

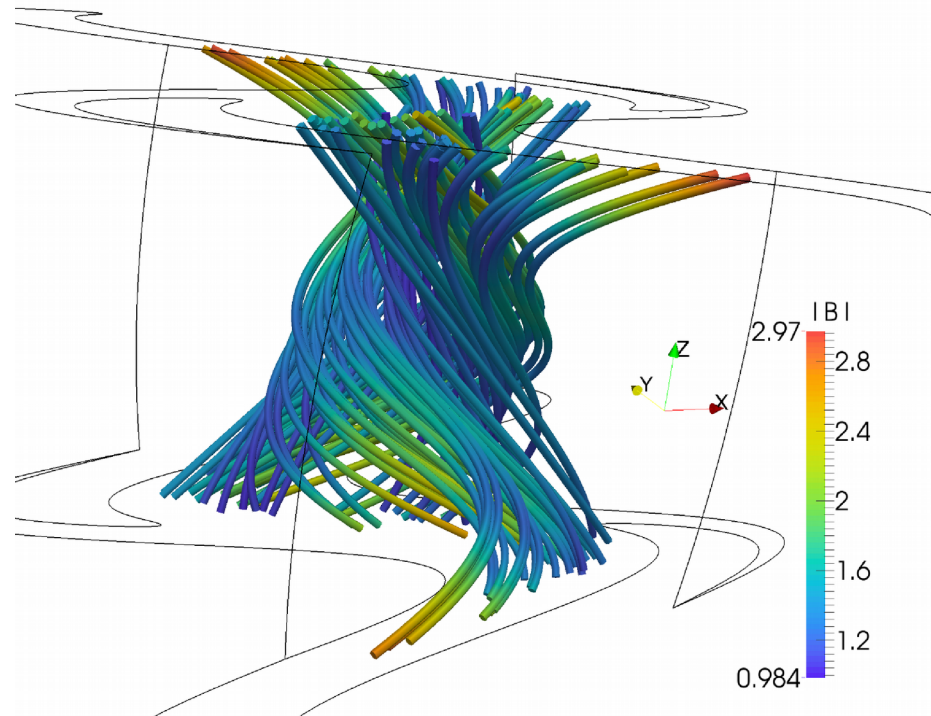


Universiteit Leiden



Topology Conserving Magnetic Field Evolution

**S. Candelaresi, D. Pontin, G. Hornig,
C. Berg-Smiet, D. Bouwmeester**



Magnetic Field Relaxation

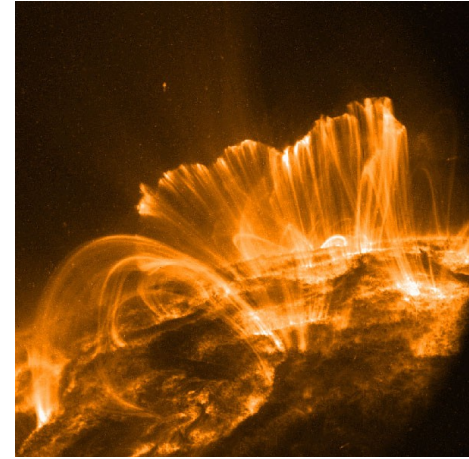
Solar corona: low plasma beta and magnetic resistivity

NASA

➔ Force-free magnetic fields

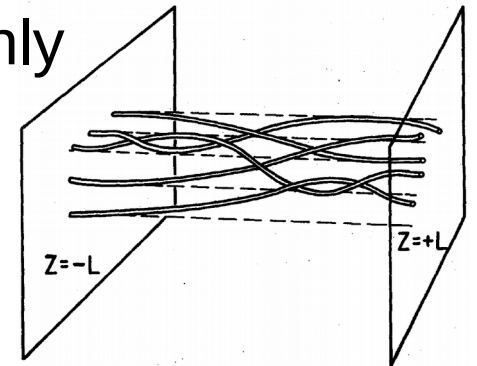
➔ Minimum energy state

$$(\nabla \times \mathbf{B}) \times \mathbf{B} = 0 \Leftrightarrow \nabla \times \mathbf{B} = \alpha \mathbf{B}$$



Parker: Equilibrium with the same topology exists only if the twist varies uniformly along the field lines. Strongly braided fields → topological dissipation.

(Parker 1972)



Braided fields from foot point motion complex enough. (Parker 1983)

Solutions possible with filamentary current structures (sheets).

(Mikic 1989, Low 2010)

Ideal Field Relaxation

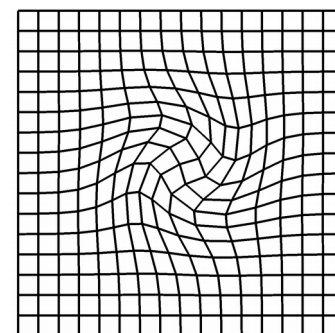
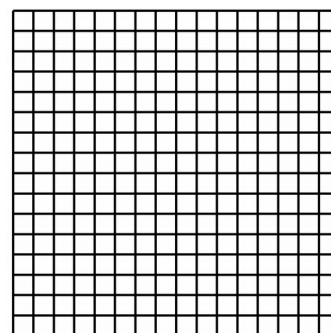
Ideal induction eq.: $\frac{\partial \mathbf{B}}{\partial t} - \nabla \times (\mathbf{u} \times \mathbf{B}) = \mathbf{0}$

Lie-transport:

$$\mathbf{x}(\mathbf{X}, 0) = \mathbf{X}$$

$$\mathbf{x}(\mathbf{X}, t)$$

$$\frac{\partial}{\partial t} \beta(\mathbf{X}, t) + \mathcal{L}_{\mathbf{u}} \beta(\mathbf{X}, t) = 0$$



Pull-back:

$$(\mathbf{x}^*(t)\beta)(\mathbf{X}, t) = \beta(\mathbf{X}, t)$$

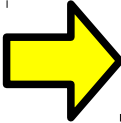
Field evolution: $B_i(\mathbf{X}, t) = \frac{1}{\Delta} \sum_{j=1}^3 \frac{\partial x_i}{\partial X_j} B_j(\mathbf{X}, 0)$

$$\Delta = \det \left(\frac{\partial x_i}{\partial X_j} \right)$$

Preserves topology and divergence-freeness.

Ideal Field Relaxation

Magneto-frictional term: $\mathbf{u} = \mathbf{J} \times \mathbf{B}$ $\mathbf{J} = \nabla \times \mathbf{B}$

 $\frac{dE_M}{dt} < 0$ (Craig and Sneyd 1986)

Fluid with pressure: $\mathbf{u} = \mathbf{J} \times \mathbf{B} - \beta \nabla \rho$

Fluid with inertia: $d\mathbf{u}/dt = (\mathbf{J} \times \mathbf{B} - \nu \mathbf{u} - \beta \nabla \rho) / \rho$

For $\mathbf{J} = \nabla \times \mathbf{B}$ use mimetic numerical operators.

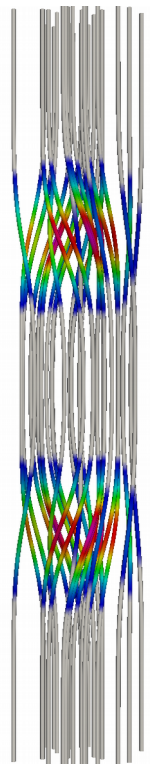
$$\nabla_m \cdot \nabla_m \times \mathbf{B} = 0 \quad \nabla_m \times \nabla_m f = 0 \quad (\text{Hyman, Shashkov 1997})$$

Own GPU code GLEMUR: (<https://github.com/SimonCan/glemur>)

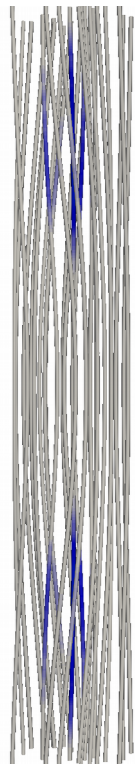
(Candelaresi et al. 2014)

Proof of Concept

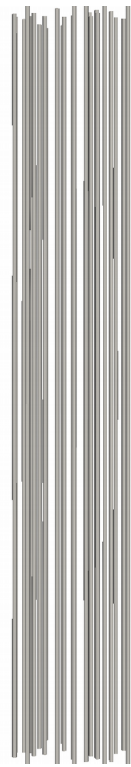
Magnetic streamlines:



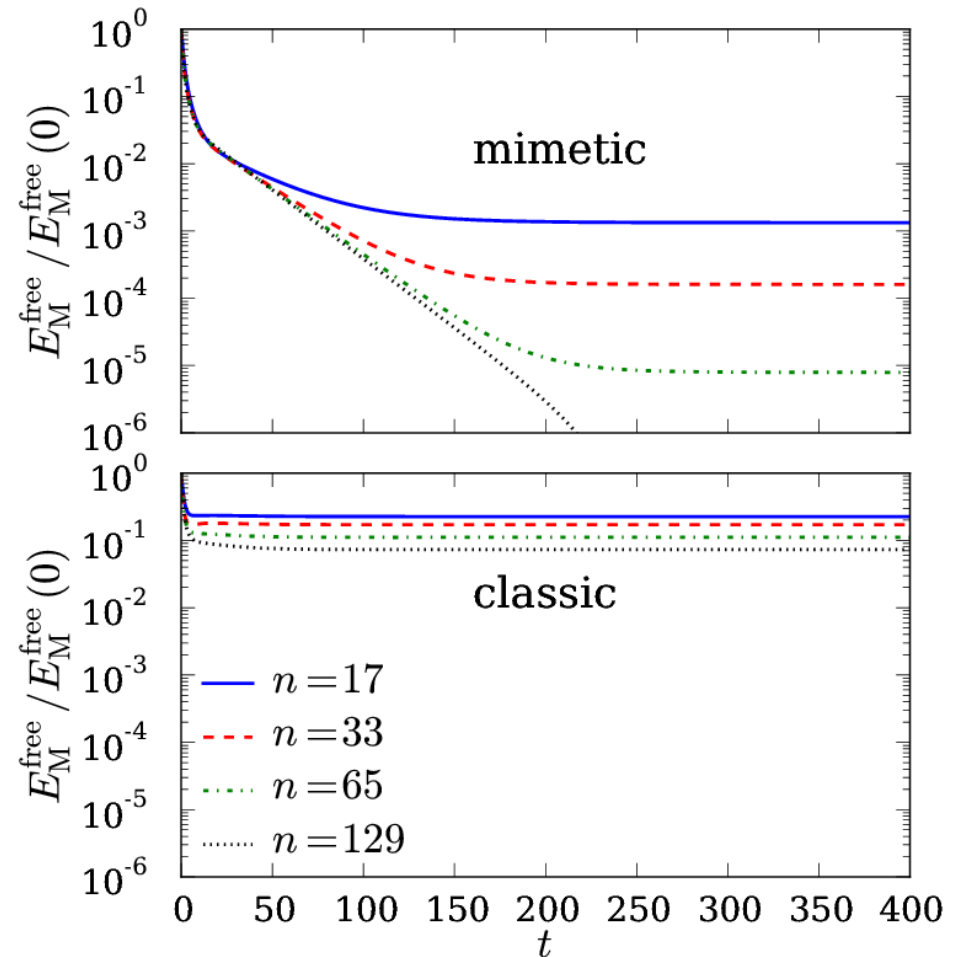
$t = 0$



$t = 10$

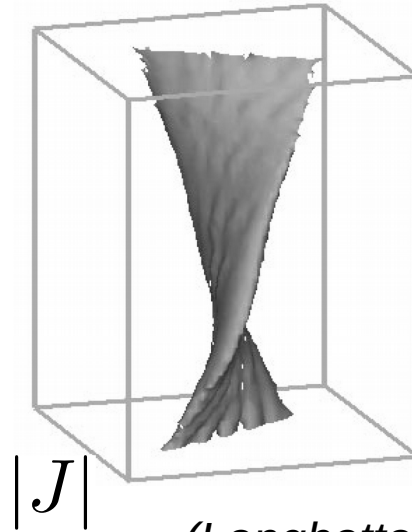
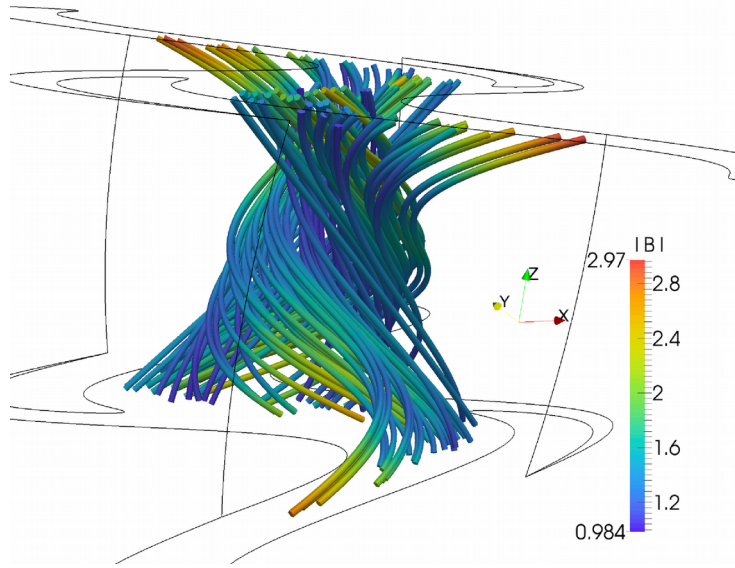


$t = 430$

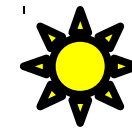
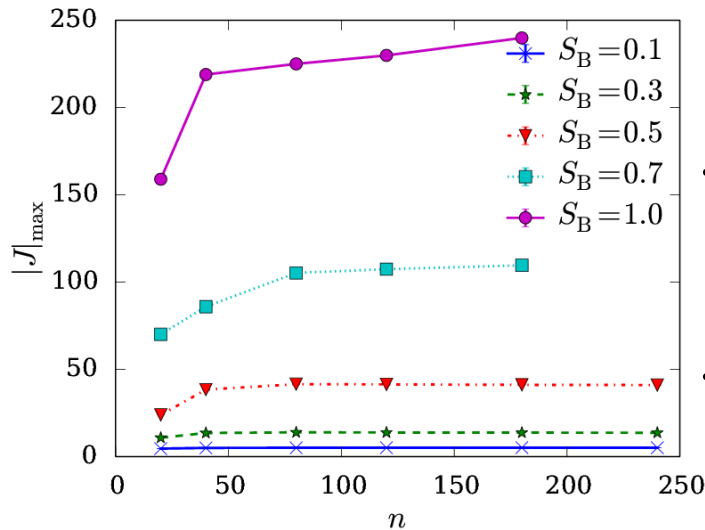
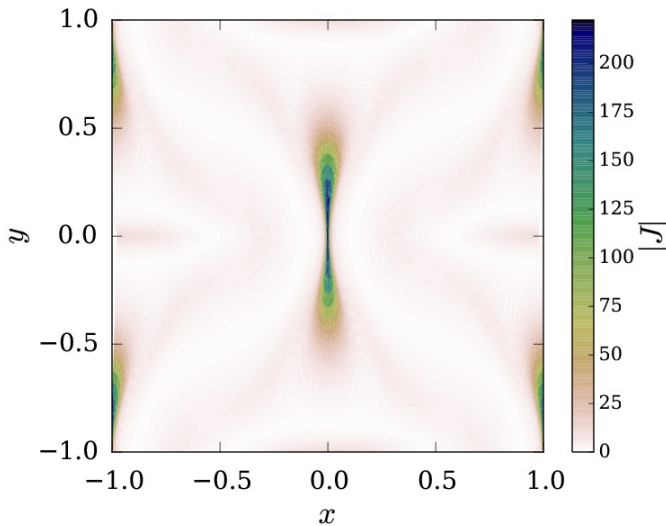
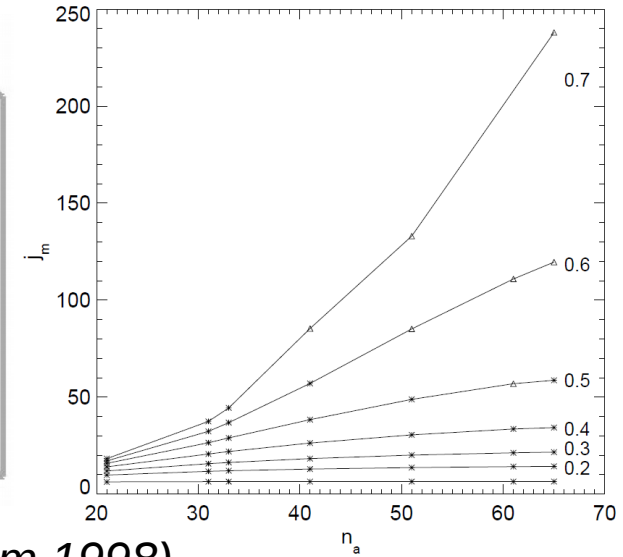


(Candelaresi et al. 2014)

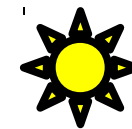
Distorted Magnetic Fields



$|J|$
(Longbottom 1998)



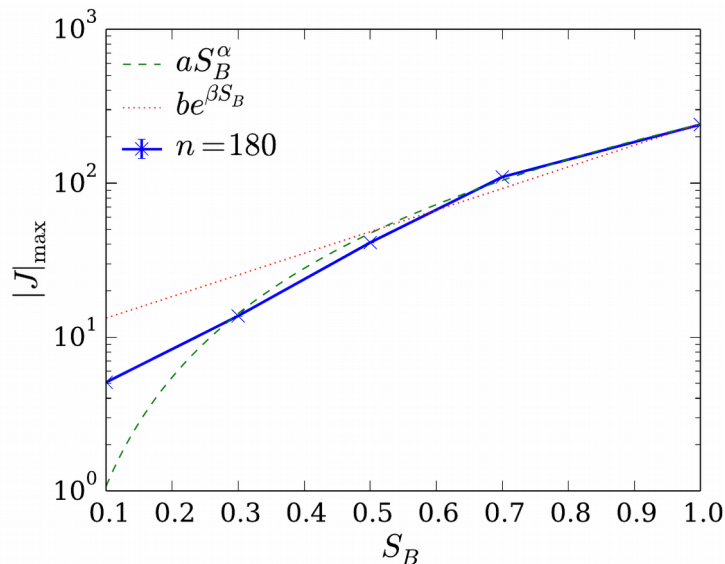
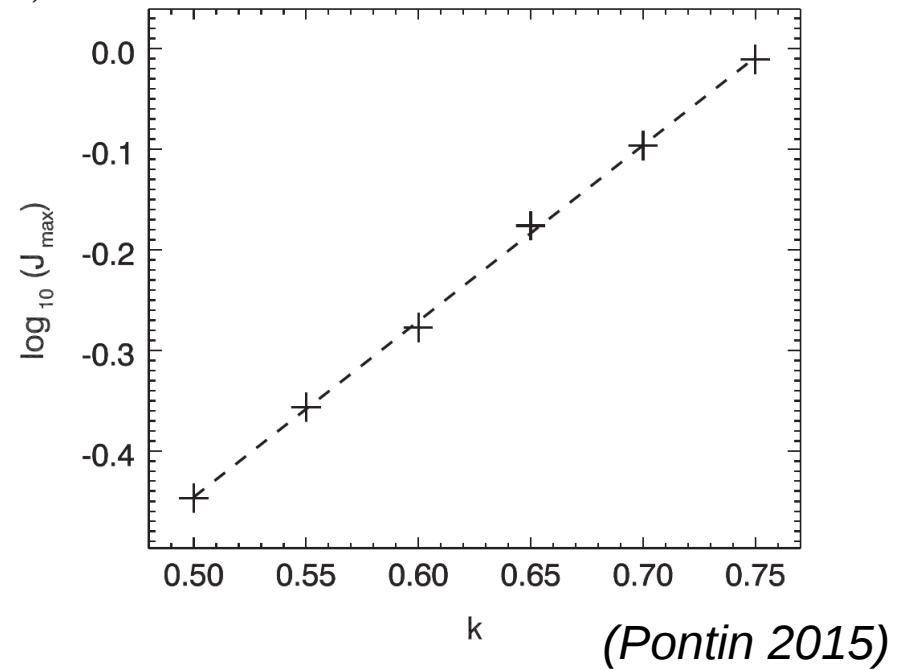
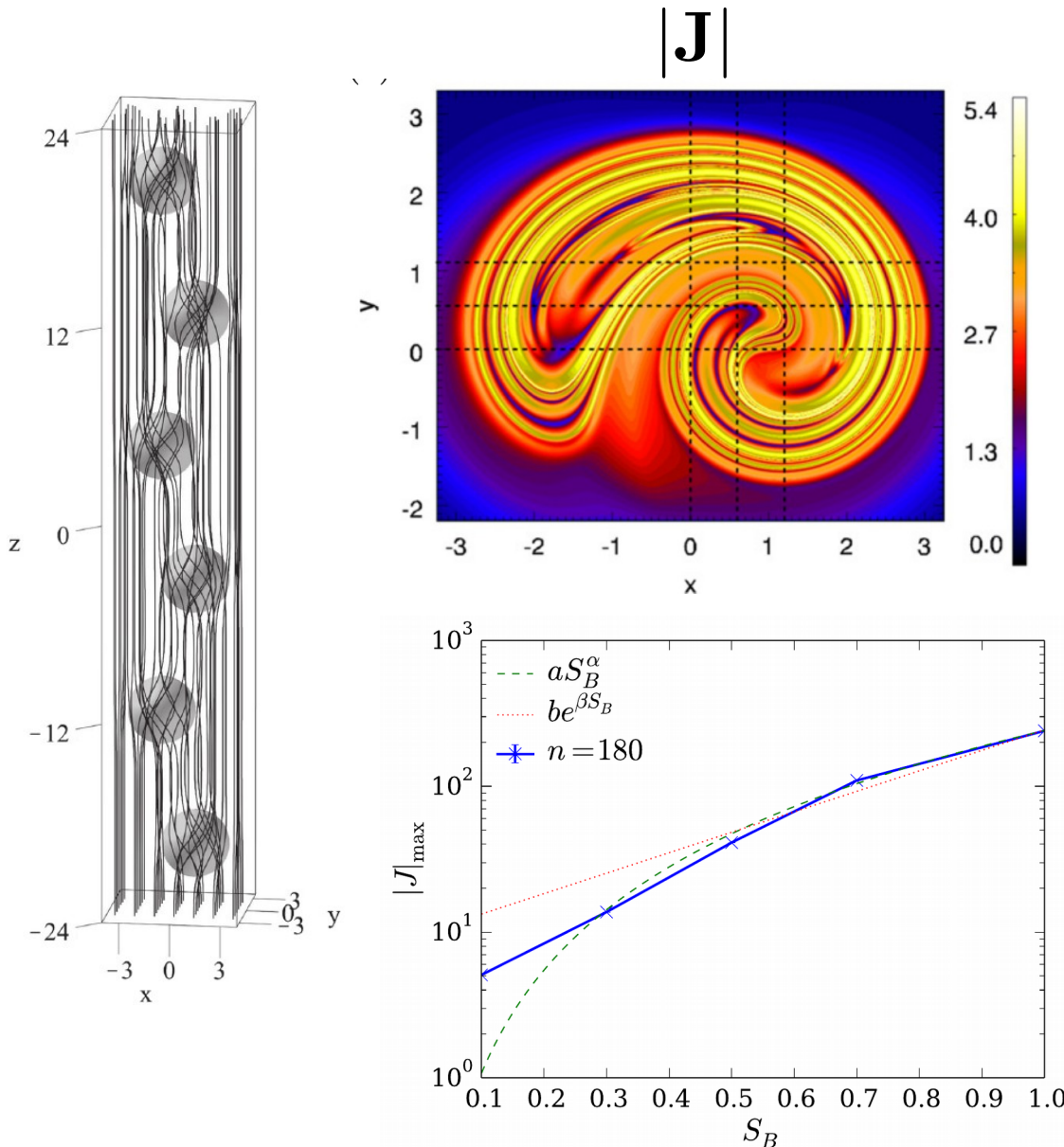
resolved current concentrations



shear leads to strong currents

(Candelaresi et al. 2015)

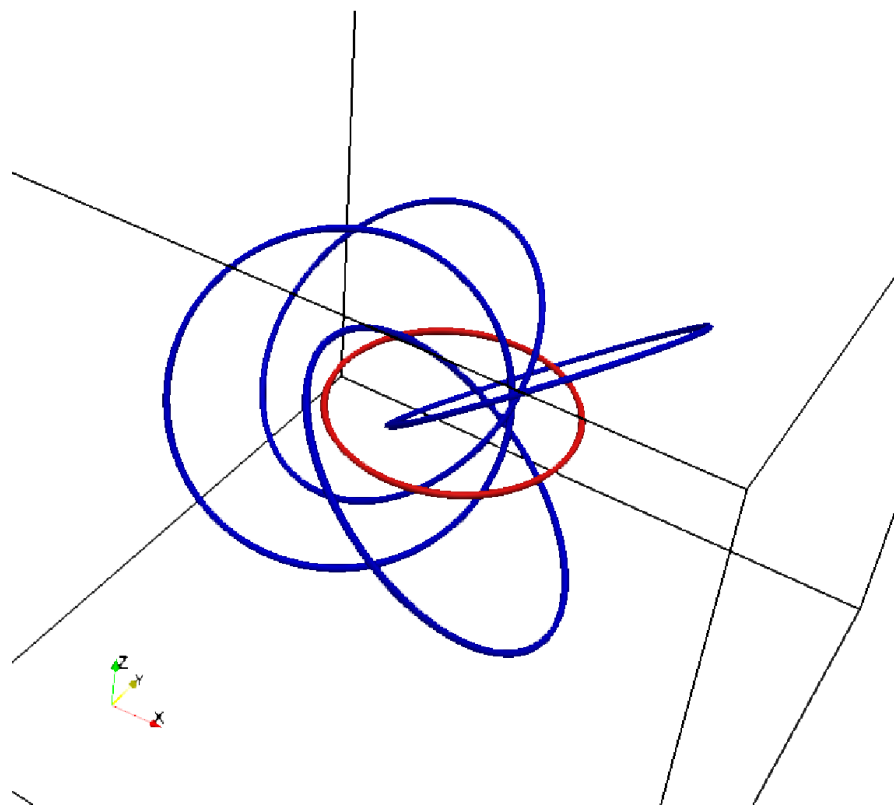
Exponential Increase in Current



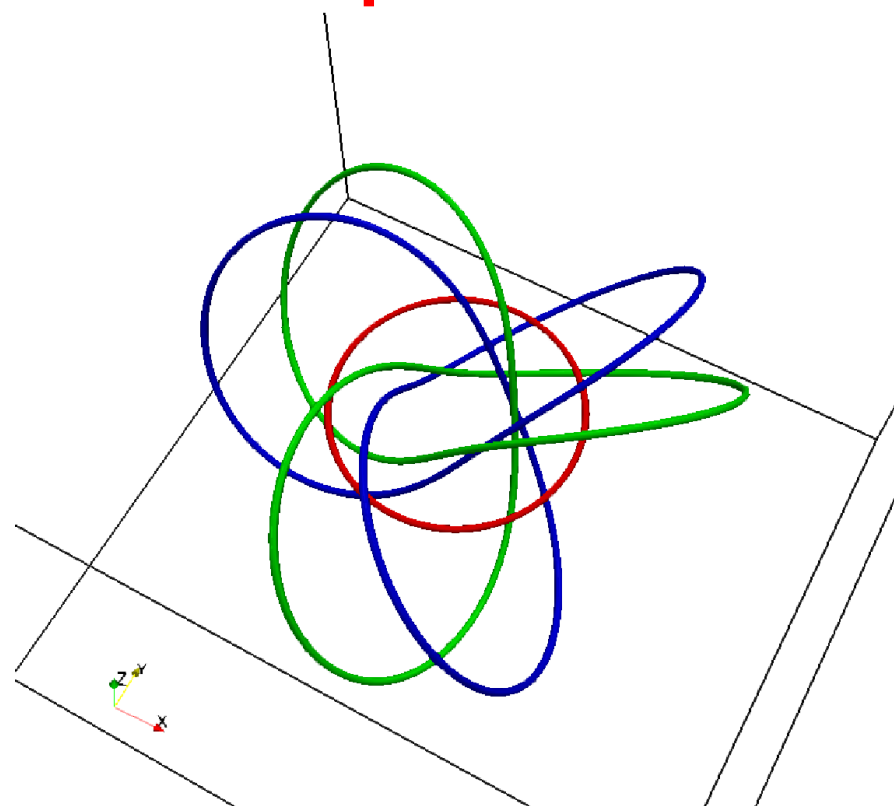
(Candelaresi 2015)

- lengths decrease exp. with complexity
- current increases exp. with complexity

Ideal Relaxation of the Hopf Fibration



$$(\omega_1, \omega_2) = (1, 1)$$



$$(\omega_1, \omega_2) = (3, 2)$$

$$\mathbf{B}_{\omega_1, \omega_2} = \frac{4\sqrt{s}}{\pi(1+r^2)^3 \sqrt{\omega_1^2 + \omega_2^2}} \begin{pmatrix} 2(\omega_2 y - \omega_1 x z) \\ -2(\omega_2 x + \omega_1 y z) \\ \omega_1(-1 + x^2 + y^2 - z^2) \end{pmatrix}$$

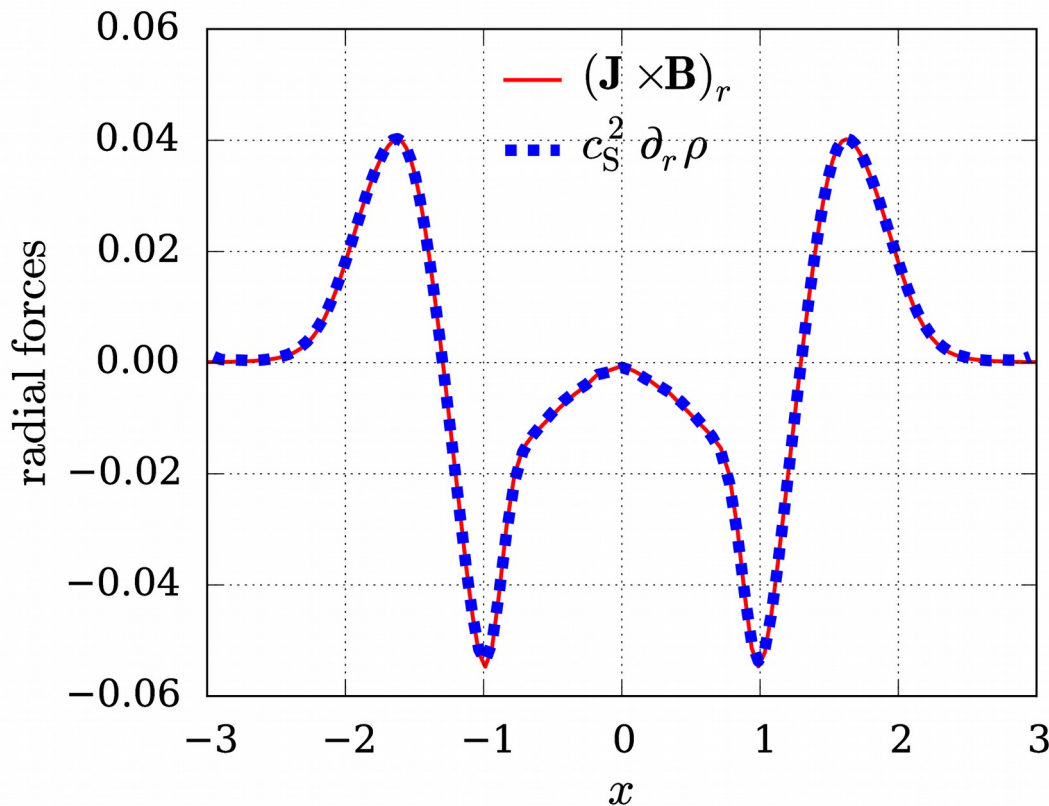
Ideal Relaxation of the Hopf Fibration

$$\frac{\partial \mathbf{B}}{\partial t} - \nabla \times (\mathbf{u} \times \mathbf{B}) = \mathbf{0}$$


Ideal MHD.

$$d\mathbf{u}/dt = (\mathbf{J} \times \mathbf{B} - \nu \mathbf{u} - \beta \nabla \rho) / \rho$$

Momentum equation with velocity damping.



 Force balanced field
(Berg-Smiet et al. 2017).

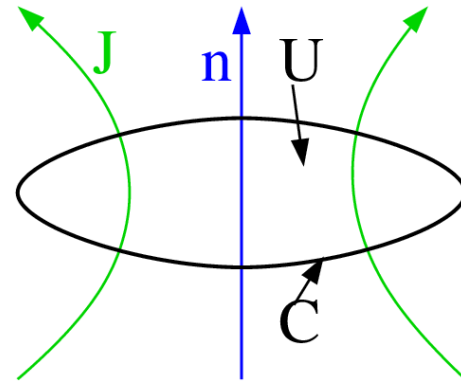
 Grad-Shafranov solutions
(Shafranov, 1966).

Conclusions

- Topology preserving relaxation of magnetic fields.
- GLEMUR: (<https://github.com/SimonCan/glemur>)
- Current concentrations not singular.
- Current increases strongly with field complexity.
- Hopf fibrations relax into Grad-Shafranov states in ideal MHD.

Mimetic Numerical Operators

$$I = \int_U \mathbf{J} \cdot \mathbf{n} \, dS = \oint_C \mathbf{B} \cdot d\mathbf{r}$$



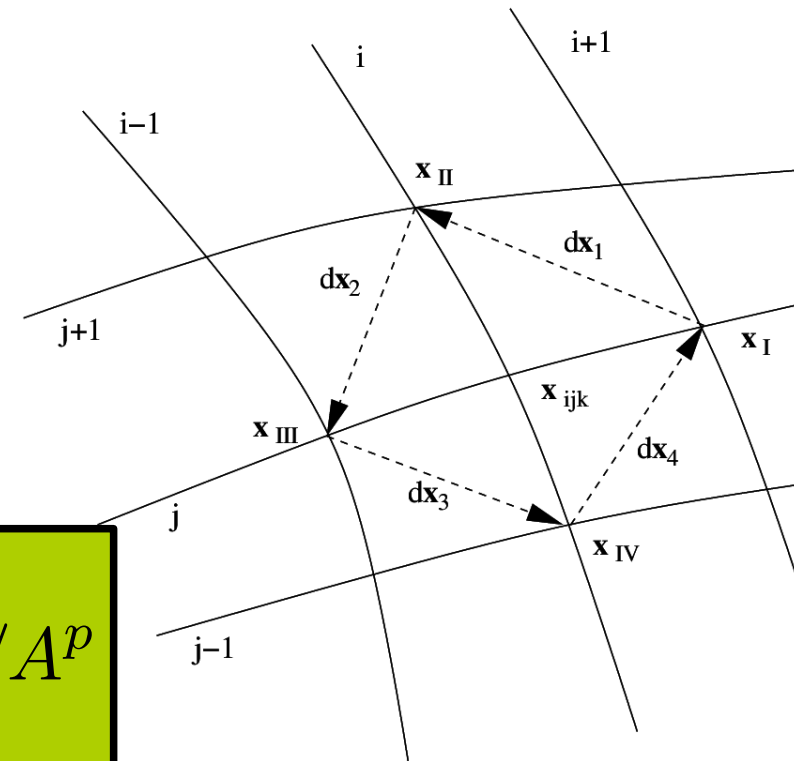
Discretized:

$$I \approx \mathbf{J}(\mathbf{X}_{ijk}) \cdot \mathbf{n} A = \sum_{r=1}^4 \mathbf{B}_r \cdot d\mathbf{x}_r$$

$$\mathbf{J}(\mathbf{X}_U) \approx \mathbf{J}(\mathbf{X}_{ijk}), \quad \mathbf{X}_U \in U$$

3 planes will give 3 l.i. normal vectors:

$$I^p = \mathbf{J}(\mathbf{X}_{ijk}) \cdot \mathbf{n}^p = \sum_{r=1}^4 \mathbf{B}_r^p \cdot d\mathbf{x}_r / A^p$$



Inversion yields \mathbf{J} with $\nabla \cdot \mathbf{J} = 0$.

(Hyman, Shashkov 1997)

Quality Parameters

For a force-free field: $\nabla \times \mathbf{B} = \alpha \mathbf{B}$

$$\mathbf{B} \cdot \nabla \alpha = 0$$

➔ Force-free parameter does not change along field lines.

➔ Measure the change of $\alpha^* = \frac{\mathbf{J} \cdot \mathbf{B}}{\mathbf{B}^2}$ along field lines:

$$\epsilon^* = \max_{i,j} \left(a_r \frac{\alpha^*(\mathbf{X}_i) - \alpha^*(\mathbf{X}_j)}{|\mathbf{X}_i - \mathbf{X}_j|} \right); \quad \mathbf{X}_i, \mathbf{X}_j \in s_\alpha$$

Particular field line: $s_\alpha = \{(0, 0, Z) : Z \in [-L_z/2, L_z/2]\}$

Quality Parameters

Deviation from the expected relaxed state:

$$\sigma_{\mathbf{x}} = \sqrt{\frac{1}{N} \sum_{ijk} (\mathbf{x}(\mathbf{X}_{ijk}) - \mathbf{x}_{\text{relax}}(\mathbf{X}_{ijk}))^2}$$

$$\sigma_{\mathbf{B}} = \sqrt{\frac{1}{N} \sum_{ijk} (\mathbf{B}(\mathbf{X}_{ijk}) - \mathbf{B}_{\text{relax}}(\mathbf{X}_{ijk}))^2}$$

Free magnetic energy:

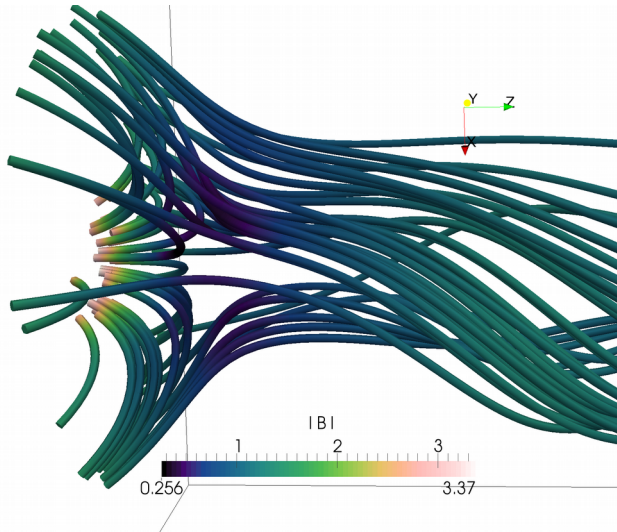
$$E_{\text{M}}^{\text{free}} = E_{\text{M}} - E_{\text{M}}^{\text{bkg}}$$

$$E_{\text{M}} = \int_V \mathbf{B}^2 / 2 \, dV \quad \mathbf{B}^{\text{bkg}} = B_0 \hat{e}_z$$

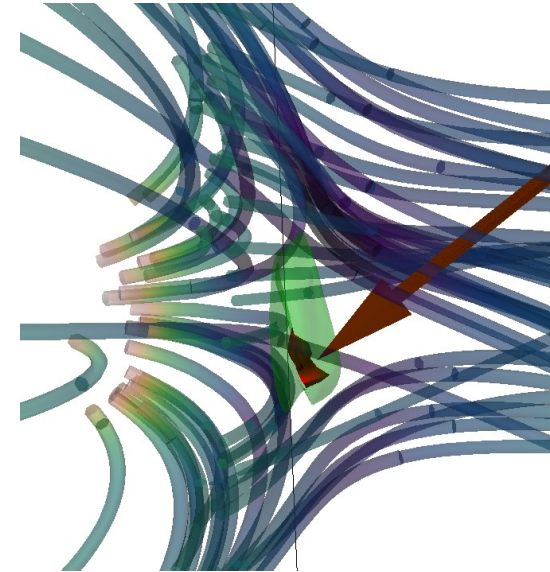
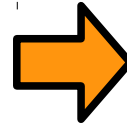
Magnetic Nulls

Singular current sheets observed at magnetic nulls ($B = 0$)

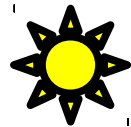
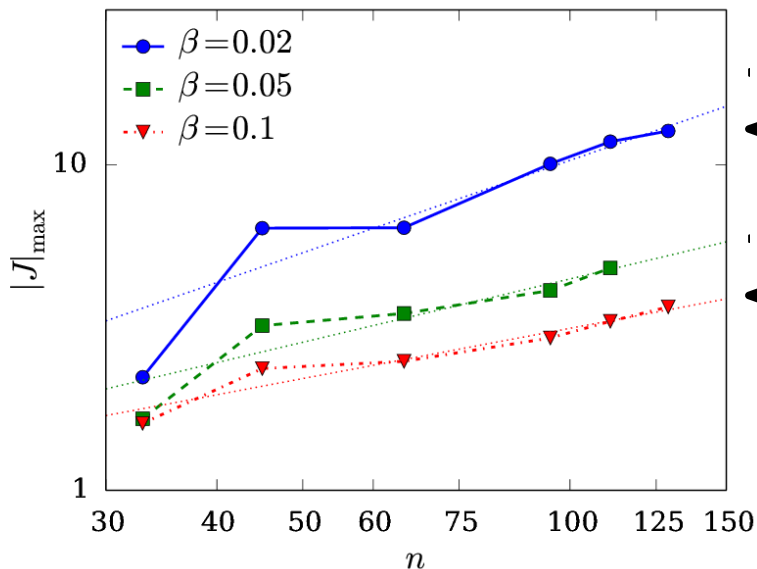
(Syrovatskiĭ 1971; Pontin & Craig 2005; Fuentes-Fernández & Parnell 2012, 2013; Craig & Pontin 2014)



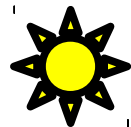
$$\mathbf{u} = \mathbf{J} \times \mathbf{B}$$



$$\mathbf{u} = \mathbf{J} \times \mathbf{B} - \beta \nabla \rho$$



singular current sheets at magnetic nulls

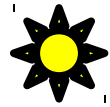


Pressure cannot balance singularity.

Code Details



written in C++



6th order Runge-Kutta time stepping



running in GPUs



periodic and line-tied boundaries



VTK data format



post processing routines in Python

```
// compute the norm of JxB/B**2
__global__ void JxB_B2(REAL *B, REAL *J, REAL *JxB_B2, int dimX, int dimY, int dimZ) {
    int i = threadIdx.x + blockDim.x * blockIdx.x;
    int j = threadIdx.y + blockDim.y * blockIdx.y;
    int k = threadIdx.z + blockDim.z * blockIdx.z;
    int p = threadIdx.x;
    int q = threadIdx.y;
    int r = threadIdx.z;
    int l;
    REAL B2;

    // shared memory for faster communication, the size is assigned dynamically
    extern __shared__ REAL s[];
    REAL *Bs = s; // magnetic field
    REAL *Js = &s[3 * dimX * dimY * dimZ]; // electric current density
    REAL *JxBs = &Js[3 * dimX * dimY * dimZ]; // JxB

    // copy from global memory into shared memory
    if ((i < dev_p.nx) && (j < dev_p.ny) && (k < dev_p.nz)) {
        for (l = 0; l < 3; l++) {
            Bs[l + p*3 + q*dimX*3 + r*dimX*dimY*3] = B[l + (i+1)*3 + (j+1)*(dev_p.nx+2)*3 + (k+1)*(dev_p.nx+2)*(dev_p.ny+2)*3];
            Js[l + p*3 + q*dimX*3 + r*dimX*dimY*3] = J[l + i*3 + j*dev_p.nx*3 + k*dev_p.nx*dev_p.ny*3];
        }

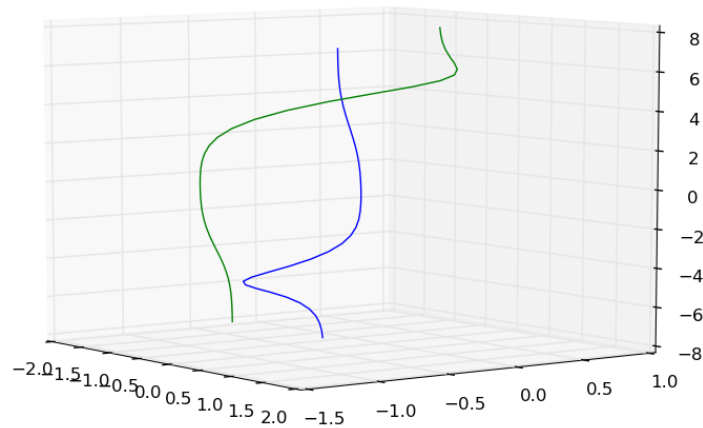
        cross(&Js[0 + p*3 + q*dimX*3 + r*dimX*dimY*3],
              &Bs[0 + p*3 + q*dimX*3 + r*dimX*dimY*3],
              &JxBs[0 + p*3 + q*dimX*3 + r*dimX*dimY*3]);

        B2 = dot(&Bs[0 + p*3 + q*dimX*3 + r*dimX*dimY*3], &Bs[0 + p*3 + q*dimX*3 + r*dimX*dimY*3]);

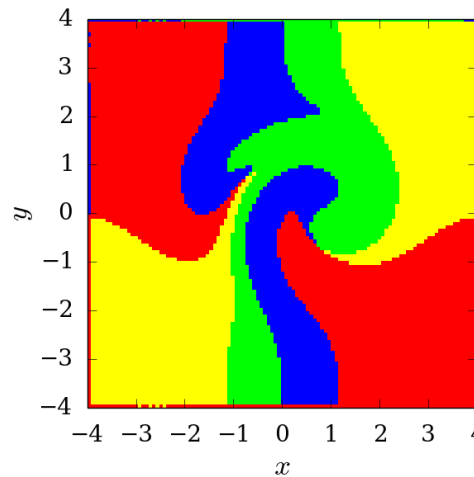
        // return result into global memory
        JxB_B2[i + j*dev_p.nx + k*dev_p.nx*dev_p.ny] = norm(&JxBs[0 + p*3 + q*dimX*3 + r*dimX*dimY*3])/B2;
    }
}
```

Post-Processing

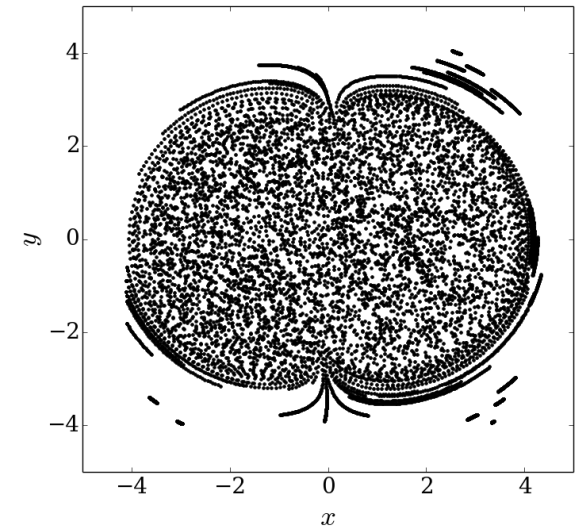
streamlines



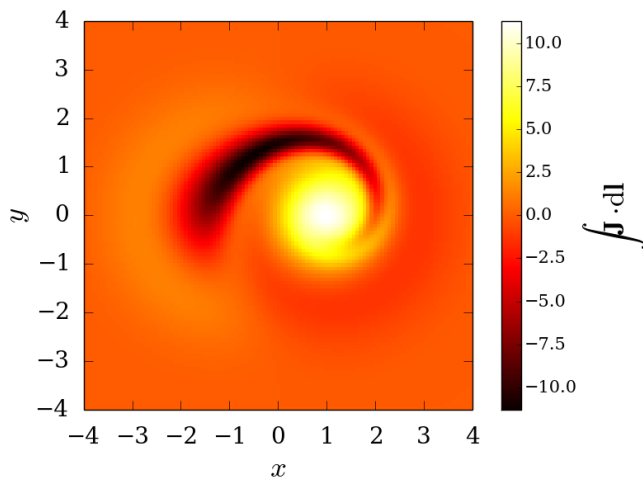
field line mapping



Poincaré maps



line integration



save and read as vtk file

```
s0 = gm.streamInit(tol = 0.01)
```

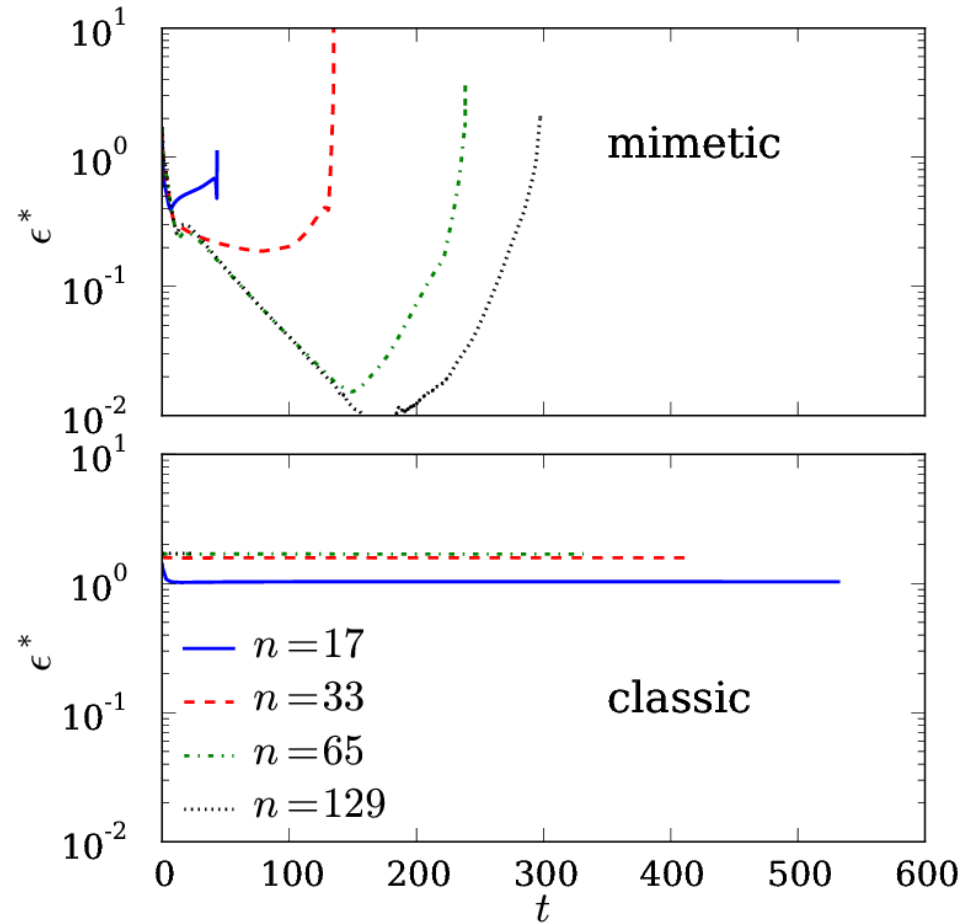
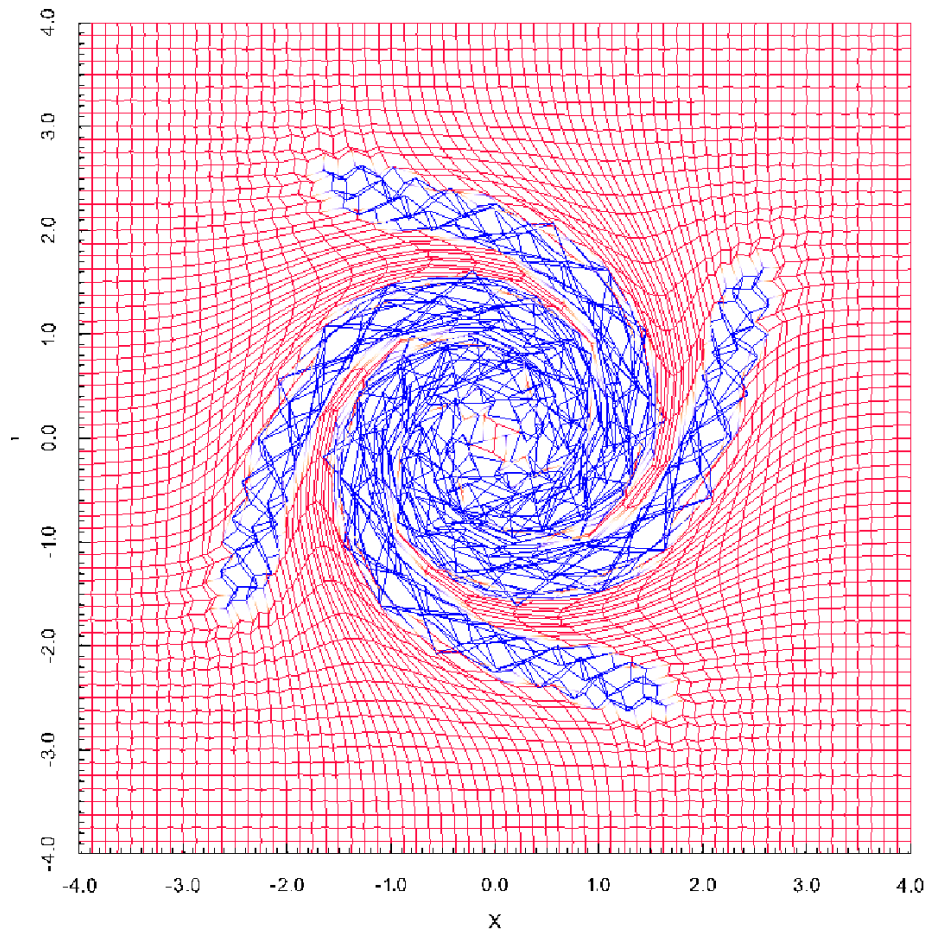


```
stream.vtk
```



```
sr = gm.readStream()
```

Limitations



red: convex
blue: concave

For concave cells the method becomes unstable.
But: results before crash better than classic method.