Iterative structured-mesh Ginzburg-Landau solver on GPUs

Dmitry Karpeev[1], Igor Aronson[2], George Crabtree[2], Andreas Glatz[2], Alexei Koshelev[2], Todd Munson[1], Jason Sarich[1], Stefan Wild[1]
[1]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA
[2]Materials Science Division, Argonne National Laboratory, Argonne, IL 60439, USA

GL model

Time-dependent Ginzburg-Landau

$$\frac{\partial \Psi}{\partial t} = -\frac{\delta \mathcal{F}_{\mathrm{GL}}}{\delta \Psi^*}, \quad \frac{\delta \mathcal{F}_{\mathrm{GL}}}{\delta \mathbf{A}} = 0$$

$$u(\partial_t + i\mu)\psi = \epsilon(\mathbf{r})\psi - |\psi|^2\psi + (\nabla - i\mathbf{A})^2\psi + \zeta(\mathbf{r},t)$$
$$\kappa^2 \nabla \times (\nabla \times \mathbf{A}) = \mathbf{J}_n + \mathbf{J}_s + \mathcal{I},$$

Coupled system for ψ and $\mathbf{A}$:

ψ	complex order parameter characterizing density of Cooper pairs
$\mathbf{A}$	vector potential for magnetic field
ζ and $\mathcal{I}$	thermal fluctuations
$\epsilon(\mathbf{r}) = \frac{T_c(\mathbf{r})-T}{T} \to 0$	for $T \to T_c$ (critical temperature)

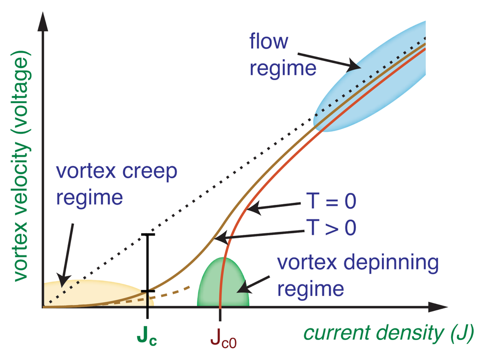
$$\mathbf{B} = \nabla \times \mathbf{A}, \quad \mathbf{E} = -\frac{1}{c}\partial_t \mathbf{A} - \nabla\mu$$

(Critical) Current

Total current: $\mathbf{J}=\mathbf{J}_s+\mathbf{J}_n$

$$\mathbf{J} = \mathrm{Im}\left[\psi^*(\nabla - i\mathbf{A})\psi\right] - (\nabla\mu + \partial_t \mathbf{A})$$

critical current J_c:
- no unique definition
- usually defined when voltage V is a small percentage δ (here 1%) of the free flow value V_{ff}
- ***J_c* calculated e.g. by a bisection method**



- ***Critical current determined by long-time evolution of TDGL (to stationary flow)***
- ***Dominated by rare events of vortex depinning and avalanches***
- ***Frequency and duration of pinning/depinning depends on configurations of inclusions***
- ***Suitable pinning configurations must be determined using geometry optimization***

Discretization & Pinning

large-λ limit

- for large-λ (or κ) our equation system reduces to the GL equation only, and we can keep the magnetic vector potential constant.
- we choose the gauge for the vector potential as $\mathbf{A} = -B_z y \mathbf{e}_x^{(0)}$
  i.e. we have constant magnetic field in z-direction
- the current simplifies to $\mathbf{J} = \mathrm{Im}\left[\psi^*(\nabla - i\mathbf{A})\psi\right] - \nabla\mu$

Discretization on regular grid

- time discretization

$$\psi_m - \psi_m^- = dt\left[\epsilon_m\psi_m^{\bowtie} - \frac{\beta}{2}|\psi_m|^2\psi_m - \frac{\beta}{2}|\psi_m^-|^2\psi_m^- + \gamma\left(\nabla - \imath\mathbf{A}\right)^2\psi_m^{\bowtie} + \zeta_m(t)\right]$$

implicit Crank-Nicolson, with

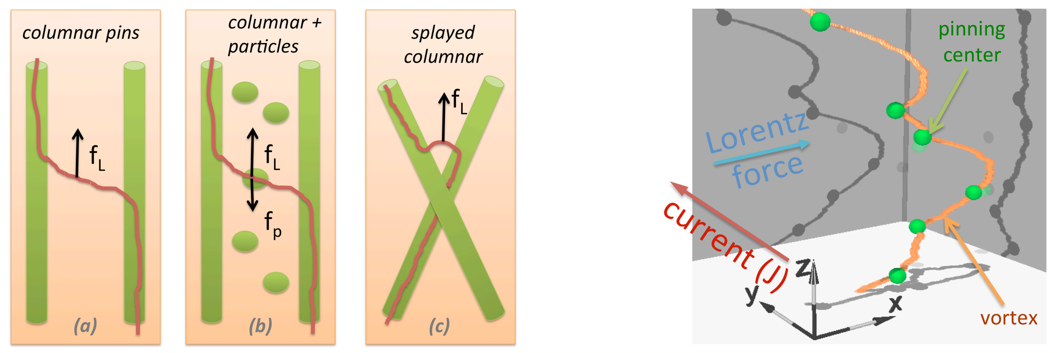
$$\begin{aligned}\psi_m &= \psi_{x,y,z}(t)\\ \psi_m^- &= \psi_{x,y,z}(t-dt)\\ \psi_m^{\bowtie} &= \left(\psi_m + \psi_m^-\right)/2\end{aligned}$$

- Laplacian

$$(\nabla - \imath\mathbf{A})^2\psi_m = \frac{U_j\psi_{i+1,j,k} + U_j^*\psi_{i-1,j,k} - 2\psi_m}{dx^2} + \frac{\psi_{i,j+1,k} + \psi_{i,j-1,k} - 2\psi_m}{dy^2} + \frac{\psi_{i,j,k+1} + \psi_{i,j,k-1} - 2\psi_m}{dz^2}$$

with $\mathbf{A} = -B_z y \mathbf{e}_x^{(0)}$, $U_j = e^{\imath j dx dy B_z}$ "link" variables

Modeling of inclusions and pinning



Inclusions and defects are modeled by T_c modulation → corresponding to normal metallic pinning centers: spatial variation of ϵ(r) [positive in the superconductor, negative in the defect]
- **arbitrary geometry, but**
- **on a regular grid (see next and poster 2)**

Dynamics & Meshing

Dynamics

- separate dc electric field

$$\begin{aligned}\mu(\mathbf{r}) &= -E_x x + \tilde{\mu}(\mathbf{r})\\ \psi(\mathbf{r}) &= \tilde{\psi}(\mathbf{r})\exp(iKx)\end{aligned} \quad (\star)$$

- fix the current (ordinary differential equation for K)

$$J_{\mathrm{ext}} = \mathrm{Im}\left[\left\langle\tilde{\psi}^*(\nabla_x - iA_x)\tilde{\psi}\right\rangle\right] + K\left\langle|\tilde{\psi}|^2\right\rangle + \partial_t K$$

this gives the voltage-current characteristics $E_x(J_{ext})$ through $\partial_t K=E_x$ which we use to obtain the critical current J_c

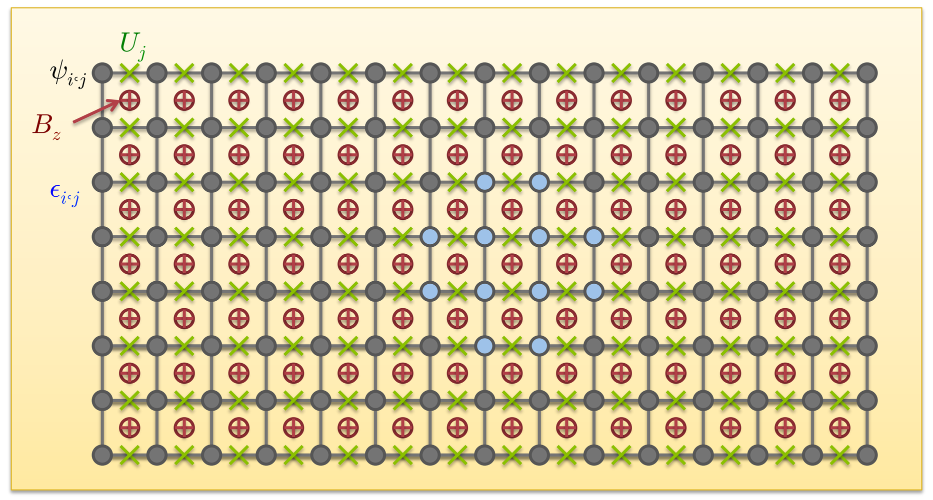
- solve the Poisson equation for μ (follows from current conservation)

$$\Delta\tilde{\mu} = \nabla\cdot\mathrm{Im}\left[\tilde{\psi}^*(\nabla - i\mathbf{A})\tilde{\psi}\right] + K\nabla_x|\tilde{\psi}|^2$$

→ solved by super-relaxation with Jacobi iterations

$$\partial_\tau\tilde{\mu} = -\Delta\tilde{\mu} + \nabla\cdot\mathrm{Im}\left[\tilde{\psi}^*(\nabla - i\mathbf{A})\tilde{\psi}\right] + K\nabla_x|\tilde{\psi}|^2$$

simulation grid



Boundary conditions

Laplacian with external current

- using transformation ($\star$), the TDGL can be written as

$$u(\partial_t + i\tilde{\mu})\tilde{\psi} = \epsilon(\mathbf{r})\tilde{\psi} - |\tilde{\psi}|^2\tilde{\psi} + \left(\nabla + i\mathbf{K} - i\mathbf{A}\right)^2\tilde{\psi} + \zeta(\mathbf{r},t)$$

where we defined $E_x=\partial_t K$ and **K**=(K,0,0)

- the discrete Laplacian in x-direction is then given by

$$\left(\partial_x - \imath(A_x - K)\right)^2\tilde{\psi}_m = \frac{U_K U_j\tilde{\psi}_{i+1,j,k} + U_K^* U_j^*\tilde{\psi}_{i-1,j,k} - 2\tilde{\psi}_m}{dx^2}$$

with $U_K=e^{iK\,dx}$

No-current BC

- for surfaces, we use the no-current condition: $\partial_{x,y,z}\psi = 0$ and $\partial_{x,y,z}\tilde{\mu} = 0$
- not applicable in x-direction if external current is applied

→ this defines the laplacian at the surface layers

Periodic/Quasi-periodic BC

- for $\lambda\to\infty$ and finite vector potential (as before) [no current], we can use periodic conditions in x- & z-directions and quasi-periodic in y-direction (y=0,…,L_y):

$$\psi_{x,L_y,z} = \psi_{x,0,z}e^{\imath 2\pi B_z x L_y/L_x}$$

(challenge on unstructured/refined meshes)

- for finite λ (finite κ) we also have boundary conditions for the vector potential.

Iterative solver on GPUs

- due to the laplacian, an explicit time integration requires very small dt, i.e., it takes a long time to reach a steady state (easy to parallelize)
- due to the boundary conditions, we cannot use a FFT-based spectral solver for *arbitrary* magnetic field (for certain discrete values it is possible)

➔ **use iterative solver for the implicit time integration**

- the GL equation can be written in the simple form $\mathcal{M}\Psi = \mathbf{b}$

$$\begin{aligned}\Psi &= (\psi_0, ..., \psi_{N-1})^T\\ \mathbf{b} &= (b_0, ..., b_{N-1})^T\end{aligned}$$

$$\begin{aligned}\mathcal{M}_{m,m} &\equiv 1 + \frac{dt}{2}\left(\beta|\psi_m^{\mathrm{est}}|^2 + 2\gamma\left[\frac{1}{dx^2} + \frac{1}{dy^2} + \frac{1}{dz^2}\right] - \epsilon_m\right)\\
\mathcal{M}_{m,i+1+N_x(j+kN_y)} &\equiv -\gamma\frac{dt}{2}\frac{U_j}{dx^2}\\
\mathcal{M}_{m,i-1+N_x(j+kN_y)} &\equiv -\gamma\frac{dt}{2}\frac{U_j^*}{dx^2}\\
\mathcal{M}_{m,i+N_x(j+1+kN_y)} &= \mathcal{M}_{m,i+N_x(j-1+kN_y)} \equiv -\frac{\gamma}{2}\frac{dt}{dy^2}\\
\mathcal{M}_{m,i+N_x(j+(k+1)N_y)} &= \mathcal{M}_{m,i+N_x(j+(k-1)N_y)} \equiv -\frac{\gamma}{2}\frac{dt}{dz^2}\end{aligned}$$

$$b_m = \psi_m^- + \frac{dt}{2}\left(\epsilon_m\psi_m^- - \beta|\psi_m^-|^2\psi_m^- + \gamma\left(\nabla - \imath\mathbf{A}\right)^2\psi_m^- + 2\zeta_m(t)\right)$$

- in order to invert the matrix $\mathcal{M}$ on a parallel machine, we use the Jacobi iteration

we write $\mathcal{M} = \mathcal{D} + \mathcal{R}$ where $\mathcal{D}$=diag($\mathcal{M}$)

then iterate $\Psi^{(k+1)} = \mathcal{D}^{-1}\left(\mathbf{b} - \mathcal{R}\Psi^{(k)}\right)$

convergence criterion

$$\|\mathbf{r}_k\|_{\max} \ll 1$$

residual $\mathbf{r}_k \equiv \mathbf{b} - \mathcal{M}\Psi^{(k)}$

This scheme is implemented for GPUs and requires for a typical dt=0.1 and dx=dy=dz=0.5, **4-5 iterations** only to reach an accuracy of 10^{-6}

Performance & Outlook

Implementation and Performance

the GPU implementation of the Jacobi scheme requires
- one CUDA-kernel routine for the construction of vectors b and $\mathcal{D}^{-1}$
- and an iteration step to get the next approximation $\Psi^{(k+1)}$
- for finite external current the ODE for K needs to be integrated (requiring efficient CUDA-reduction kernels, and
- the Poisson equation for μ is solved by super-relaxation & Jacobi iterations

→ **The performance is a few 100x faster than explicit integration, which would require a very small *dt* to be stable**

On a single 6GB Tesla or Kepler GPU card we can simulate system sizes up to 512^3

Next steps

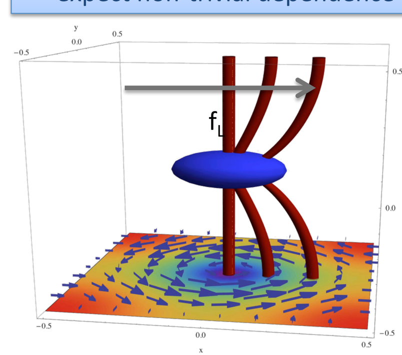
General
- data management and analysis optimizations
- visualization

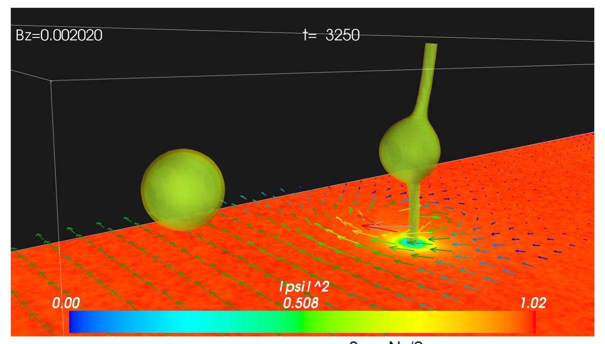
Large-λ limit code (in progress)
- implementation of different magnetic field directions
- application and comparison with experiments

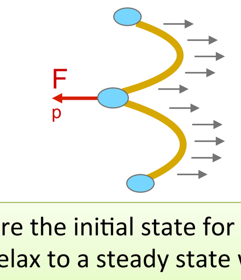
Major plans to the next year
- Implementation of full GL+Maxwell equation system on GPUs
- Development of unstructured FEM codes for extended pinning structures (cont)
- Automation of pinning structure meshing
- Design of optimal pinning structure sampling on leadership-class machines

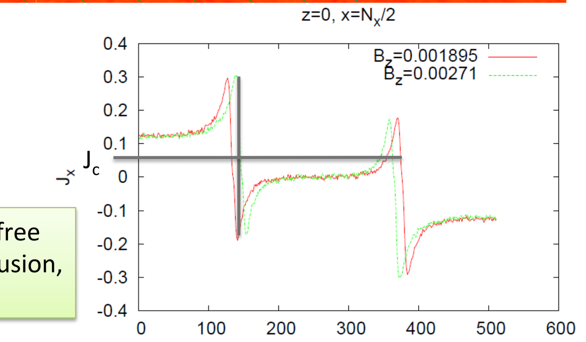
Example: Single inclusion

Find the pin-braking force to detach a vortex from a single nanoscale ellipsoidal inclusion
- key parameter in strong-pinning theory
- expect non-trivial dependence on defect size

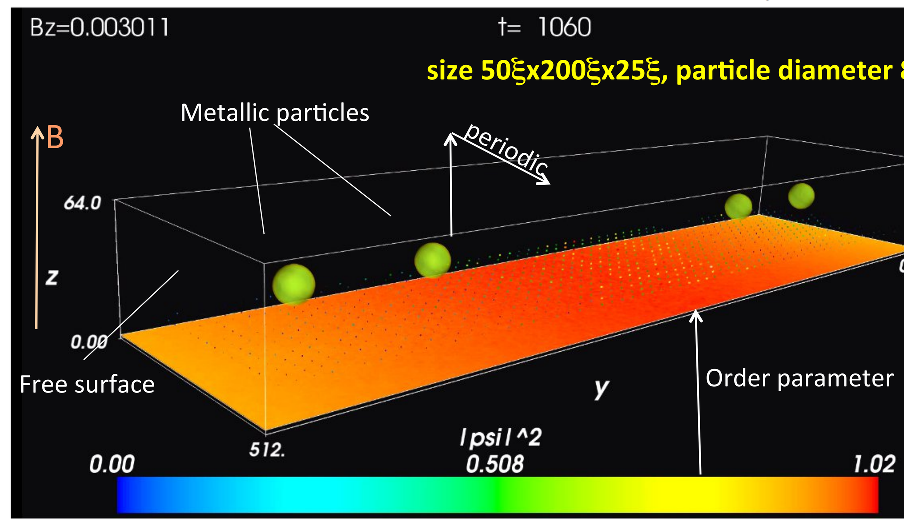






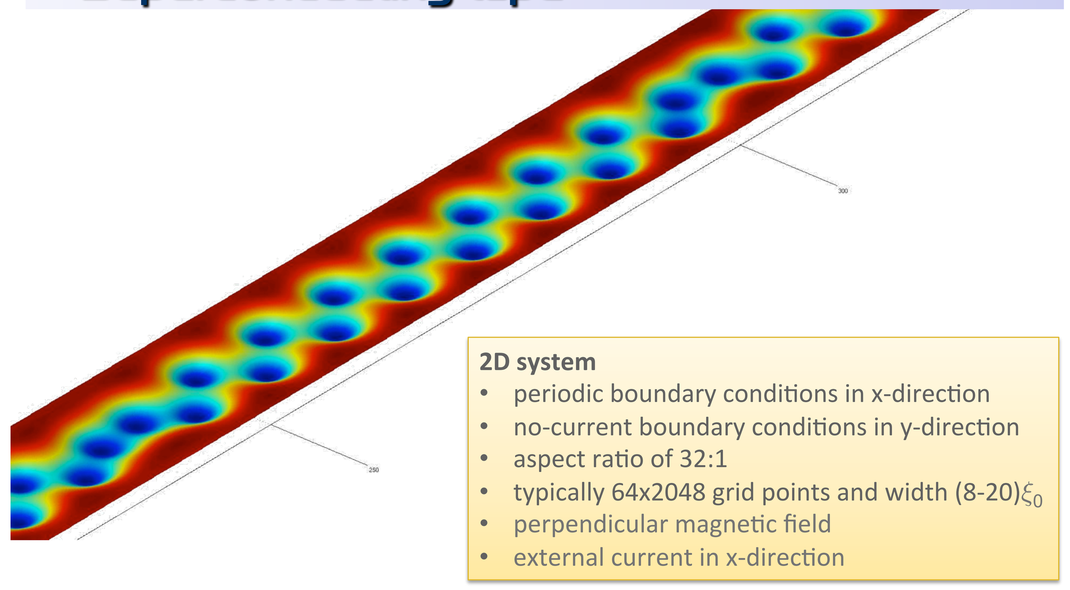


→ prepare the initial state for a single free vortex, relax to a steady state with inclusion, then slowly increase the field.



Example: Geometrical constraints

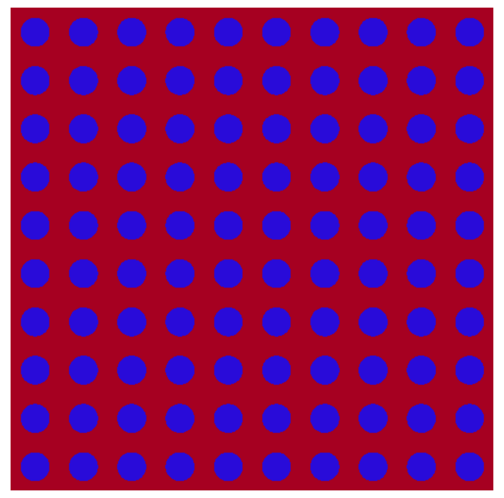
Superconducting tape



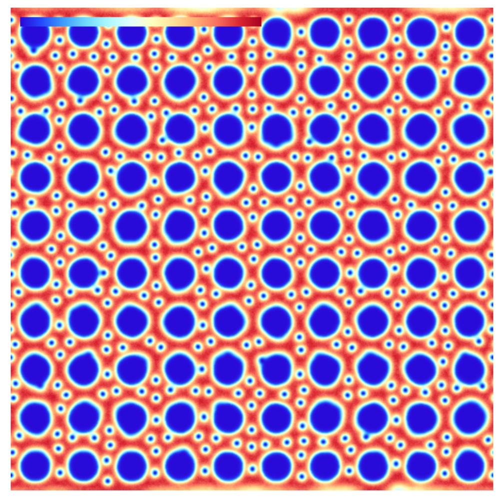
2D system
- periodic boundary conditions in x-direction
- no-current boundary conditions in y-direction
- aspect ratio of 32:1
- typically 64x2048 grid points and width (8-20)ξ_0
- perpendicular magnetic field
- external current in x-direction

Perforated film

- modulation of the linear coefficient ε(r), where $T>T_c$ in the "holes"



pattern, prescribed by ε(r), $N=512^2$



order parameter amplitude at finite field

SciDAC-3 PI Meeting ● July 24-26, 2013 ● Washington DC, USA