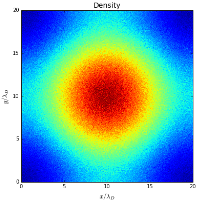
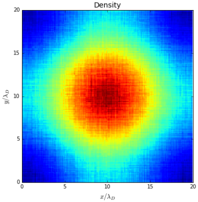
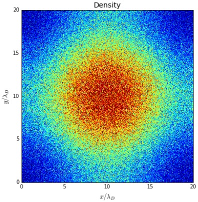
Sparse grids Particle-in-Cell scheme for noise reduction in beam simulations







Antoine Cerfon, *Courant Institute NYU*
with **Lee Ricketson**, *Lawrence Livermore National Laboratory*
ICAP'18, Key West, Florida

October 21, 2018

Curse of dimensionality vs Curse of noise

- Grid based algorithms scale badly with dimension:

  $$\text{Run time complexity } \kappa \sim \frac{h^{-d}}{\Delta t}$$

  Standard 2nd order scheme in space and time: $\epsilon \sim \Delta t^2 \quad , \quad \epsilon \sim h^2$:

  $$\kappa \sim \epsilon^{-\frac{d+1}{2}} \quad \text{"Curse of dimensionality"}$$

  Particularly bad for beam dynamics described by the Vlasov equation $\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f = 0$ with $d = 6$.
- Pure particle methods scale much better with dimension:

  $$\kappa \sim \frac{dN}{\Delta t}$$

  Note: Ideal scaling, through Fast Multipole Method
- Penalty paid with particle methods: slow convergence of error with N: $\epsilon \sim N^{-1/2}$

  For 2nd order in time scheme $\kappa \sim d\epsilon^{-5/2}$ "Curse of noise"

THE PIC APPROACH – THE WORST OF BOTH WORLDS?

- N_g: Total # of grid cells

$$\epsilon \sim \Delta t^2 \,, \quad \epsilon \sim h^2 \,, \quad \epsilon \sim \left(\frac{N_p}{N_g}\right)^{-1/2} \qquad \text{(Standard PIC scheme)}$$

- Let $d_{\mathbf{r}}$ be the number of spatial dimensions

$$N_g \sim h^{-d_{\mathbf{r}}} \;\Rightarrow\; \left(\frac{N_p}{N_g}\right)^{-\frac{1}{2}} \sim N_p^{-\frac{1}{2}} h^{-d_{\mathbf{r}}/2}$$

- Complexity: $\kappa \sim d\frac{p}{\Delta t} \sim d\epsilon^{(-\frac{5}{2}+d_{\mathbf{r}}/2)}$

	Grid based	Pure particle	PIC
1D problem	$\kappa \sim \epsilon^{-3/2}$	$\kappa \sim \epsilon^{-5/2}$	$\kappa \sim \epsilon^{-3}$
2D problem	$\kappa \sim \epsilon^{-5/2}$	$\kappa \sim \epsilon^{-5/2}$	$\kappa \sim \epsilon^{-7/2}$
3D problem	$\kappa \sim \epsilon^{-7/2}$	$\kappa \sim \epsilon^{-5/2}$	$\kappa \sim \epsilon^{-4}$

Improving PIC

Bottom line:

- Grid based solvers scale best for 1-D and 2-D problems
- Pure particle based solvers scale best for 2-D and 3-D problems
- **PIC always scale worse** because of
  - Slow-converging statistical error: “Curse of noise”
  - Exponential dependence of κ on dimension: “Curse of dimensionality” (Although dimension reduced from d to $d_{\mathbf{r}}$)

BUT

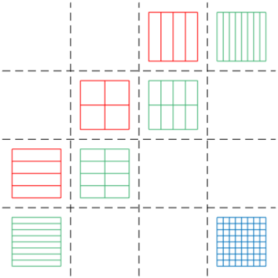
- Many advantages of PIC
  - Relative ease of implementation
  - Easier boundary conditions than in grid based codes
  - Well-suited for massive parallelization
  - Existence of many well established, successful solvers

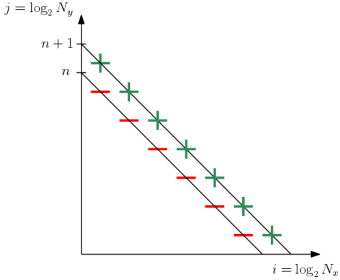
Can one improve PIC by reducing both the grid based error and the noise?

Sparse grids combination technique

- Exact formula for the error from bilinear interpolation with discretization sizes h_x and h_y:

$$\epsilon := u_n(x,y) - u_E(x,y) = C_1(h_x)h_x^2 + C_2(h_y)h_y^2 + C_3(h_x,h_y)h_x^2 h_y^2$$

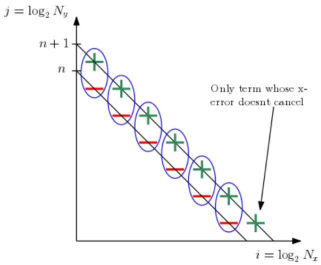


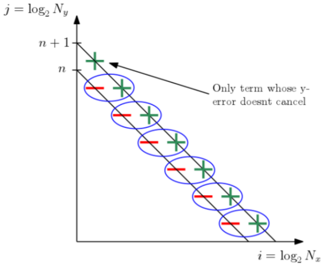


$$u_n(x,y) = \sum_{i+j=n+1} u_{i,j}(x,y) - \sum_{i+j=n} u_{i,j}(x,y)$$

Error cancellation in the combination technique

$$u_n(x,y) = \sum_{i+j=n+1} u_{i,j}(x,y) - \sum_{i+j=n} u_{i,j}(x,y)$$





$$\epsilon = \cancelto{0}{C_1(h_x)h_x^2} + C_1(h_n)h_n^2 + C_2(h_y)h_y^2 + C_3(h_x,h_y)h_x^2h_y^2$$

$$\epsilon = \cancelto{0}{C_2(h_y)h_y^2} + C_2(h_n)h_n^2 + C_1(h_x)h_x^2 + C_3(h_x,h_y)h_x^2h_y^2$$

Error and complexity analysis

$$u_n(x,y) = \sum_{i+j=n+1} u_{i,j}(x,y) - \sum_{i+j=n} u_{i,j}(x,y)$$

- The only surviving terms are $O(h_n^2)$;there are $2n+1$ such terms:

$$\epsilon = O(nh_n^2) = O(h_n^2 |\log h_n|)$$

  For full grid, $\epsilon = O(h_n^2)$

- Solve on $2n+1$ grids, each with $O(h_n^{-1})$ grid points :

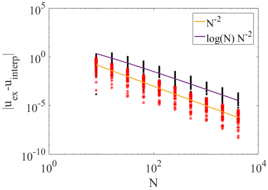
$$\kappa = O(h_n^{-1} |\log h_n|)$$

  For full grid, $\kappa = O(h_n^{-2})$

- **Sparse:** $\kappa \sim \epsilon^{-1/2} |\log \epsilon|^2$ **Full:** $\kappa \sim \epsilon^{-1}$

Breaking the curse of dimensionality

Example: Bilinear interpolation of $\sin(2\pi x)\cos(3\pi y)$ at 50 random points in $[-1,1]\times[-1,1]$



- Same idea applicable in higher dimension
  e.g in 3D: combination along a diagonal *plane*

Sparse: $\kappa \sim \epsilon^{-1/2}|\log\epsilon|^{2(d-1)}$ **Full:** $\kappa \sim \epsilon^{-d/2}$

Much weaker dependence on dimension

- Applicable to higher order schemes
- Schemes with adaptive refinement have been proposed
- Limitations
  - Requires structured mesh
  - Requires more smoothness of function to be represented / Alignment with grid $\to$ we will return to this

Sparse grids for PIC

- Clear that sparse grids can be benefitial for the field parts of the PIC scheme (solve and interpolation)
- What about particle part of PIC?
- Error of approximating true particle density ρ with approximate density ϱ using hat functions

$$\rho(\mathbf{x}_k) - \varrho(\mathbf{x}_k) = C_1(h_x)h_x^2 + C_2(h_y)h_y^2 + C_3(h_x, h_y)h_x^2 h_y^2 + \xi_k,$$

  with ξ_k r. v. with $E[\xi_k] = 0$ and $\mathrm{Var}(\xi_k) \approx 4Q\rho(\mathbf{x}_k)/9 \cdot 1/(h_x h_y N_p)$
- Blue terms are sources of grid based error. Same form as interpolation formula $\Rightarrow$ good for sparse grids
- Yellow term is source of statistical noise. Using the Schwarz equality, one can show that sparse grids have the same benefitial effect on statistical noise[1]

[1] L.F. Ricketson and A.J. Cerfon, *Plasma Phys. Control. Fusion* ***59*** *024002 (2017)*

Sparse grids for PIC – Intuitive advantages

- Statistical noise in PIC depends on the # of particles per cell
- Sparse grids have larger cells ⇒ Sparse PIC has more particles per cell than standard PIC for a given overall # of particles

  ⇒ Sparse PIC has less numerical noise than standard PIC, for a given overall # of particles

  ⇒ Sparse PIC has lower memory requirements than standard PIC, for a given target accuracy

- The grids at the extremes of the grid hierarchy are the only ones to resolve the DeBye length, and do so only in one direction
- Sparse grids require far less memory to store than their full-grid counterparts

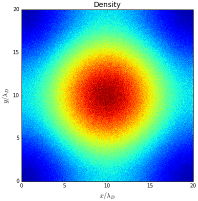
  ⇒ An entire sparse spatial grid easily fits on a single compute node

  ⇒ Much less communication and load-balancing overhead

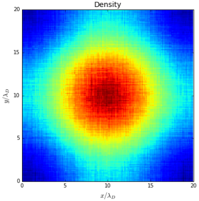
Sparse grids results

2D plasma oscillations

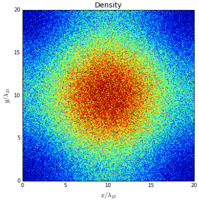
$N_p \sim 10^9$



$N_p \sim 10^7$, **Sparse PIC**

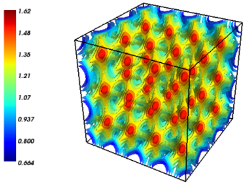


$N_p \sim 10^7$

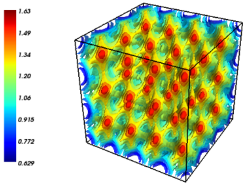


3D plasma oscillations

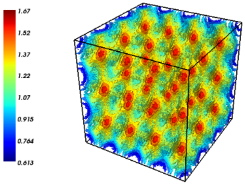
Reference Solution
Time = 500.5 hours



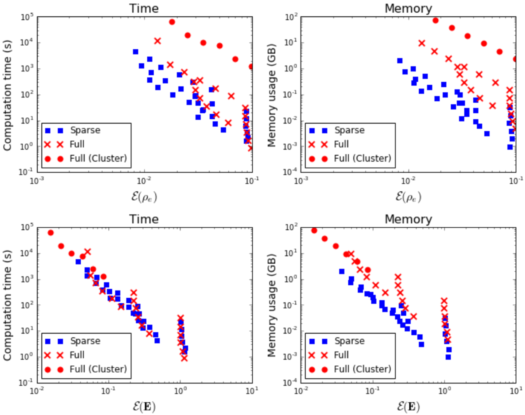
Sparse Solution
Time = 22.4 minutes



Regular Solution
Time = 17.5 hours



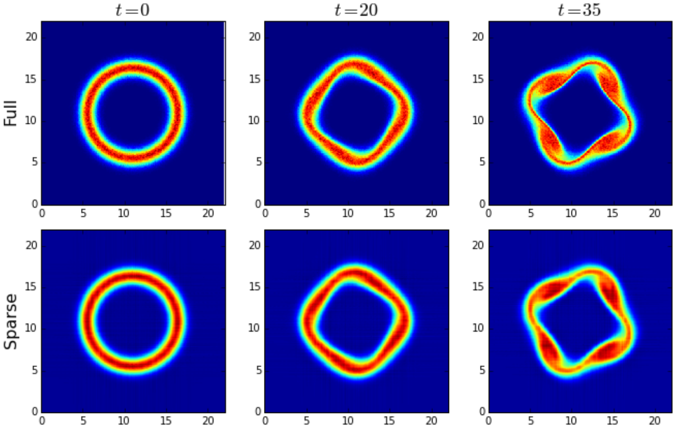
Sparse grids accuracy and memory usage



- Sparse grids often lead to a significant reduction in computational time
- Sparse grids always lead to a significant reduction in memory usage

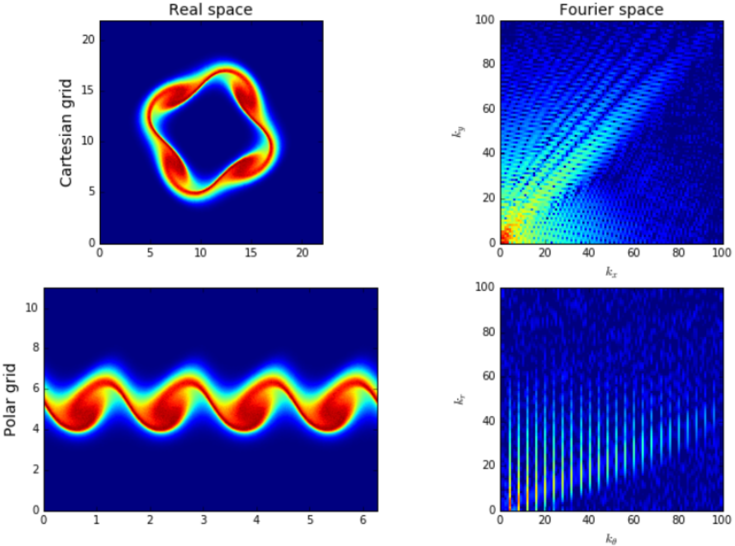
Current limits of Sparse PIC

Illustration with the diocotron instability

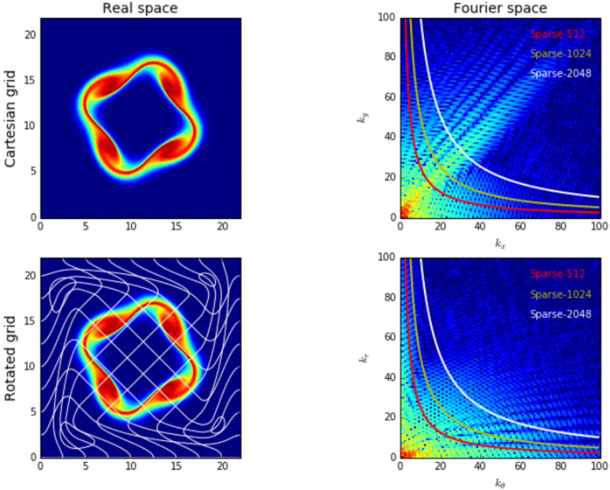


- **Sparse PIC**: 7.8×10^5 particles, 1024×1024 grid, 225 seconds
- **Regular PIC**: 2.6×10^6 particles,256×256 grid, 246 seconds
- Details of billows smeared out in sparse PIC

Polar coordinates for diochotron instability



Optimized coordinates for sparse grids



<u>Idea</u>: use inexpensive, low dimensional continuum equations or inexpensive, high noise PIC simulation to construct optimized coordinate system for sparse grids PIC

Summary – Future work

- Sparse grids can significantly decrease the grid based error and the numerical noise in PIC simulations
- The noise reduction comes from the larger number of particles per cell in sparse PIC than in a standard PIC scheme
- For certain test problems, we demonstrated gains in terms of accuracy and major drops in memory requirements
  For other problems, sparse PIC did not perform better than standard PIC

- Sparse PIC is still in its infancy. We are exploring ways to make it perform consistently better than standard PIC:
  - Use higher order interpolation and higher order shape functions to reduce the grid-based error
  - Construct optimized coordinate systems which best align with solution structure
  - Combine sparse grids with adaptive mesh refinement
- Let us know if you would like to code a sparse PIC solver tailored for your needs!