambda)^3$ results in $\approx 10^{12}$ cells
  - Time scales: from fs (PIC) to ns (MHD)



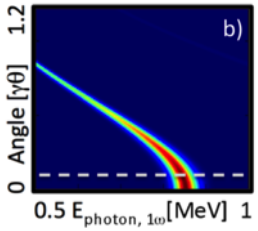
➔ **100M to Billions of CPU-HRs/run**

- **Others also stressing computational needs**

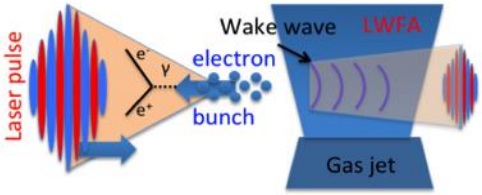
Flying plasma mirrors



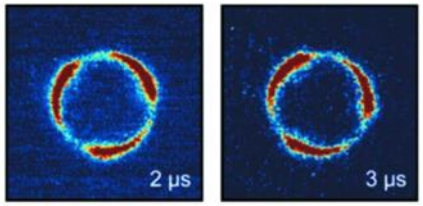
Laser-driven X-ray sources



High-intensity laser-matter interactions



Self-organization in ExB discharges



Confluence of exascale computers & novel algorithms to deliver better, bigger & faster computing

Goal is multi-scale multi-physics:

- Combine modules/codes (MHD+PIC+target physics+…)

➔ Virtual experimentation

Runtime

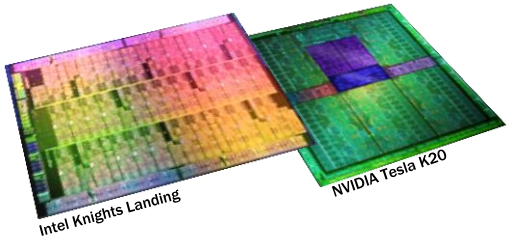
<1h

Experiment "real-time" feedback

Enabler 1 - new hardware (Intel Phi/GPU):

- Optimize algorithms that scale to 100s of million cores

➔ Speedup ×10-100/ larger ×10-1000



Enabler 2 - improved algorithms:

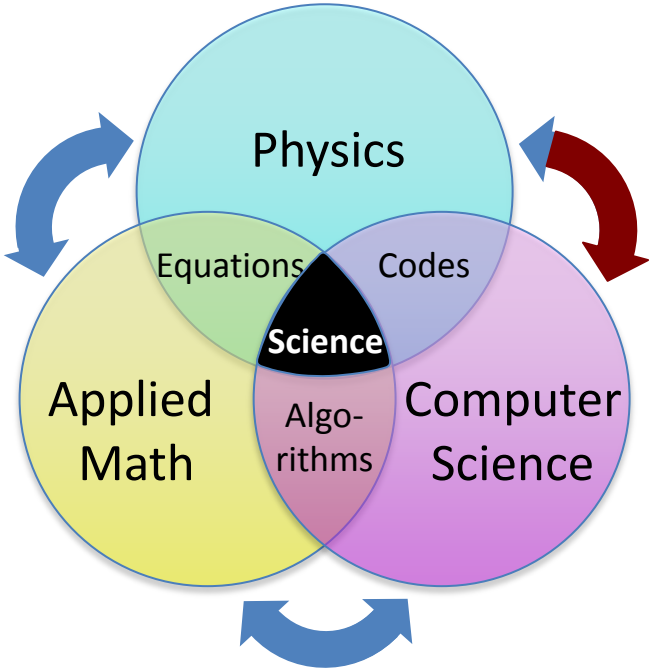
- Spectral solvers + AMR + Ohm + boosted frame + …

➔ Speedup $\times 10^2$-10^8

Days

Task is complex & calls for cross-discipline teams

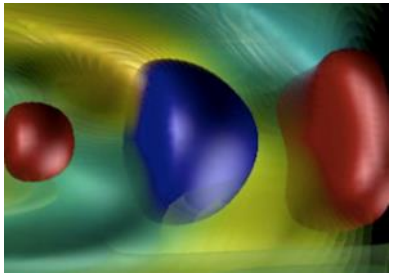
Need for tight coupling, as process not unidirectional:



w/ physics used to alter algorithms & codes.

Examples

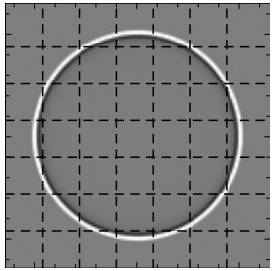
Lorentz boosted frame



Uses special relativity to speedup simulations by orders of magnitude.

J.-L. Vay, *Phys. Rev. Lett.* **98**, 130405 (2007)

Parallel spectral solver



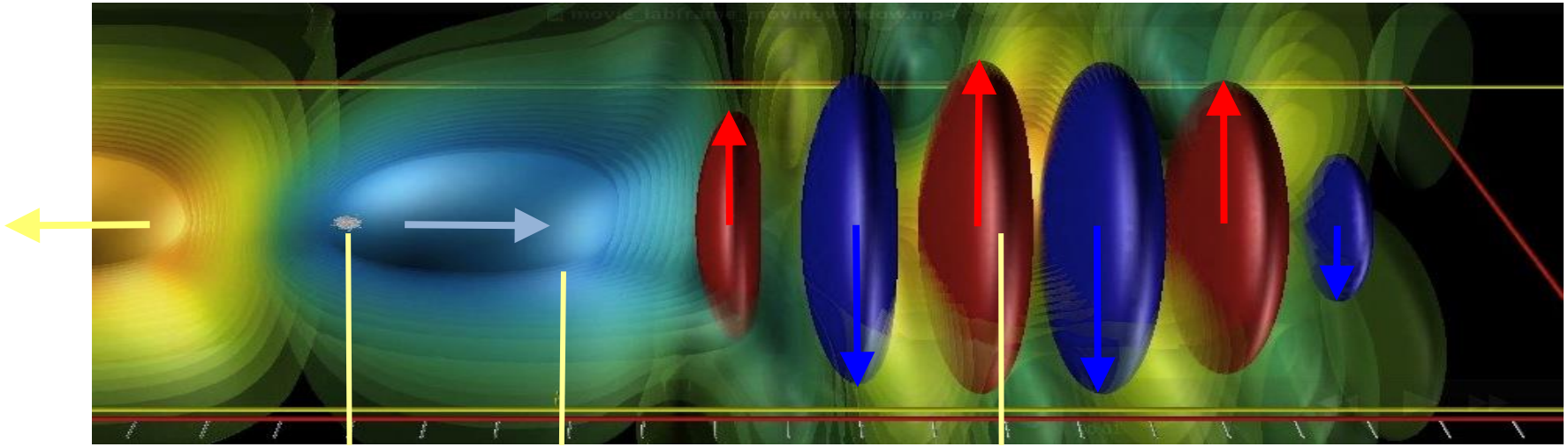
Uses finite speed of light to enable direct scaling to many cores.

J.-L. Vay, I. Haber & B. B. Godfrey, *J. Comput. Phys.* **243**, 260-268 (2013)

Algorithmic developments can lead to very high payoffs

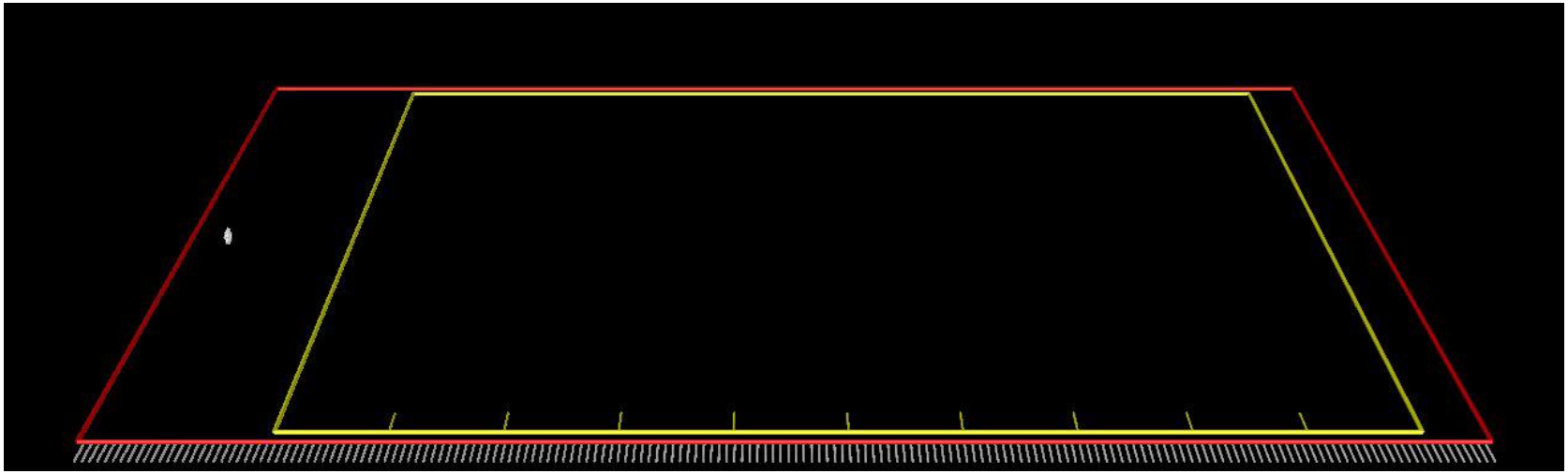
Laser plasma acceleration (LPA) analogy

surfer wake boat



e- beam wake laser

Modeling from first principle is very challenging



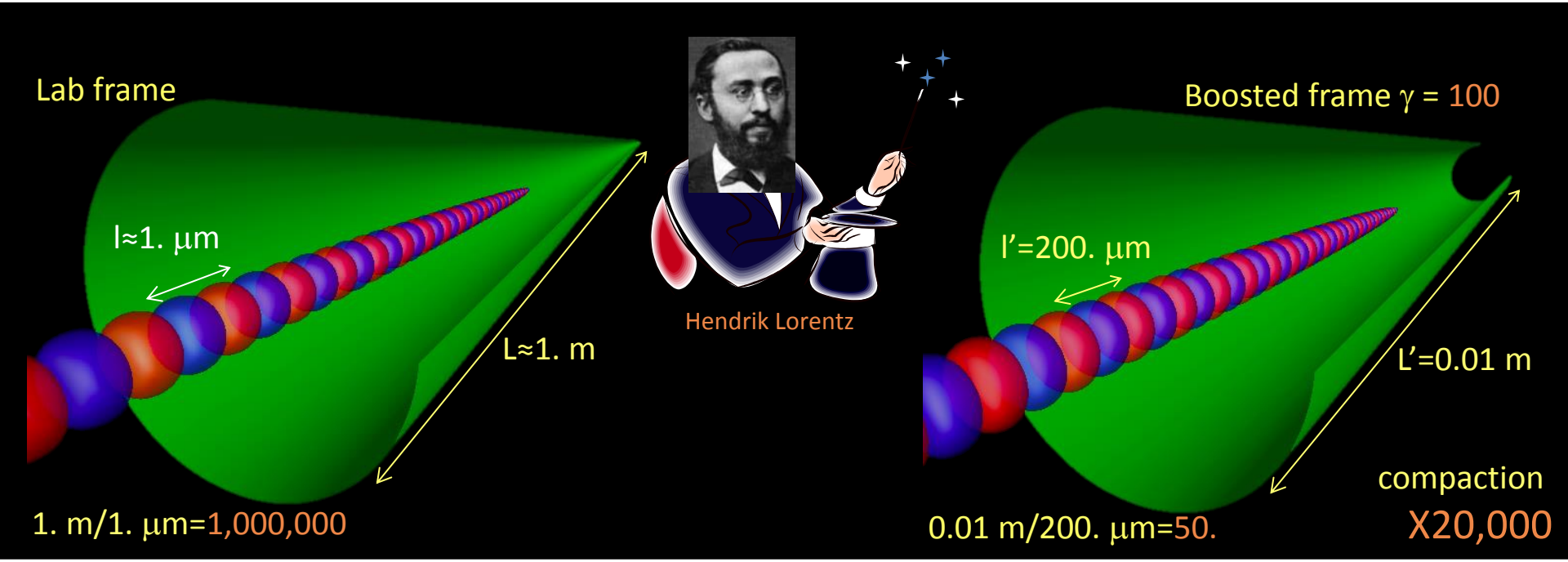
For a 10 GeV scale stage:

~1μm wavelength laser propagates into ~1m plasma

➔ millions of time steps needed

(similar to modeling 5m boat crossing ~5000 km Atlantic Ocean)

Solution: model in frame moving near the speed of light*

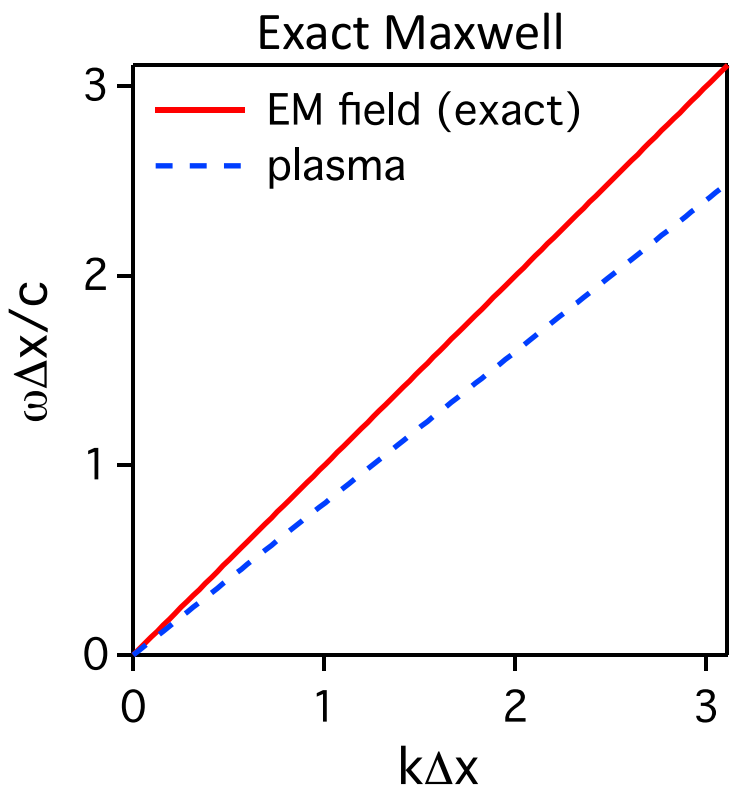


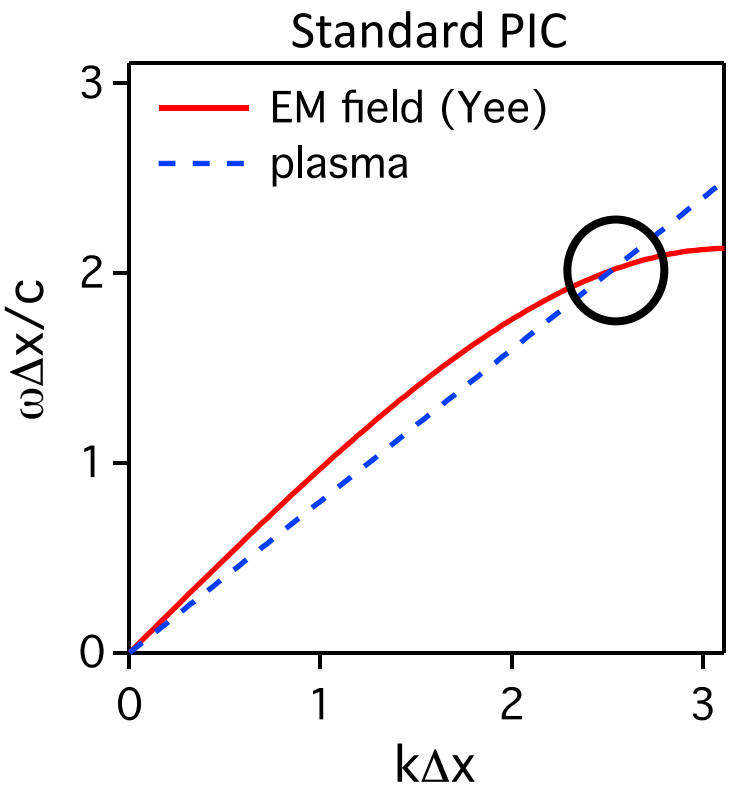
- BELLA-scale w/ ~ 5k CPU-Hrs: 2006 - 1D run ➔ 2011: 3D run
- Progress in envelope solvers (Benedetti) can be combined for further speedups

*J.-L. Vay, *Phys. Rev. Lett.* **98**, 130405 (2007)

Relativistic plasmas PIC subject to "numerical Cherenkov"

Numerical dispersion leads to crossing of EM field and plasma modes -> instability.

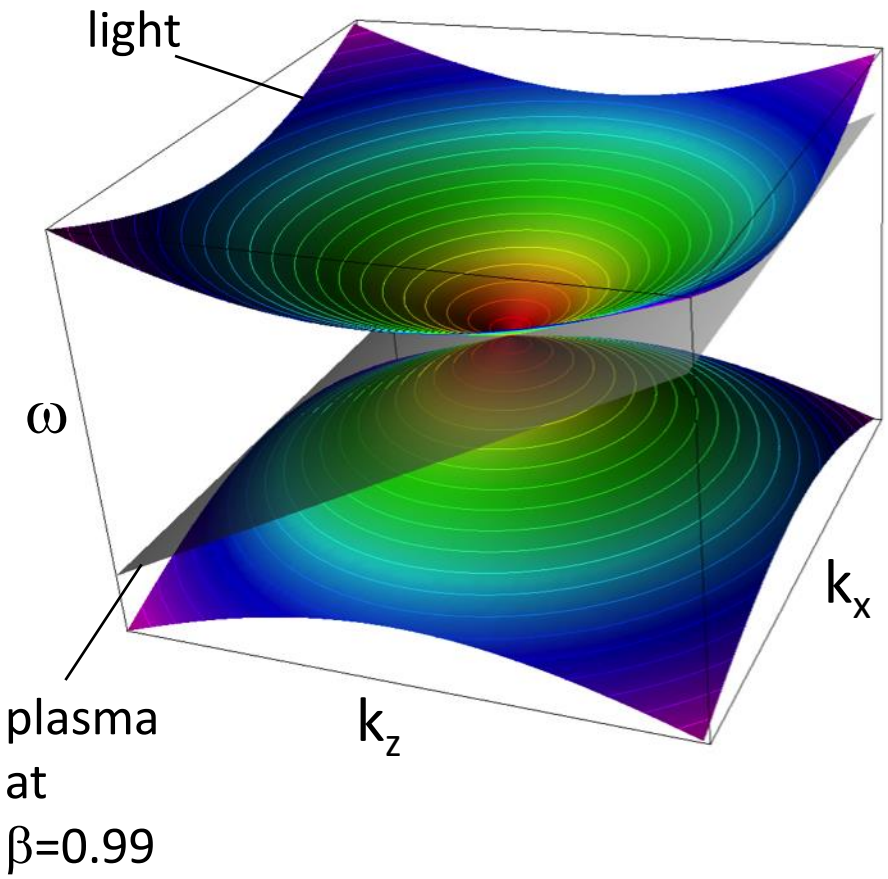




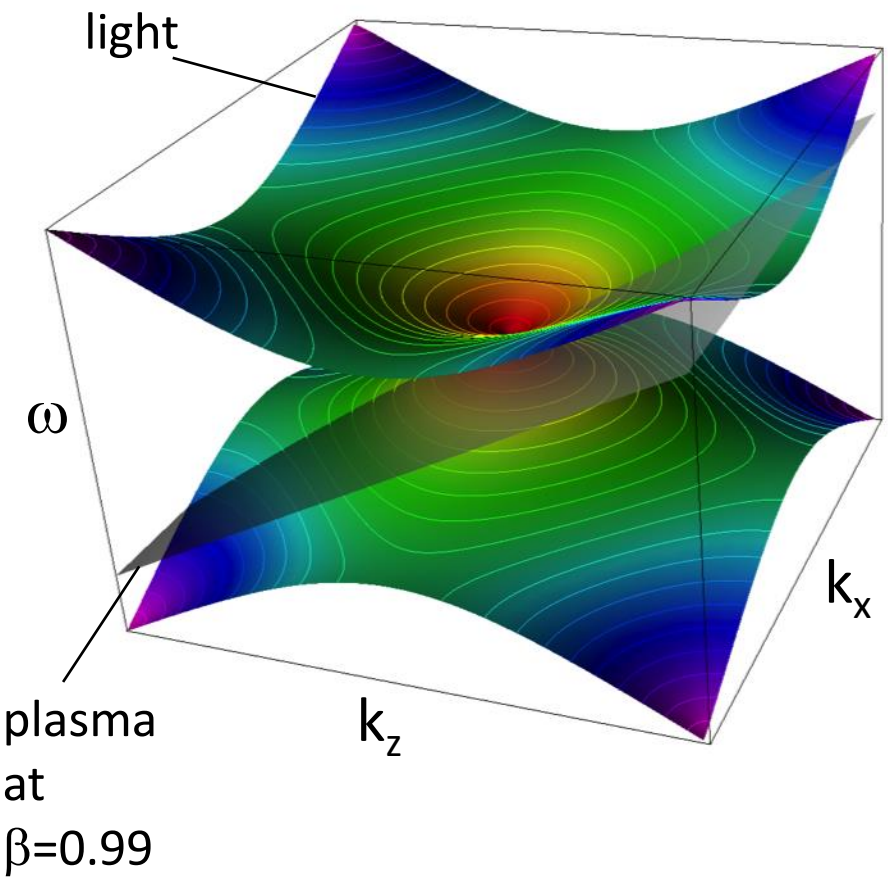
*B. B. Godfrey, "Numerical Cherenkov instabilities in electromagnetic particle codes", *J. Comput. Phys.* **15** (1974)

Space/time discretization aliases → more crossings in 2/3-D

Exact Maxwell

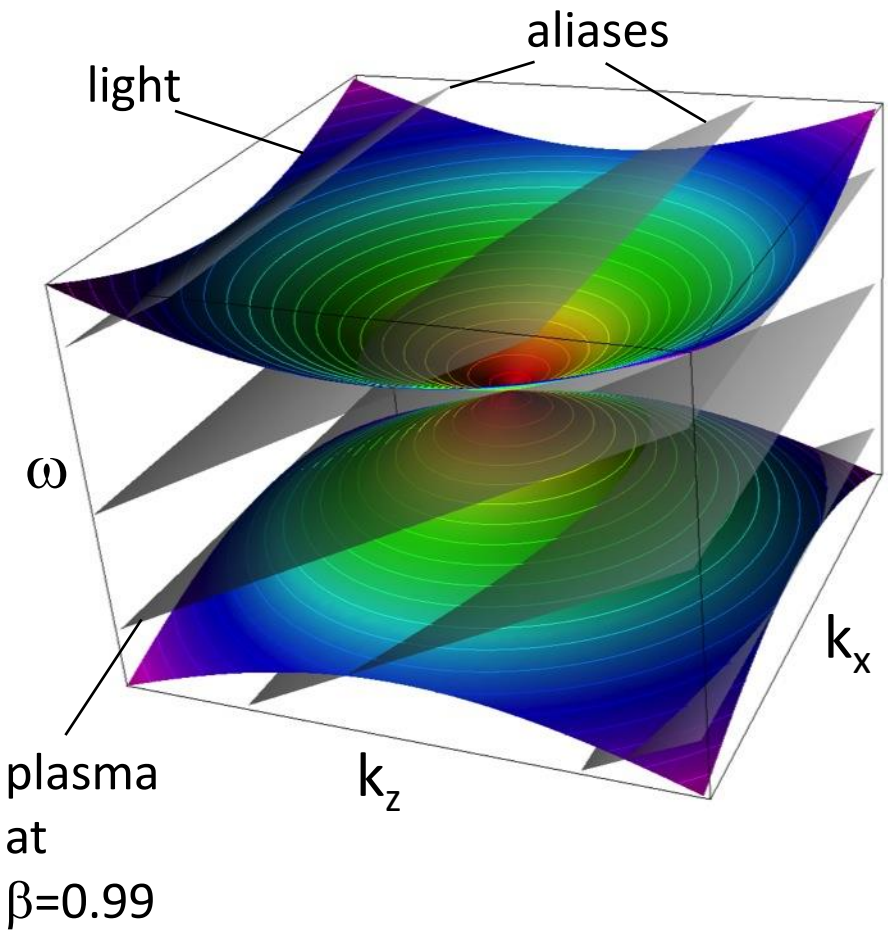


Standard PIC

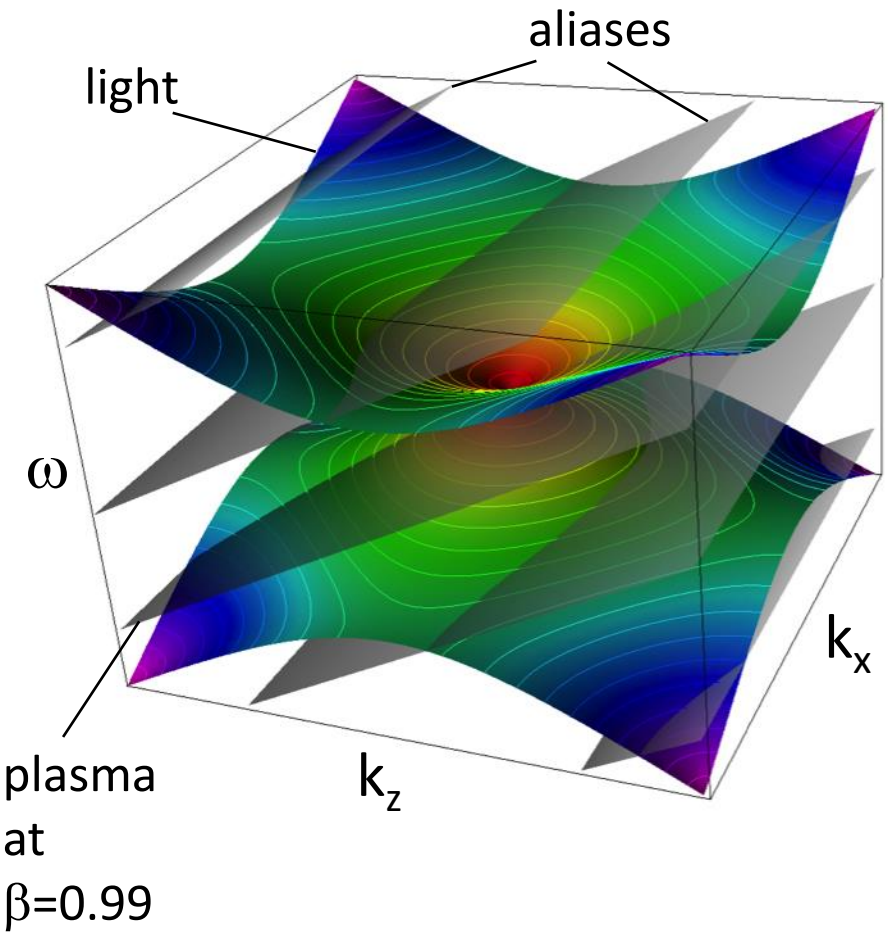


Space/time discretization aliases → more crossings in 2/3-D

Exact Maxwell



Standard PIC



Analysis calls for full PIC numerical dispersion relation

Numerical dispersion relation of full-PIC algorithm*

2-D relation (Fourier space):

$$\begin{pmatrix} \xi_{z,z} + [\omega] & \xi_{z,x} & \xi_{z,y} + [k_x] \\ \xi_{x,z} & \xi_{x,x} + [\omega] & \xi_{x,y} - [k_z] \\ [k_x] & -[k_z] & [\omega] \end{pmatrix} \begin{pmatrix} E_z \\ E_x \\ B_y \end{pmatrix} = 0.$$

$$[\omega] = \sin\left(\omega\frac{\Delta t}{2}\right) / \left(\frac{\Delta t}{2}\right) \quad [k_z] = k_z \sin\left(k\frac{\Delta t}{2}\right) / \left(k\frac{\Delta t}{2}\right) \quad [k_x] = k_x \sin\left(k\frac{\Delta t}{2}\right) / \left(k\frac{\Delta t}{2}\right)$$

$$S^J = \left[\sin\left(k_z'\frac{\Delta z}{2}\right) \Big/ \left(k_z'\frac{\Delta z}{2}\right)\right]^{\ell_z+1} \left[\sin\left(k_x'\frac{\Delta x}{2}\right) \Big/ \left(k_x'\frac{\Delta x}{2}\right)\right]^{\ell_x+1},$$

$$S^{E_z} = \left[\sin\left(k_z'\frac{\Delta z}{2}\right) \Big/ \left(k_z'\frac{\Delta z}{2}\right)\right]^{\ell_z} \left[\sin\left(k_x'\frac{\Delta x}{2}\right) \Big/ \left(k_x'\frac{\Delta x}{2}\right)\right]^{\ell_x+1} (-1)^{m_z},$$

$$S^{E_x} = \left[\sin\left(k_z'\frac{\Delta z}{2}\right) \Big/ \left(k_z'\frac{\Delta z}{2}\right)\right]^{\ell_z+1} \left[\sin\left(k_x'\frac{\Delta x}{2}\right) \Big/ \left(k_x'\frac{\Delta x}{2}\right)\right]^{\ell_x} (-1)^{m_x},$$

$$S^{B_y} = \cos\left(\omega\frac{\Delta t}{2}\right) \left[\sin\left(k_z'\frac{\Delta z}{2}\right) \Big/ \left(k_z'\frac{\Delta z}{2}\right)\right]^{\ell_z} \left[\sin\left(k_x'\frac{\Delta x}{2}\right) \Big/ \left(k_x'\frac{\Delta x}{2}\right)\right]^{\ell_x} (-1)^{m_z+m_x}.$$

*B. B. Godfrey, J. L. Vay, I. Haber, J. Comp. Phys. 248 (2013)

Numerical dispersion relation of full-PIC algorithm (II)

$$\xi_{z,z} = -n\gamma^{-2}\sum_m S^J S^{E_z} \csc^2\left[(\omega - k_z' v)\frac{\Delta t}{2}\right] \left(k k_z^2 \Delta t + \zeta_z k_x^2 \sin(k\Delta t)\right) \Delta t\,[\omega]\, k_z'/4k^3 k_z,$$

$$\xi_{z,x} = -n\sum_m S^J S^{E_x} \csc\left[(\omega - k_z' v)\frac{\Delta t}{2}\right] \eta_z k_x'/2k^3 k_z,$$

$$\xi_{z,y} = nv\sum_m S^J S^{B_y} \csc\left[(\omega - k_z' v)\frac{\Delta t}{2}\right] \eta_z k_x'/2k^3 k_z,$$

$$\xi_{x,z} = -n\gamma^{-2}\sum_m S^J S^{E_z} \csc^2\left[(\omega - k_z' v)\frac{\Delta t}{2}\right] \left(k\Delta t - \zeta_z \sin(k\Delta t)\right) \Delta t\,[\omega]\, k_x k_z'/4k^3,$$

$$\xi_{x,x} = -n\sum_m S^J S^{E_x} \csc\left[(\omega - k_z' v)\frac{\Delta t}{2}\right] \eta_x k_x'/2k^3 k_x,$$

$$\xi_{x,y} = nv\sum_m S^J S^{B_y} \csc\left[(\omega - k_z' v)\frac{\Delta t}{2}\right] \eta_x k_x'/2k^3 k_x,$$

*B. B. Godfrey, J. L. Vay, I. Haber, J. Comp. Phys. 248 (2013)

Numerical dispersion relation of full-PIC algorithm (III)

$$\eta_z = \cot\left[(\omega - k_z' v)\frac{\Delta t}{2}\right]\left(k k_z^2 \Delta t + \zeta_z k_x^2 \sin(k\Delta t)\right)\sin\left(k_z' v \frac{\Delta t}{2}\right) + \left(k\Delta t - \zeta_x \sin(k\Delta t)\right) k_z^2 \cos\left(k_z' v \frac{\Delta t}{2}\right),$$

$$\eta_x = \cot\left[(\omega - k_z' v)\frac{\Delta t}{2}\right]\left(k\Delta t - \zeta_z \sin(k\Delta t)\right) k_x^2 \sin\left(k_z' v \frac{\Delta t}{2}\right) + \left(k k_x^2 \Delta t + \zeta_x k_z^2 \sin(k\Delta t)\right)\cos\left(k_z' v \frac{\Delta t}{2}\right).$$

Then simplify and solve with Mathematica…

Tremendous progress on analysis/mitigation of NC in recent years

- **Analysis of Numerical Cherenkov has been generalized:**
  - **to finite-difference PIC codes ("Magical" time step explained):**
    - B. B. Godfrey and J.-L. Vay, J. Comp. Phys. 248 **(2013)** 33.
    - X. Xu, et. al., Comp. Phys. Comm., 184 **(2013)** 2503.
  - **to pseudo-spectral PIC codes:**
    - B. B. Godfrey, J. -L. Vay, I. Haber, J. Comp. Phys., 258 **(2014)** 689.
    - P. Yu et. al, J. Comp. Phys. 266 **(2014)** 124.

- **Suppression techniques were developed:**
  - **for finite-difference PIC codes:**
    - B. B. Godfrey and J.-L. Vay, J. Comp. Phys. 267 **(2014)** 1.
    - B. B. Godfrey and J.-L. Vay, Comp. Phys. Comm., **in press**
  - **for pseudo-spectral PIC codes:**
    - B. B. Godfrey, J.-L. Vay, I. Haber, IEEE Trans. Plas. Sci. 42 **(2014)** 1339.
    - P. Yu, et. al., arXiv:1407.0272 **(2014)**

Applications to relativistic laboratory and space plasmas.

A more coordinated program would boost the development of the next generation of simulations tools

Computational toolkits can rival (or surpass) the **complexity of experiments**

- needs to reproduce the **complexity of experimental setups**,
- discretization -and other numerical artifacts- add **layer of complexity** that needs to be **fully understood and controlled**.

A **coordinated, community program** would be **most effective**:

- develop & maintain **integrated comprehensive** kit is an **enormous task**:
  - collection of exascale-ready (PIC, MHD, fluid, radiation, QED, MD, EOS, etc.) modules,
- **current situation is suboptimal:** many codes within projects without coordination,
- **very few codes are open source:** inhibits co-development, verifiability and adoption.

Can **benefit from** links with/experience from **other communities**:

- **SciDAC programs**
- DOE-HEP Forum for Computational Excellence (**HEP-FCE**): coordinates across HEP
- Consortium for Advanced Modeling of Particle Accelerators (**CAMPA**)
  - coordinates code development between LBNL, SLAC and FNAL
- Astrophysics: e.g. **FLASH**

Conclusion

Plasma physics involves tightly coupled non-linear multi-scale/physics phenomena

- **large-scale modeling**, w/ theory/experiment, is **key to understanding of complexity**

Computational plasma physics is a **very active** field; better computational tools **accelerate discoveries** across plasma science

- **computational methods/codes are very general**
- benefits to society over a wide range of applications
  - **medical technology** (e.g. ion-driven cancer therapy, light sources for bioimaging),
  - **energy** (e.g. fusion sciences),
  - **basic physics** (e.g. HEDLP science, space plasmas science, charged particle traps),
  - **accelerators** (e.g. HEP and rare isotope facilities),
  - **industry** (e.g. plasma processing),
  - **national security** (e.g. science-based stockpile stewardship, gamma sources).

Exascale computing is key to enabling **"real-time" virtual experiments**

- but not silver bullet: **significant investments are needed** in algorithms/codes

A **coordinated, community effort** will be **most effective**

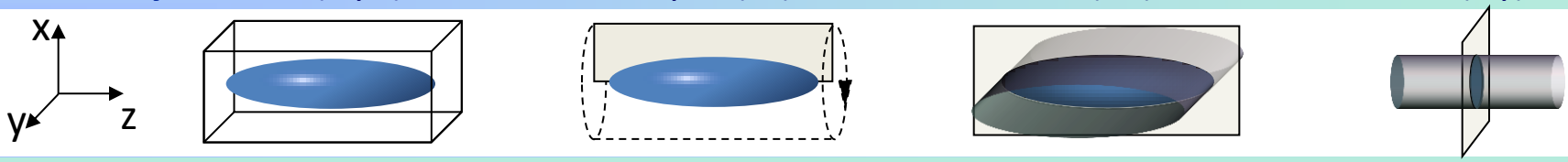
- **cross-institutional development** of integrated comprehensive kit **to maximize resources**

Extras

Warp: PIC framework for modeling of beams, plasmas & accel.

http://blast.lbl.gov/BLASTcodes_Warp.html; http://warp.lbl.gov

- **Geometry:** 3-D (x,y,z) | axisym. (r,z) | 2-D (x,z) | 2-D (x,y)

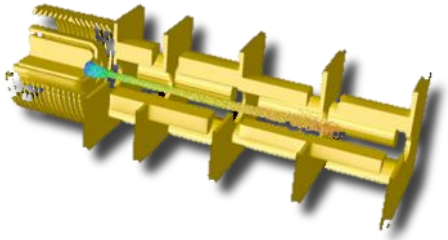


- **Reference frame:** **lab** | **moving-window** | **Lorentz boosted**

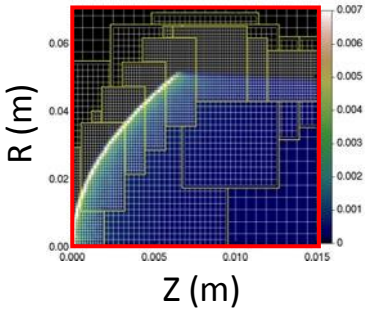
  z | z-vt | $\gamma(z-vt)$; $\gamma(t-vz/c^2)$

- **Field solvers**
  - electrostatic/magnetostatic - FFT, multigrid; AMR; implicit; cut-cell boundaries

Versatile conductor generator accommodates complicated structures



Automatic meshing around ion beam source emitter



  - Fully electromagnetic – Yee/node centered mesh, arbitrary order, spectral, PML, MR

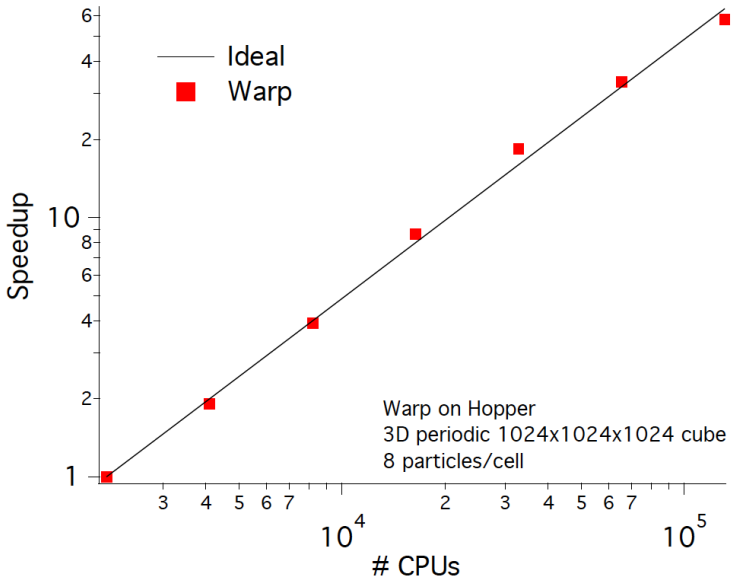
- **Accelerator lattice:** general; non-paraxial; can read MAD files
  - solenoids, dipoles, quads, sextupoles, linear maps, arbitrary fields, acceleration.

- **Particle emission & collisions**
  - particle emission: space charge limited, thermionic, hybrid, arbitrary,
  - secondary e- emission (Posinst), ion-impact electron emission (Txphysics) & gas emission,
  - Monte Carlo collisions: ionization, capture, charge exchange.

Warp is parallel, combining modern and efficient programming languages

- **Parallellization:** MPI (1, 2 and 3D domain decomposition)

Parallel strong scaling of Warp 3D PIC-EM solver on Hopper supercomputer (NERSC)



- **Python and FORTRAN*:** "steerable," input decks are programs

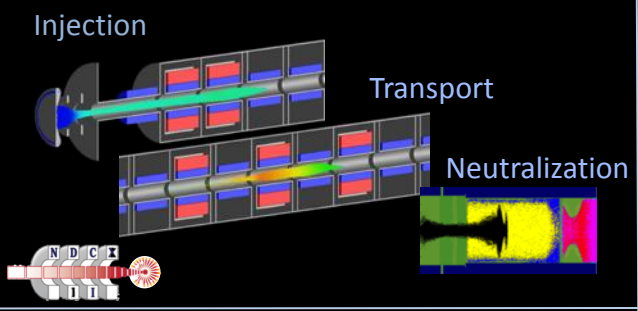
```
From warp import *
…
nx = ny = nz = 32
dt = 0.5*dz/vbeam
…
initialize()
step(zmax/(dt*vbeam))
…
```

← Imports Warp modules and routines in memory
← Sets # of grid cells
← Sets time step
← Initializes internal FORTRAN arrays
← Pushes particles for N time steps with FORTRAN routines

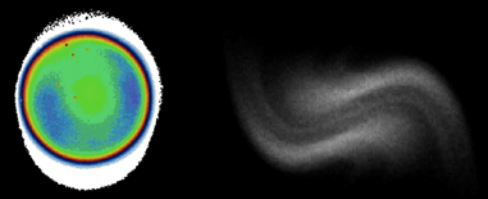
*http://hifweb.lbl.gov/Forthon – dpgrote@lbl.gov

Sample applications

Space charge dominated beams

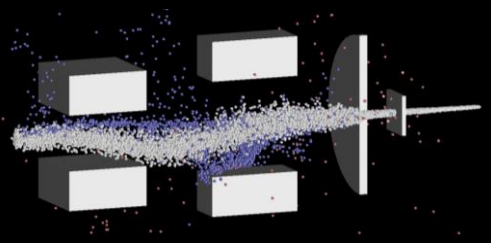


Beam dynamics in rings



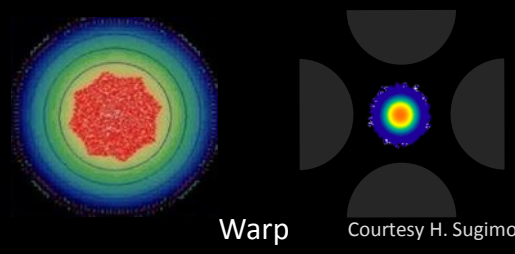
UMER

Multi-charge state beams



LEBT – Project X

Traps



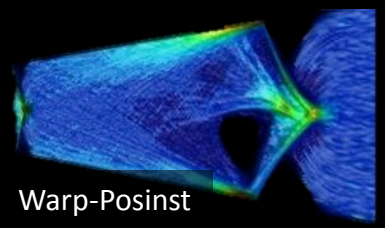
Warp

Courtesy H. Sugimoto

Alpha anti-H trap

Paul trap

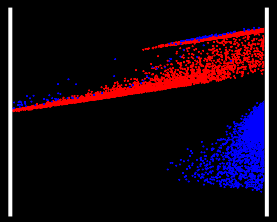
Electron cloud effects



Warp-Posinst

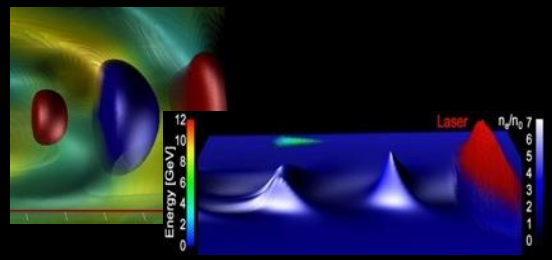
SPS

Multi-pacting



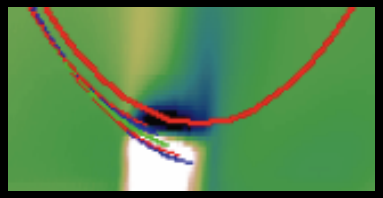
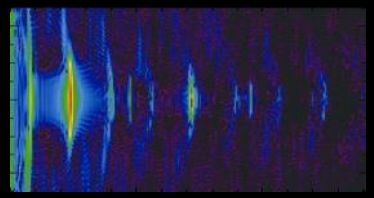
"Ping-Pong" effect

Laser plasma acceleration



BELLA

3D Coherent Synchrotron Radiation



Free Electron Lasers