# Boris correction: exact and approximate

K.G. Miller[1], P.Q. Elias[2], R.A. Fonseca[3], B.J. Winjum[1], F.S. Tsung[1], V.K. Decyk[1], W.B. Mori[1]

[1]UCLA, [2]ONERA, [3]IST

# Outline

UCLA

- OSIRIS current deposits
  1. Charge-conserving current deposit
     - Standard in OSIRIS, satisfies continuity equation
  2. Direct current deposit
     - Use particle positions/shape functions
     - Less noisy, does not satisfy continuity equation

- Divergence cleaning (Boris correction)
  1. Iterative multigrid solver
  2. Exact spectral solver
  3. Exact spectral and finite-difference solver

# PIC method

$k$: continuous (particles)
$ij$: gridded (fields)

**Particle push**

$$\frac{d\boldsymbol{x}_k}{dt} = \boldsymbol{v}_k$$

$$\frac{d\boldsymbol{p}_k}{dt} = q(\boldsymbol{E}_k + \boldsymbol{v}_k \times \boldsymbol{B}_k)$$

**Charge/current deposition**

Calculate $\rho_{ij}$ and $\boldsymbol{j}_{ij}$ from the $\boldsymbol{x}_k$ and $\boldsymbol{p}_k$.

**Field solve**

$$\frac{\partial \boldsymbol{B}_{ij}}{\partial t} = -\nabla \times \boldsymbol{E}_{ij}$$

$$\frac{\partial \boldsymbol{E}_{ij}}{\partial t} = c^2 \nabla \times \boldsymbol{B}_{ij} - \mu_0 c^2 \boldsymbol{j}_{ij}$$

**Field interpolation**

Interpolate $\boldsymbol{E}_{ij}$ and $\boldsymbol{B}_{ij}$ onto the $\boldsymbol{x}_k$.

$\Delta t$

# Gauss's law

UCLA

- Gauss's law not explicitly solved in OSIRIS
- If initial $\rho$ and $\boldsymbol{E}$ satisfy Gauss's law, then continuity equation guarantees Gauss's law at future times

$$\frac{\partial \rho_{ij}}{\partial t} + \nabla \cdot \boldsymbol{j}_{ij} \neq 0$$

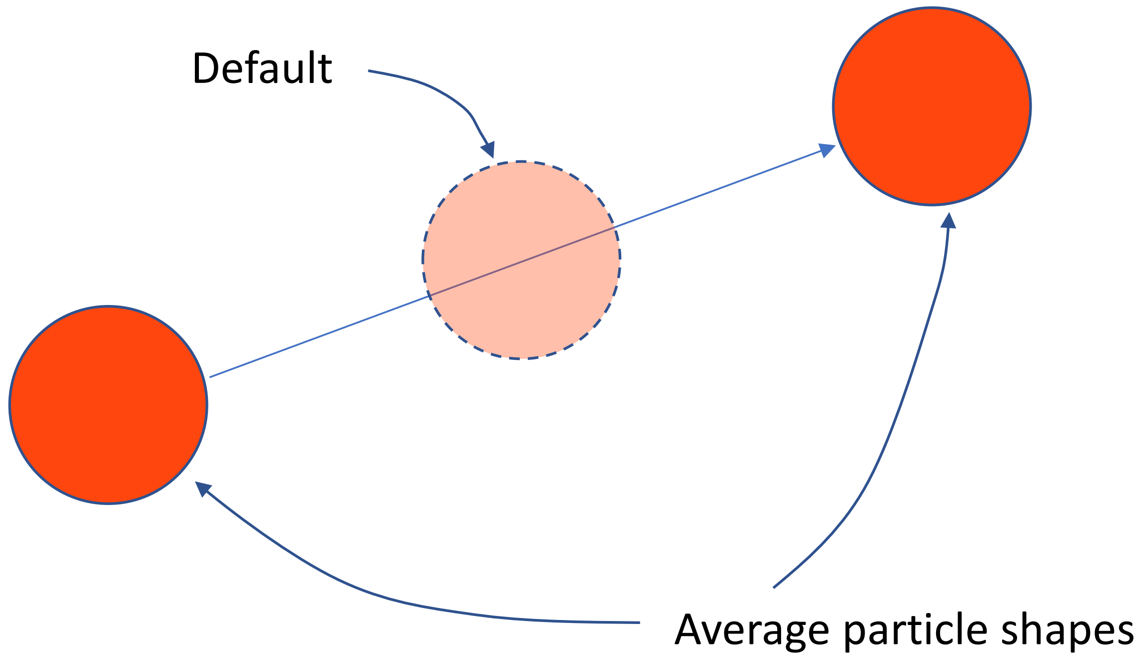
$$\Downarrow$$

$$\nabla \cdot \boldsymbol{E}_{ij} \neq \frac{\rho_{ij}}{\epsilon_0}$$

# Current deposits

UCLA

- Charge-conserving current deposit inherently satisfies continuity equation (and thus Gauss's law)
  - J. Villasenor and O. Buneman, Comput. Phys. Commun. **69**, 306-316 (1992).
- Direct deposit—uses particle shapes



# Boris correction

UCLA

- Solve Poisson's equation to correct electric field

$$\nabla^2\psi = \nabla \cdot \boldsymbol{E} - \frac{\rho}{\epsilon_0} \equiv \rho'$$

- where $\rho'$ is the residual charge error. Then

$$\boldsymbol{E}' = \boldsymbol{E} - \nabla\psi$$

$$\nabla \cdot \boldsymbol{E}' = \frac{\rho}{\epsilon_0}$$

# Iterative multigrid solver: V-cycle

UCLA

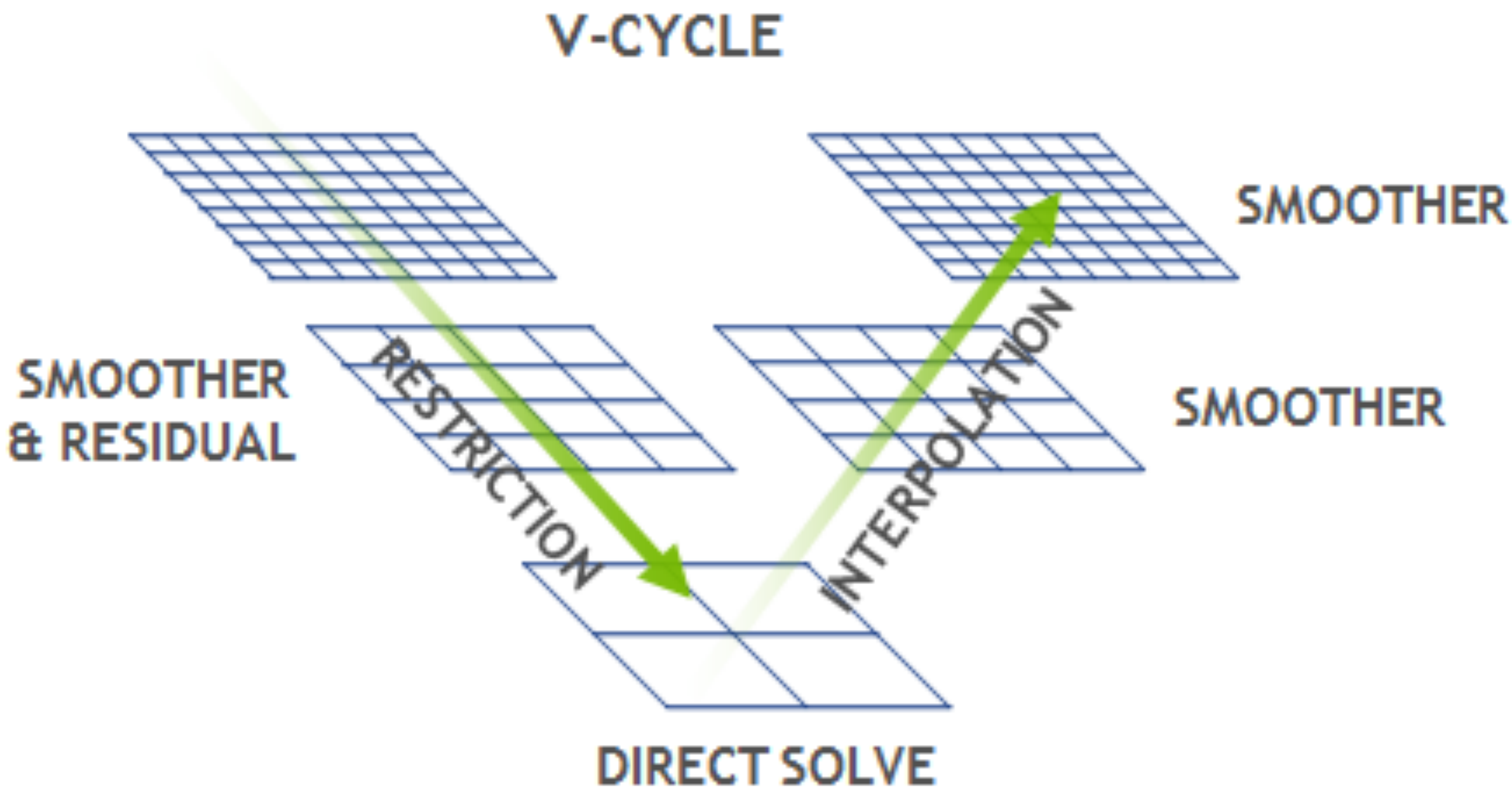
- Solve the matrix equation

$$A\boldsymbol{x} = \boldsymbol{b} \quad (\text{i.e.}, \nabla^2\psi = \rho'),$$

- Start with a guess of $\psi = 0$ $(\boldsymbol{x} = 0)$
- Compute residual $(\boldsymbol{r} = A\boldsymbol{x} - \boldsymbol{b})$
- Restrict onto coarser grid $(\boldsymbol{r} \rightarrow \boldsymbol{r}_1)$
- Smooth solution using point-Jacobi $(A_1\boldsymbol{e}_1 = \boldsymbol{r}_1)$
- Direct matrix solve of small problem $(A_N\boldsymbol{e}_N = \boldsymbol{r}_N)$
- Interpolate onto finer grid $(\boldsymbol{e}_N \rightarrow \boldsymbol{e}_{N-1}^*)$
- Correct the solution $(\boldsymbol{e}_{N-1} = \boldsymbol{e}_{N-1} - \boldsymbol{e}_{N-1}^*)$, $(\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{e}_0^*)$
- Iterate

# Iterative multigrid solver

UCLA



# Exact spectral solver

UCLA

- Assuming Fourier behavior, ODE → algebraic equation

$$\nabla^2\psi = \rho' \rightarrow -k^2\psi = \rho'$$

- However, finite-difference derivative is approximate:

$$\mathcal{F}\left\{\frac{\psi(x+h) - 2\psi(x) + \psi(x-h)}{h^2}\right\} = -\frac{4}{h^2}\sin^2\left(\frac{\pi k}{N}\right)$$

- One dimension ($x$) must be entirely local

# Hybrid spectral/finite-difference

- 3D solver
  - Spectral in $x$ and $y$
  - Finite-difference in $z$

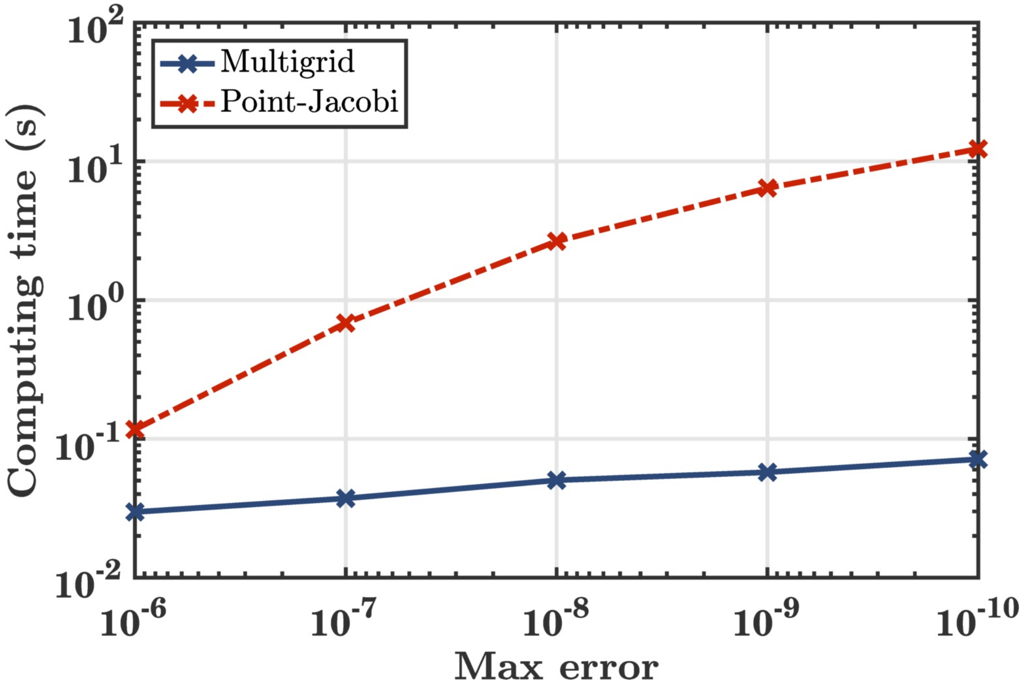
$$\frac{\psi(z+h) - \left(2 + k_x^2 + k_y^2\right)\psi(z) + \psi(z-h)}{h^2} = \rho(z)$$

- Should be more consistent than fully spectral

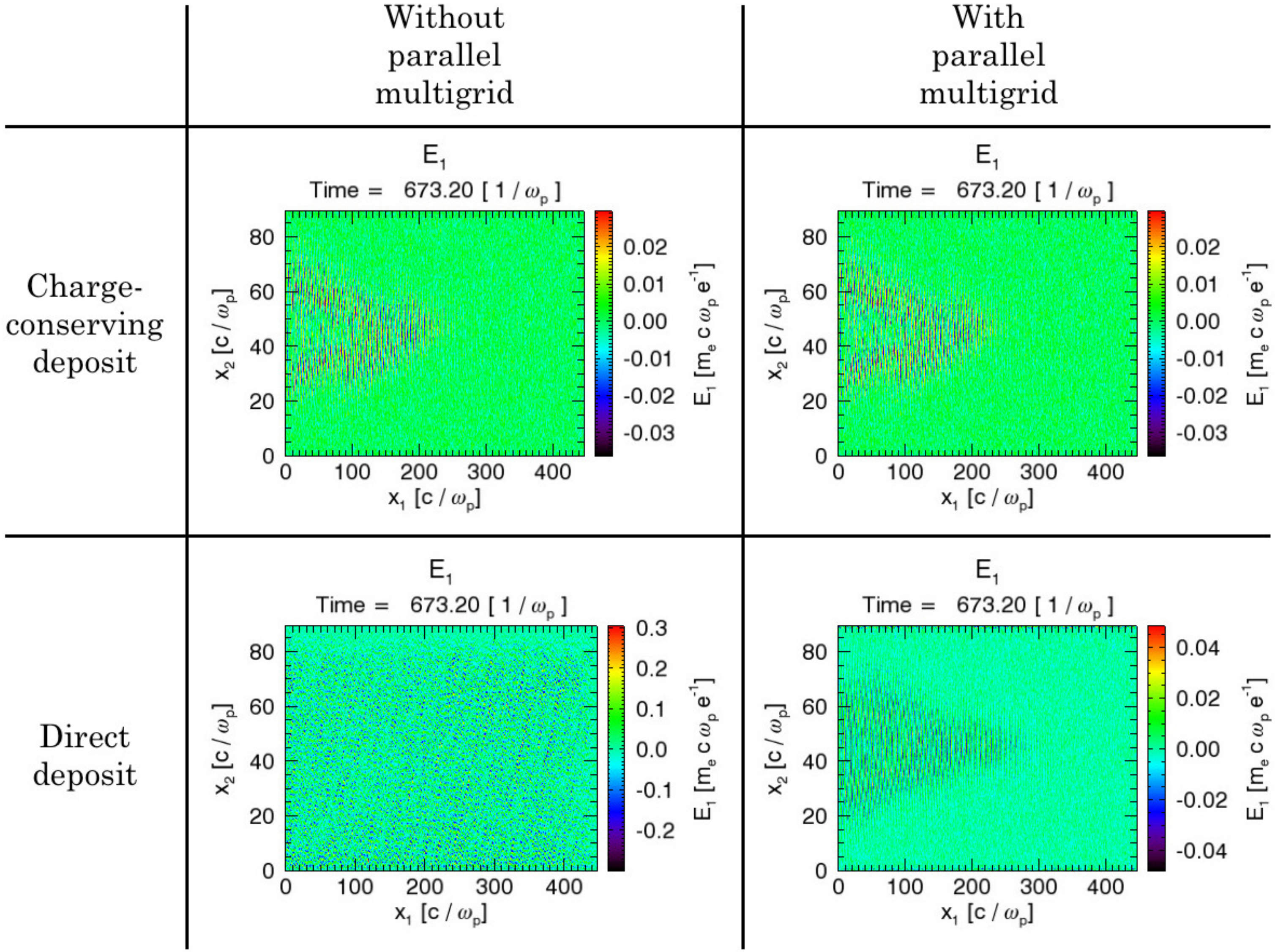
# Speedup

UCLA

- Compared to point-Jacobi, multigrid is much faster
- Spectral methods should be even faster

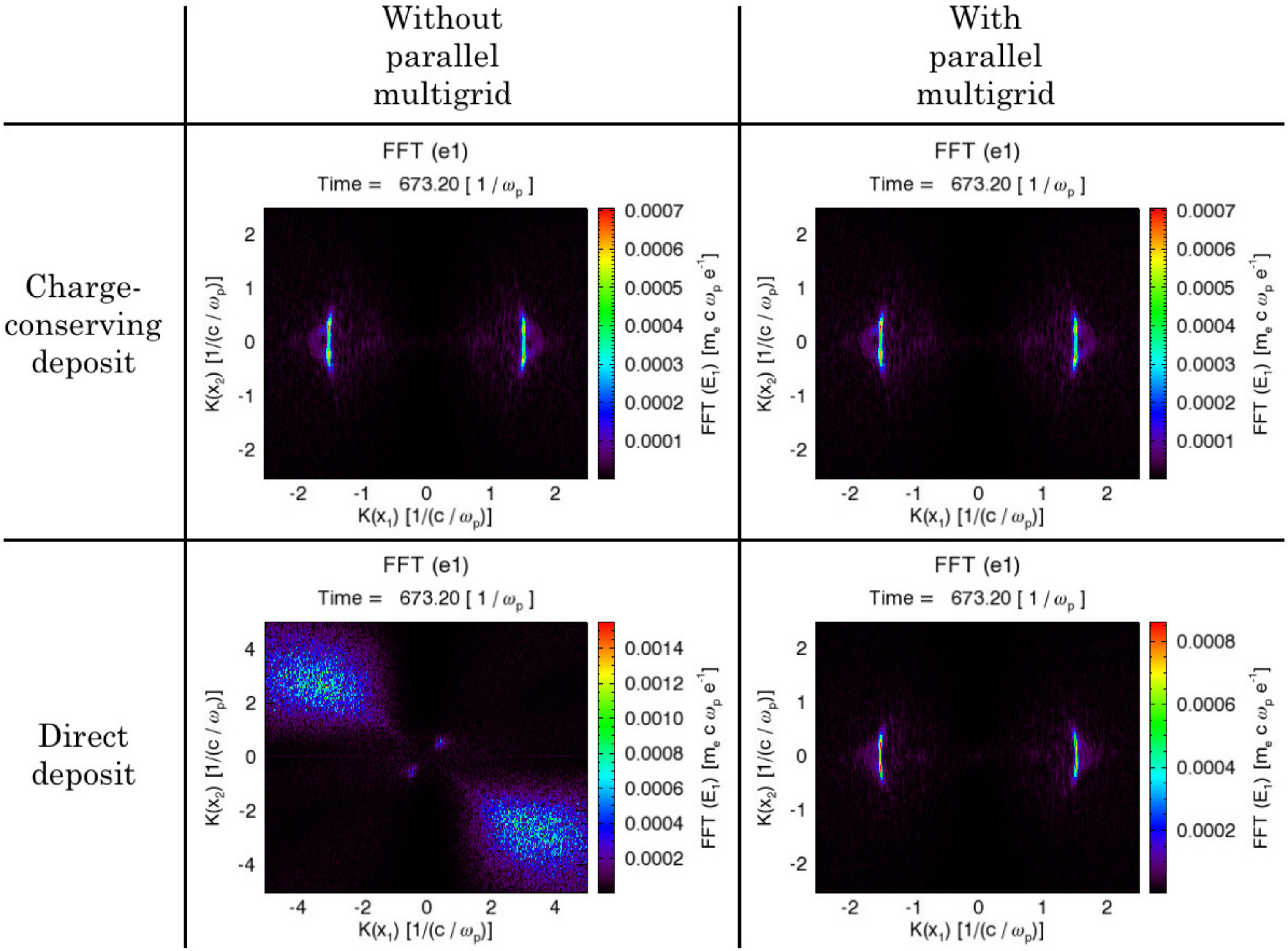


# Test: stimulated Raman scattering



# Test: stimulated Raman scattering

UCLA



# Use direct current deposit

UCLA

- #define DIRECT_DEPOSIT
  - in os-config.h
- #define AVG_SHAPES
  - in os-config.h

# Use Boris correction

UCLA

- if_boris = .true.,
- boris_solver = ’multigrid’, ‘fft’, ‘hybrid’, or ’point-Jacobi’,
  - in el_mag_fld of input file
- Other options:
  - boris_tol = 1.e-14,
  - boris_n = 4,

# Important files

UCLA

- Multigrid:
  - pmg/*   (esp m_parallel_multigrid_solver.f90)
  - os-emf-es-solver-pmg.f03
  - os-emf-marder.f90
  - os-emf-math.f03
  - os-vdf-math.f90
- Exact Boris
  - boris/*
- Branch parallel-multigrid on Github

# Conclusions

UCLA

- Implementations:
  - Direct current deposit
    - Using average position or average particle shapes
  - Boris correction for use with direct deposit
    - Using iterative multigrid, exact spectral, or point-Jacobi
- Uses
  - Less noisy current deposit
  - Solver may be used for other initialization purposes
- Future work
  - Finish implementation, run more physics test cases