# Scalable Implicit Solvers for Continuum Kinetic Runaway Electron Simulations

**Qi Tang**

School of Computational Science & Engineering, Georgia Institute of Technology
Theoretical Division, Los Alamos National Laboratory

qtang@gatech.edu

13th Runaway Electron Modelling (REM) Meeting

# We will discuss two complementary developments

## 1. Dynamic AMR + fully implicit solver (PETSc + p4est)

Collaborators: Johann Rudi, Max Heldman *(Virginia Tech)* · Emil M. Constantinescu *(Argonne)* · Xian-Zhu Tang *(LANL)*

## 2. Robust $\ell$AIR AMG preconditioning (hypre)

Collaborators: Ahsan Ali *(Baylor University)* · Ben Southworth, Xian-Zhu Tang *(LANL)* · Jacob Schroder *(Univ. of New Mexico)* · Johann Rudi *(Virginia Tech)*

# The continuum kinetic runaway-electron model

Relativistic drift-kinetic **Fokker–Planck–Boltzmann (RFP)** model for REs. Guiding-center, gyro-averaged, azimuthally symmetric $\Rightarrow$ **2D momentum space** $(p, \xi)$ with a continuous model.

$$\frac{\partial f}{\partial t} - E\left(\xi \frac{\partial f}{\partial p} + \frac{1-\xi^2}{p}\frac{\partial f}{\partial \xi}\right) = \underbrace{C(f)}_{\text{Coulomb}} + \alpha \underbrace{R(f)}_{\text{radiation}} + \underbrace{S(f)}_{\text{knock-on}}$$

- $p = \|\mathbf{p}\|/(m_e c)$ momentum, $\xi = p_{\|}/p$ pitch
- LHS: strong **advection** by parallel field $E$
- $C(f)$: small-angle Coulomb (Fokker–Planck)
- $\alpha R(f)$: synchrotron radiation damping, $\alpha = \tau_c/\tau_s$
- $S(f)$: secondary **knock-on** avalanche (Boltzmann)

**Physics question:** competition between parallel-field acceleration and collisional / radiative drag $\Rightarrow$ primary RE seeds + avalanche growth.

# Why this is numerically challenging

1. **Localized "hot tail".** A thin, *anisotropic-in-pitch* runaway tail; $f$ spans **10–20 orders of magnitude**.
2. **Dynamics vary in time & space.** Advection- vs. diffusion-dominated regions; bulk Maxwellian and runaway tail coexist; advective $\ll$ diffusive time scales $\Rightarrow$ **implicit time stepping**.
3. **Strong convection.** The advection–diffusion operator is **strongly nonsymmetric** (worse as $E$ grows) $\Rightarrow$ standard AMG degrades.

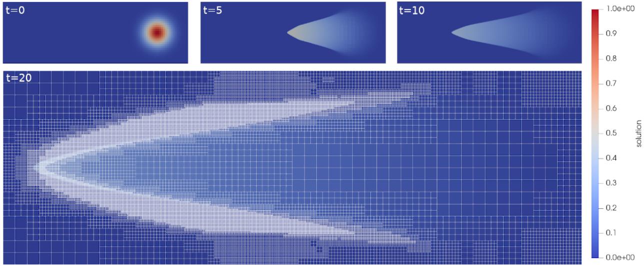
**Goal**

**Scalable, adaptive** solvers — for *both* steady-state and dynamical RE simulations.

# Adaptive mesh refinement (AMR), powered by p4est

**p4est** [Burstedde et al.]

- forest-of-octrees, quad/oct-tree meshes
- space-filling curves $\Rightarrow$ cheap repartitioning + memory locality
- 2:1 balancing, refine/coarsen *without* communication
- scalable to $\mathcal{O}(10^5\text{–}10^6)$ cores




Nonlinear convection–diffusion benchmark: a shock forms and AMR (wireframe, $t$=20) tracks the sharp feature dynamically.

## Our contribution

**DMBF** — a new PETSc data-management structure for p4est in PETSc.

# Novelty 1 — AMR solver with *indicator prediction* for dynamical RE

## Problem

Mesh adapts only every $n_{\text{adapt}}$ steps; features *migrate* between regrids $\Rightarrow$ mesh lags. Frequent regridding is costly.

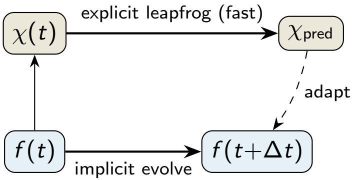
## Idea: predict

Leapfrog the AMR *indicator* ahead with a cheap explicit solver; reduce to a time-independent

$$\chi_{\text{pred}} = \max_{\tau \in [t,\, t+\Delta t_{\text{pred}}]} \chi(\tau),$$

then adapt on the *predicted refinement path*.

**Leapfrog the indicator ahead of the solution**



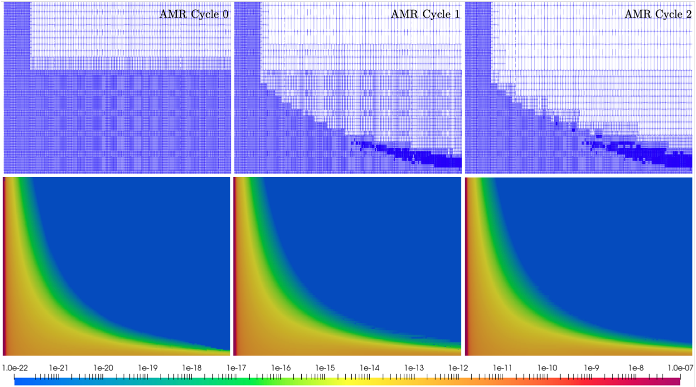

The mesh is refined where the solution *is going*, not where it *was*.

# Novelty 2 — AMR solver for *steady-state* RE

Wrap an **outer AMR loop** around PETSc's (non)linear solvers:

1. solve on current mesh
2. refine tail / coarsen flat regions
3. interpolate as **warm start**, re-solve

Progressively **captures the localized tail**; each warm start makes the next solve **more robust** and faster.




Three steady-state AMR cycles ($E$=3.5): the tail is better resolved each cycle while flat regions ($f \approx 0$) are coarsened.

# Novelty 3 — $\ell$AIR reduction-based AMG preconditioner

**Problem** BoomerAMG degrades / fails for the **nonsymmetric**, advection-dominated RFP operator at large $E$.

$\ell$**AIR** (local approximate ideal restriction) — a **reduction-based**, Petrov–Galerkin AMG:

- $F/C$ splitting, $A = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix}$
- Local ideal restriction $R_{\text{ideal}} = [\,-A_{cf}A_{ff}^{-1} \quad I\,]$
- Schur-complement coarse grid $+$ $F$-relaxation

**Robust for nonsymmetric systems**

Provable scalability for transport, upwind-DG advection–diffusion.

**Our contribution**

Adapt $\ell$AIR solver parameters for the RFP problem

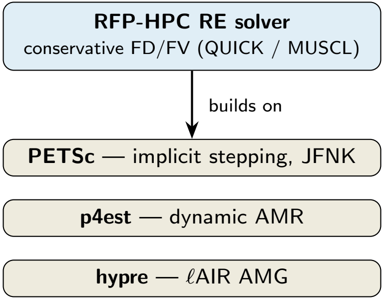
T. A. Manteuffel, J. Ruge, B. S. Southworth, *Nonsymmetric algebraic multigrid based on local approximate ideal restriction ($\ell$AIR)*, SIAM J. Sci. Comput. **40**(6), A4105–A4130, 2018.

# Status of our RFP-HPC solver

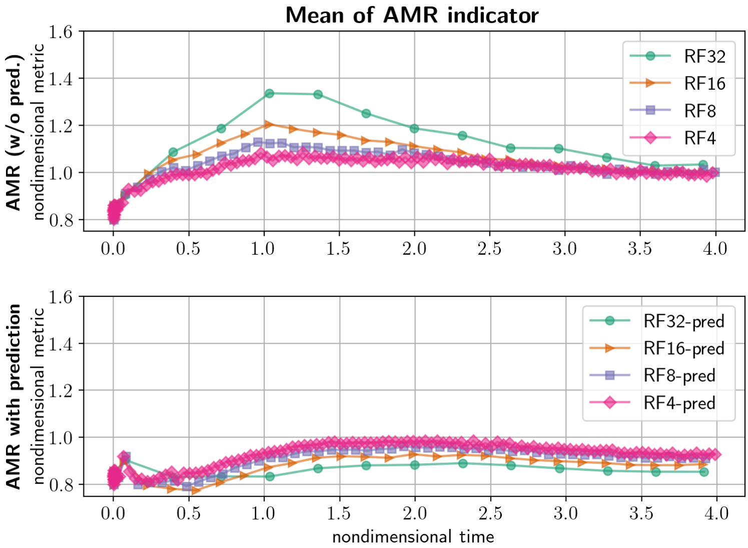
A unified, scalable **0D2P** RE solver:

- **PETSc** — DMBF data structures, fully implicit time stepping (DIRK / ESDIRK), JFNK nonlinear solves, Krylov
- **p4est** — dynamic AMR, parallelism, load balancing
- **hypre** — AMG preconditioning ($\ell$AIR / BoomerAMG)

Supports **small-angle** (Fokker–Planck) *and* **large-angle** (knock-on) collisions + synchrotron radiation; **dynamic and steady-state REs**.



# AMR-prediction pays off (1/2): controlling the indicator

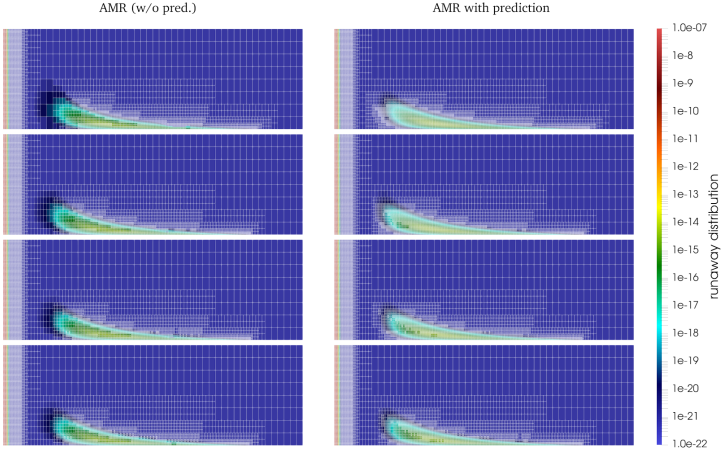



Spatial mean of the metric vs. time, for refinement frequencies RF32/16/8/4.

- **Top (no prediction):** need *frequent* regridding (RF4) to keep the metric near 1.

- **Bottom (prediction):** metric stays low & clustered *for any* frequency — 10–40% lower.

# AMR-prediction pays off (2/2): better solution, modest cost



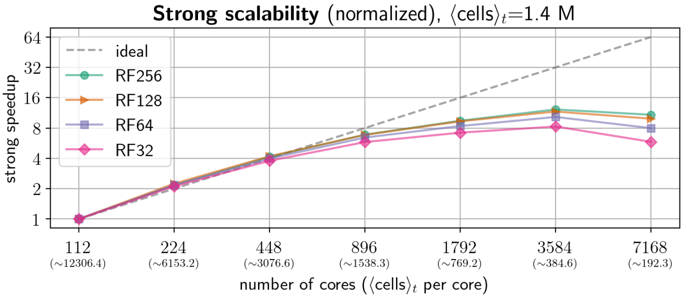

$t = 1.99$: no prediction (left) vs. prediction (right), RF32→RF4 (top→bottom).

- Without prediction the mesh **lags**: artifacts at the distribution edges (worst for RF32).
- With prediction, features are resolved **as well as / better than** the most-frequent RF4 case.

**Cost:** mesh grows only $<$10–20%; runtime $\sim$ 15% more (e.g. RF16 vs RF16-pred) — a modest price for a large accuracy gain.

$\Rightarrow$ adapt *less often* (better scalability) without losing accuracy.

# Algorithmic & parallel scalability with smaller $\Delta t$ (Frontera)



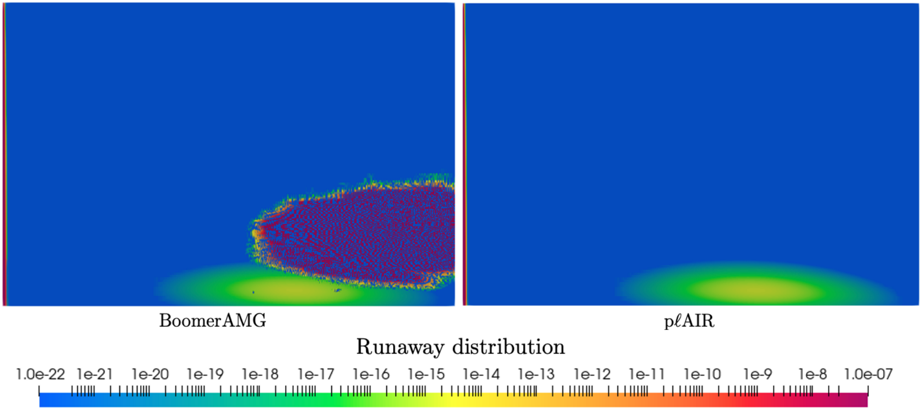

Strong scaling, 1.4 M cells, 112→7,168 cores (speedup up to **12.2**).

**Algorithmic scalability — GMRES iters / step**

| Refine | 1.4 M | 2.9 M | 4.4 M |
|---|---|---|---|
| RF256 | 12.7 – 17.1 | 6.0 | 6.0 |
| RF128 | 11.2 – 14.5 | 6.0 | 6.0 |
| RF64 | 13.6 – 14.1 | 6.0 | 6.0 |
| RF32 | 13.6 – 14.6 | 6.0 | 6.0 |

- Iteration counts **do not grow** with mesh size ⇒ optimal algorithmic scalability.
- Strong scaling near-ideal then communication-bound; weak scaling near-ideal to 7,168 cores.
- Less-frequent regridding (RF256) scales best — enabled by AMR prediction.

# $\ell$AIR advantage (1/2): capturing the right physics with larger $\Delta t$




Time-dependent RFP, $E$=5, $\alpha$=0.1. **BoomerAMG (left):** even after 1000 GMRES iterations it fails to reach tolerance — the tail perturbation is *wrong*. **$\ell$AIR (right):** captures both the Maxwellian bulk and the runaway tail accurately.

# $\ell$AIR advantage (2/2): $\mathcal{O}(1)$ iterations across wide $E$ range with larger $\Delta t$

**$\ell$AIR, time-dependent**

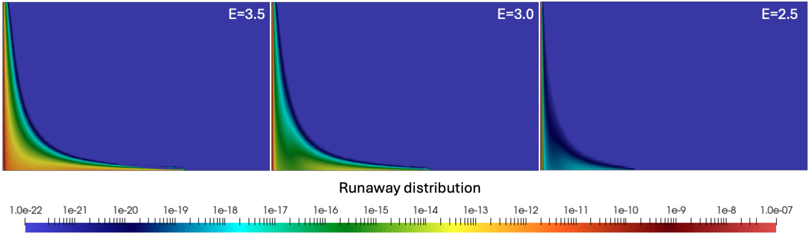
| $E$ | DOFs | avg. GMRES iters |
|---|---|---|
| 5 | 233,678 | 4.63 |
| 10 | 233,742 | 5.11 |
| 15 | 233,652 | 5.95 |
| 20 | 233,129 | 4.39 |
| 25 | 232,131 | 4.91 |
| 30 | 231,086 | 4.95 |
| 35 | 229,920 | 5.10 |

- $\ell$AIR stays at $\sim$**5 iterations** for $E \in [5, 35]$ — a very stiff, strongly advective regime.
- Default **BoomerAMG** diverges (large residual) or hits the 400-iteration cap $\Rightarrow$ omitted.

**Takeaway**

$\ell$AIR turns a regime where AMG *fails* into one solved in $\mathcal{O}(1)$ iterations.

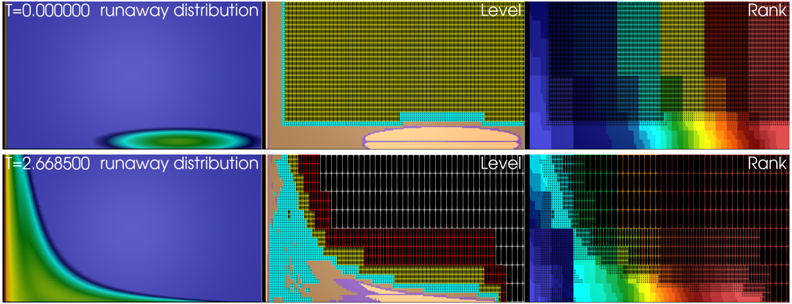
# Steady-state solver — the more challenging case




| FGMRES + $\ell$AIR | | |
|---|---|---|
| $E$ | DOFs | FGMRES |
| 3.5 | 141,804 | 9.25 |
| 3.0 | 124,612 | 9.83 |
| 2.5 | 102,048 | 8.50 |

- Standard BoomerAMG **does not converge at all** for steady state.
- $\ell$AIR keeps low, stable iteration counts.

# Physics demo: a Maxwellian bulk interacting with a runaway tail




Distribution (left) — adaptive mesh & levels (mid) — MPI ranks (right) at $t = 0$ (top) and final time (bottom)

Full physics: Fokker–Planck + synchrotron + knock-on source + partial screening.

- AMR resolves *both* the bulk boundary layer ($\sim 10^{-2}$) and the runaway tail ($\sim 10^{-20}$).
- Dynamic **load balancing** across MPI ranks (space-filling curve).

Avg. DOFs $\approx$ 71k vs. 0.62 M on a structured stretched grid in earlier work.

# Conclusions & outlook

**Scalable implicit solvers for continuum kinetic RE simulations:**

- **Dynamic AMR with indicator prediction** — resolves migrating features with cheap explicit leapfrogging; adapt less often, scale better, improved accuracy.
- **Steady-state AMR** with warm starts — robustly captures the localized tail.
- $\ell$**AIR reduction-based AMG** — $\mathcal{O}(1)$ iterations across a wide $E$ range where BoomerAMG stalls or fails; the enabler for steady-state solves.
- **Robust + scalable:** mesh-independent iteration counts, strong/weak parallel scalability to $\mathcal{O}(10^4)$ cores.

**RFP-HPC = PETSc + p4est + hypre** for a 0D2P RE model with small- and large-angle collisions.

**Outlook:** self-consistent $E$-field coupling and GPU acceleration.

Thank you! `qtang@gatech.edu`