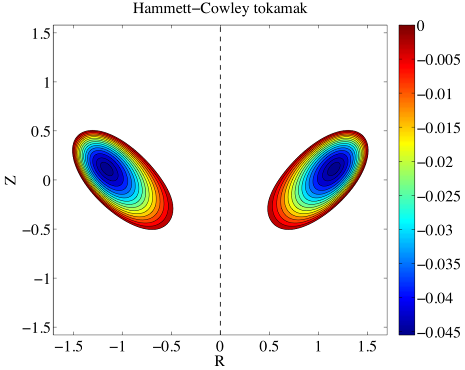
A fast high-order solver for the Grad-Shafranov equation



Andras Pataki, CIMS
with L. Greengard (CIMS), A. Cerfon (MIT), J. Freidberg (MIT)

Grad-Shafranov equation as Poisson's equation

$$R\frac{\partial}{\partial R}\left(\frac{1}{R}\frac{\partial\Psi}{\partial R}\right)+\frac{\partial^2\Psi}{\partial Z^2}=-\mu_0R^2\frac{dp}{d\Psi}-F\frac{dF}{d\Psi}$$

$$\Psi = Cst \qquad \text{on plasma boundary}$$

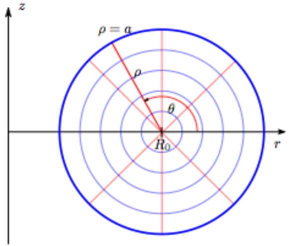
- Change the unknown function: $\Psi = \sqrt{R}U$

$$\frac{\partial^2 U}{\partial R^2}+\frac{\partial^2 U}{\partial Z^2}=\frac{3}{4}\frac{U}{R^2}-\mu_0R\frac{dp}{dU}-\frac{1}{2R}\frac{dF^2}{dU}$$

$$\sqrt{R}U = Cst \qquad \text{on plasma boundary}$$

- The left-hand side is the 2-D Laplacian
- We want the first and second derivatives of U with very good accuracy$\Rightarrow$ Spectral methods

Fast Poisson solver on a circle



$$\Delta u(\rho,\theta) = f(\rho,\theta)$$
$$u(a,\theta) = g(\theta)$$

- Write data and solution as Fourier series (use FFT)

$$u(\rho,\theta) = \sum \hat{u}_n(\rho)e^{in\theta}\ ,\ \ f(\rho,\theta) = \sum \hat{f}_n(\rho)e^{in\theta}\ ,\ \ g(\theta) = \sum \hat{g}_n e^{in\theta}$$

- Plugging into Poisson's equation, we get mode-by-mode ODE

$$\hat{u}_n''(\rho) + \frac{1}{\rho}\hat{u}_n'(\rho) - \frac{n^2}{\rho^2}\hat{u}_n(\rho) = \hat{f}_n(\rho)$$
$$\hat{u}_n(a) = \hat{g}_n$$

Fast Poisson solver on a circle

- Using Green's functions to construct particular solution (that does not satisfy B.C.)

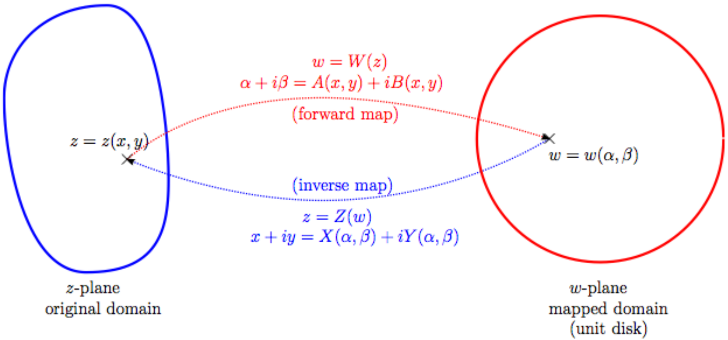
$$\begin{aligned}\hat{u}_n^{part}(\rho) &= \int_0^\infty G_n(\rho, s) f(s) ds \\ &= -\frac{1}{2n}\left[\rho^{-n}\int_0^\rho s^{n+1} f(s) ds + \rho^n \int_\rho^\infty s^{-n+1} f(s) ds\right]\end{aligned}$$

- Piecewise Chebyshev/Legendre grid crucial for high order
- To match the B.C., add homogeneous correction:

$$\hat{u}_n(\rho) = \hat{u}_n^{hom}(\rho) + \hat{u}_n^{part}(\rho) \qquad \text{with } \hat{u}_n^{hom}(\rho) = c_n \left(\frac{\rho}{a}\right)^n$$

- Boundary condition yields explicit condition $c_n = \hat{g}_n - \hat{u}_n^{part}$
- Major advantage of Green's function method: differentiation

Conformal mapping

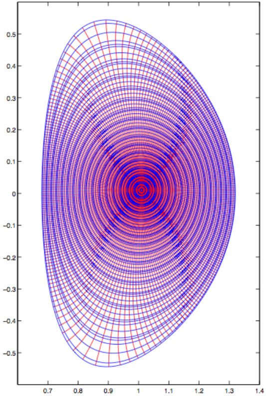


$$\Delta u(x,y) = f(x,y) \quad \rightarrow \quad \Delta v(\alpha,\beta) = f(X(\alpha,\beta), Y(\alpha,\beta)) \cdot \left|\frac{dZ}{dw}\right|^2$$

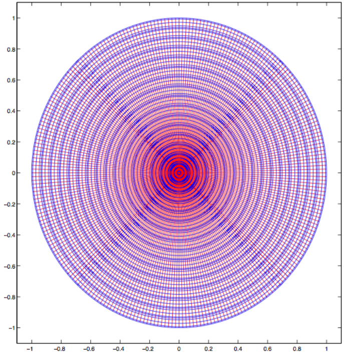
$$u|_{\partial\Omega} = g(x,y) \quad \rightarrow \quad v|_{\partial\Omega} = g(X(\alpha,\beta), Y(\alpha,\beta))$$

- W computed with the Kerzman-Stein integral equation
- Problem with crowding for large aspect ratio

Conformal mapping

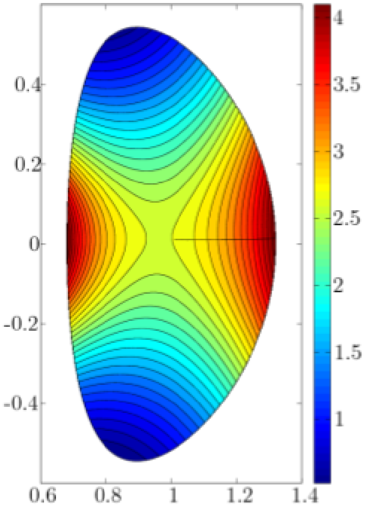


z-plane
original domain



w-plane
mapped domain
(unit circle)

Conformal mapping

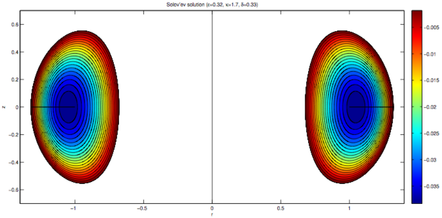


ITER crowding factor: **5**

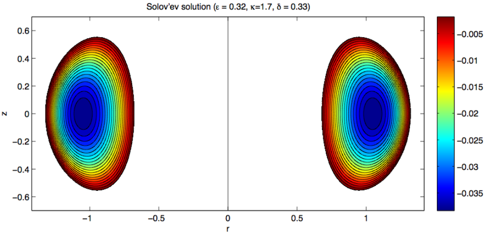
NSTX crowding factor: **20**

ITER – Results

Numerical solution

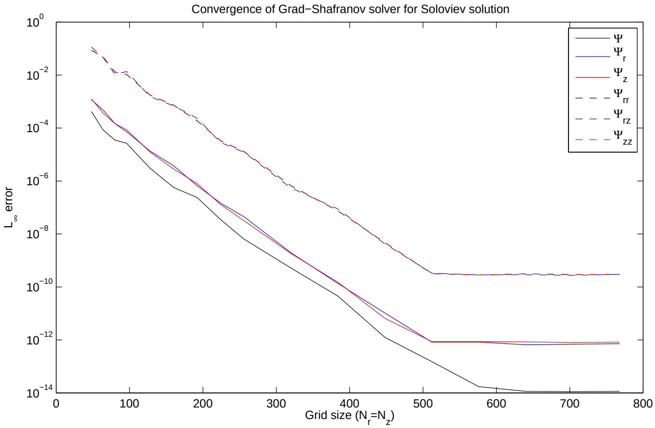


Exact solution



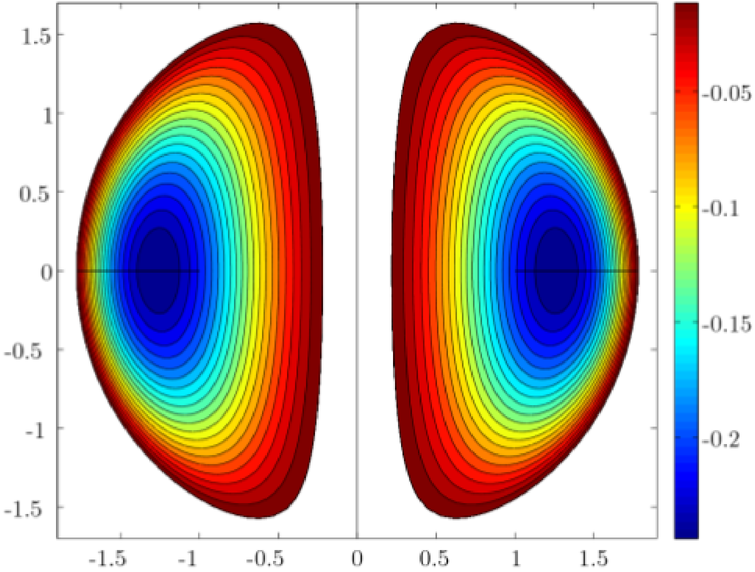
$$\epsilon = 0.32, \kappa = 1.7, \delta = 0.33$$

ITER – Convergence

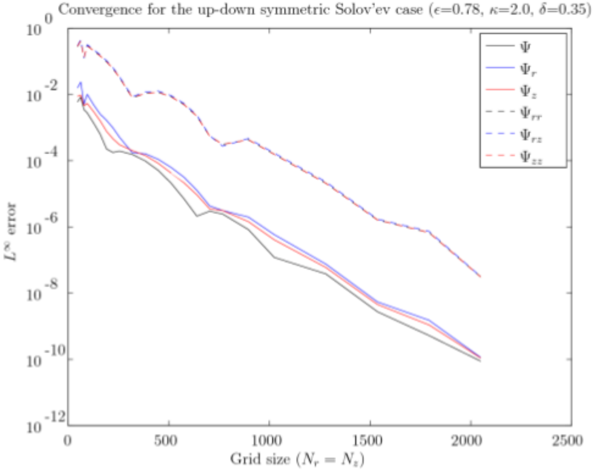


- Exponential convergence – solution almost to machine accuracy
- Good accuracy for derivatives (error due to numerical derivative of boundary conditions)

NSTX – Results



NSTX – Convergence



Results are not as good for STs

Due to:

- Oversampling of 3 compared to ITER
- Domain boundary requires slightly more points to be resolved