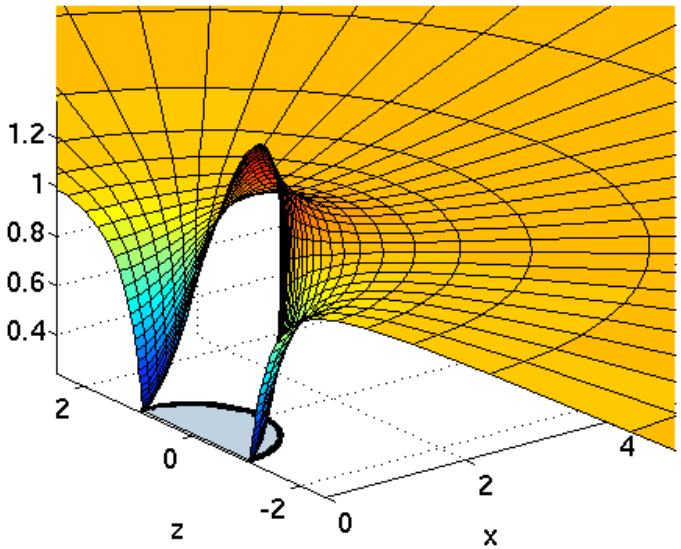
Less is More: Efficient Numerics for Model Physics Problems in Simple Geometries

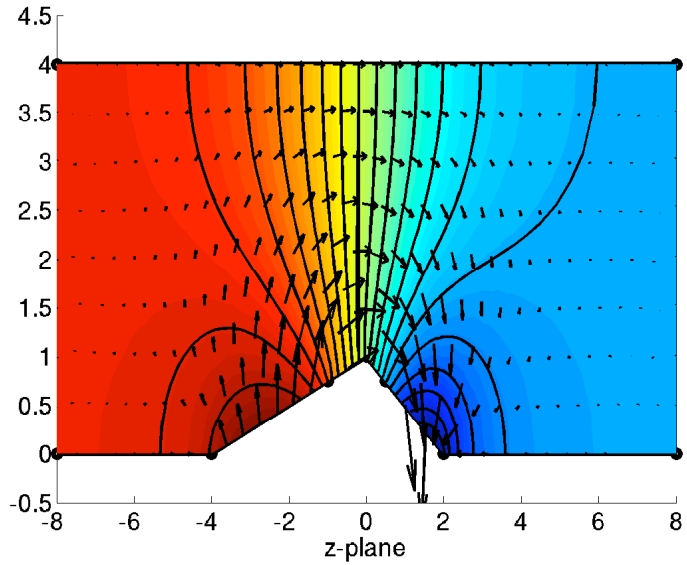
Kevin T. Chu

Advisor: Martin Z. Bazant

Department of Mathematics, MIT

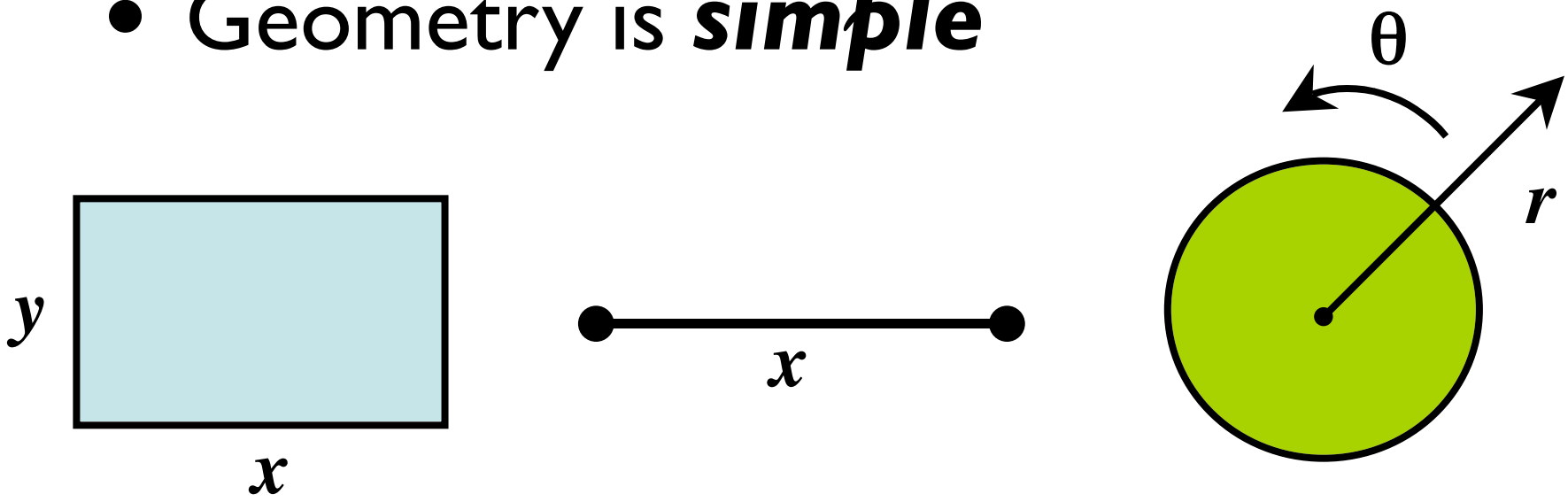
June 23, 2005





Features of Model Problems in Physics & Engineering

- Focus on dominant physics
- Geometry is ***simple***



Pseudospectral Methods

- Advantages
  - ***Very*** computationally efficient:

$$\text{error} \sim \mathrm{O}(p^N), \quad p < 1 \longrightarrow$$

"Infinite" Order

  - Implementation effort similar to FD
- Disadvantages
  - Difficult to handle irregular geometries
  - Effort needed to handle non-smooth solutions

Two Basic Ideas

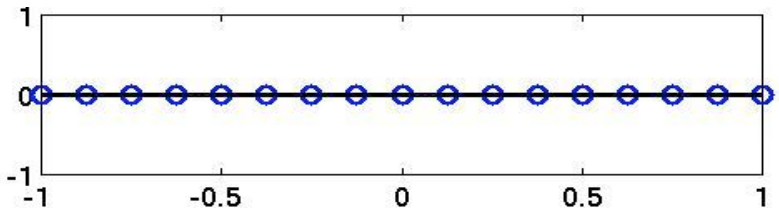
- Choose grid points appropriately based on computational domain
  - e.g. finite interval → Chebyshev grid
  - ↛ uniform grid
- Use ***ALL*** grid points to compute derivatives

Fully dense differentiation matrices
... for *SMALL* N !

Computational Domains & Grids

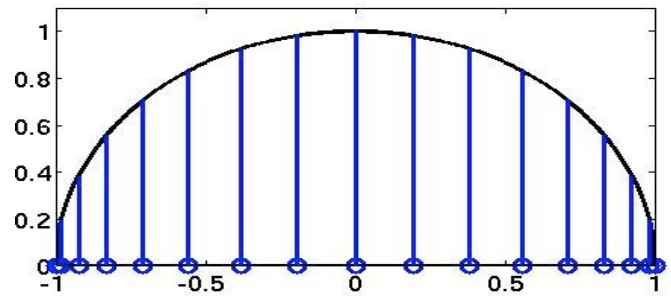
- Periodic

$$x_k = \frac{2k - N}{N}$$



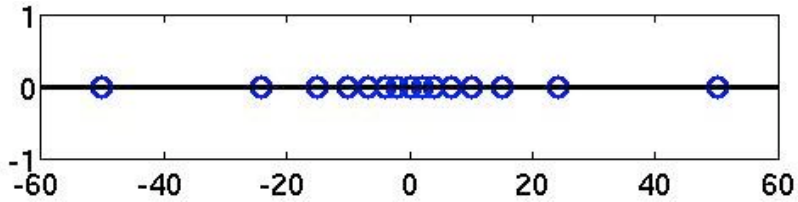
- Finite, non-periodic

$$x_k = \cos(k\pi/N)$$



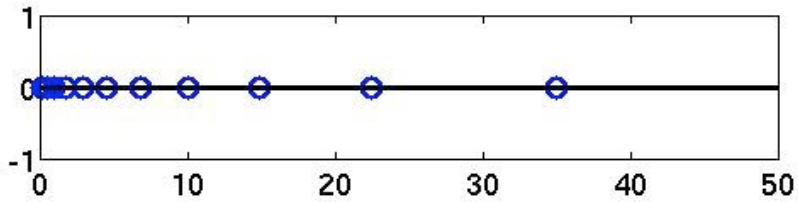
- Infinite

$$y_k = \frac{Lx_k}{\sqrt{1 - x_k^2}}$$



- Semi-infinite

$$y_k = L\left(\frac{1 - x_k}{1 + x_k}\right)$$



From FD to PS

1. Uniform grid ⟶ pseudospectral grid
2. Low-order discrete derivative
   ⟶ pseudospectral differentation matrix
3. Use standard numerical methods
   (*e.g.* Newton iteration, discrete time-stepping)

> "Inside every low-order program is a high order algorithm waiting to burst free!"
> – J.P. Boyd

Direct Matrix Method for Computing Exact Jacobians

- Write discretized differential equations ***explicitly*** in terms of differentation matrices
- Examples:

$$\frac{\partial^2 \phi}{\partial x^2} = \rho \longrightarrow D^2\phi = \rho$$

$$(\nabla\phi)\cdot(\nabla\psi) + \psi\nabla^2\phi = 0 \longrightarrow$$

$$(D_x\phi) .* (D_x\psi) + (D_y\phi) .* (D_y\psi) + \psi .* (L\phi) = 0$$

Direct Matrix Method for Computing Exact Jacobians

- Simple differentiation rules to compute ***exact*** Jacobian

$$\frac{\partial}{\partial u}(A * u) = A$$

$$\frac{\partial f(u)}{\partial u} = \mathrm{diag}\left[f'(u)\right]$$

$$\frac{\partial}{\partial u}\left[A * f(u)\right] = A * \mathrm{diag}\left[f'(u)\right]$$

$$\frac{\partial}{\partial u}\left[f(u) .* g(u)\right] = \mathrm{diag}\left[g(u)\right] * \frac{\partial f}{\partial u} + \mathrm{diag}\left[f(u)\right] * \frac{\partial g}{\partial u}$$

Less tedious and less error-prone Jacobian calculation!

Direct Matrix Method for Computing Exact Jacobians

- Example

$$F(u) = u'' + \sin(2u)u'$$

$$F(u) = D^2 * u + \sin(2u).* (D * u)$$

$$\frac{\partial F}{\partial u} =$$

$$D^2 + \mathrm{diag}[\sin(2u)] * D + 2\mathrm{diag}[D * u] * \mathrm{diag}[\cos(2u)]$$

Making the Most of MATLAB

- Advantages
  - Many useful built-in functions
  - Easy visualization
- Disadvantage: considered *slow*
  - Solution: mixed-language programming (*i.e.* MEX-files)

High-efficiency programming
+
High-performance computing

Electrochemical Thin-Films

- Integro-differential equation

$$\epsilon^2 \left[\frac{d^2 E}{dx^2} - \frac{1}{2} E^3 \right] - (c_0 + 2jx)\, E = 2j$$

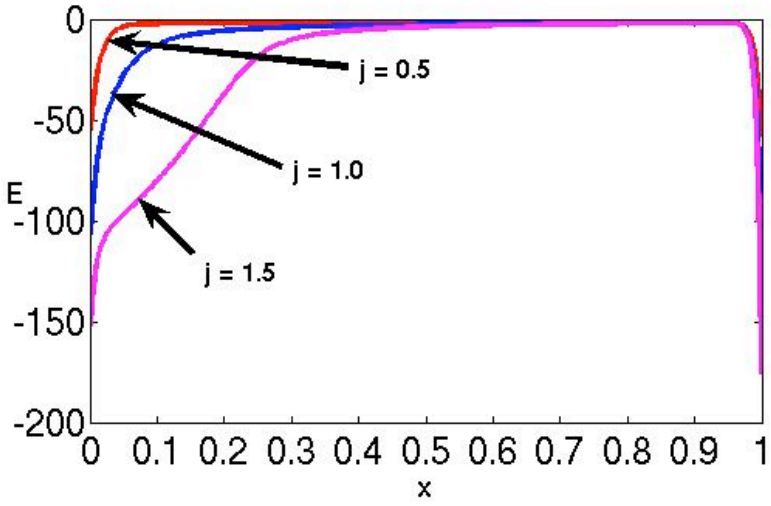
$$c_0 = (1 - j) + \epsilon^2 \left[E(1) - E(0) - \frac{1}{2} \int_0^1 E^2 dx \right]$$

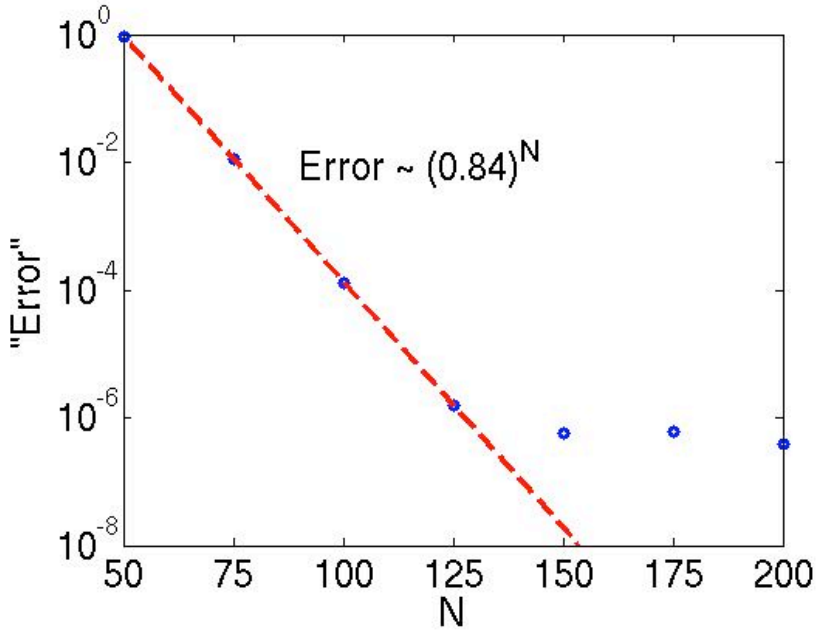
- Nonlinear reaction boundary conditions

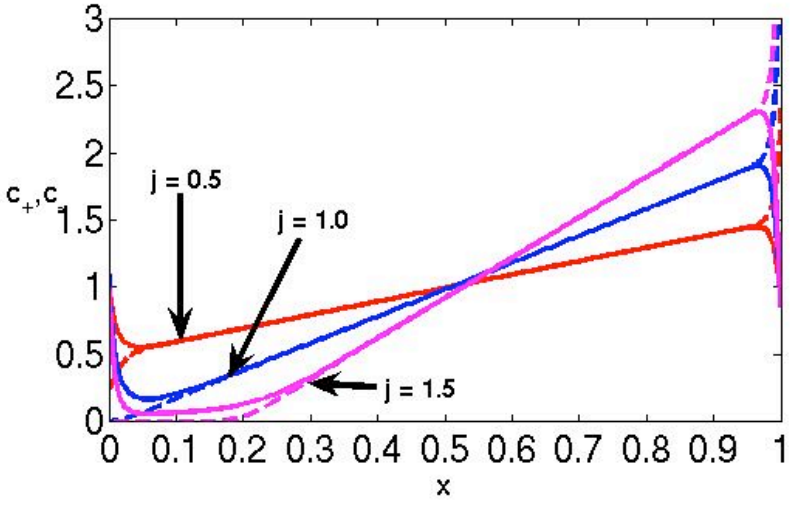
$$k_c c_+ e^{-\alpha_c \delta \epsilon E(0)} - j_r e^{\alpha_a \delta \epsilon E(0)} = j$$

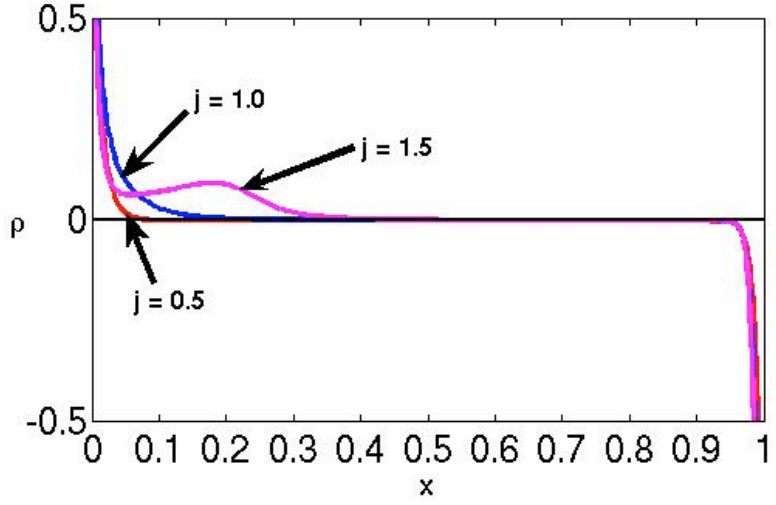
$$-k_c c_+ e^{\alpha_c \delta \epsilon E(1)} + j_r e^{-\alpha_a \delta \epsilon E(1)} = j$$

Electrochemical Thin-Films

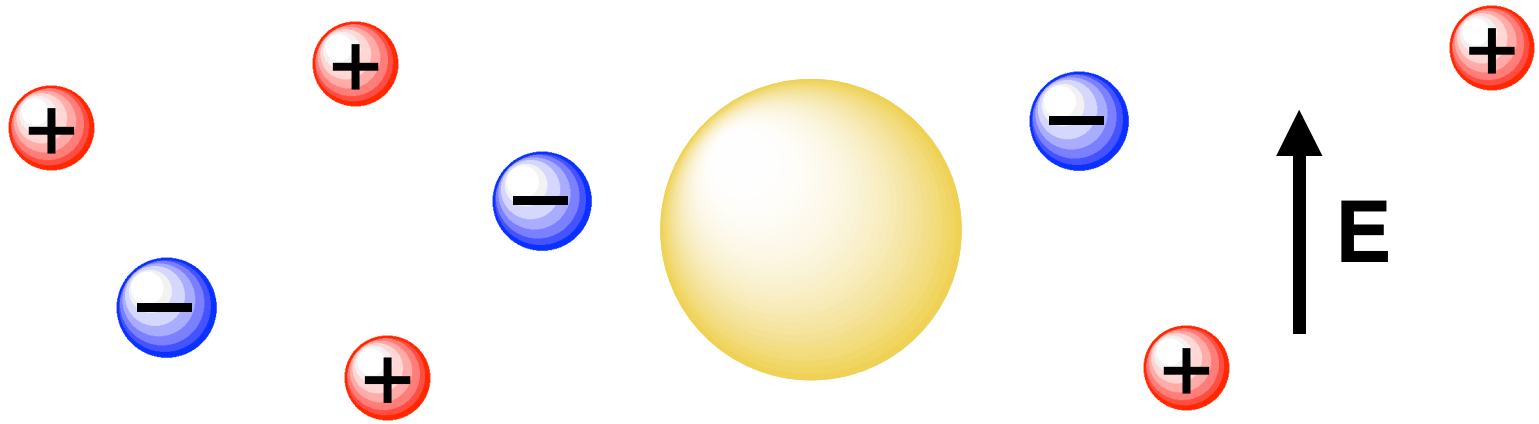








Charging of Metal Colloid Sphere



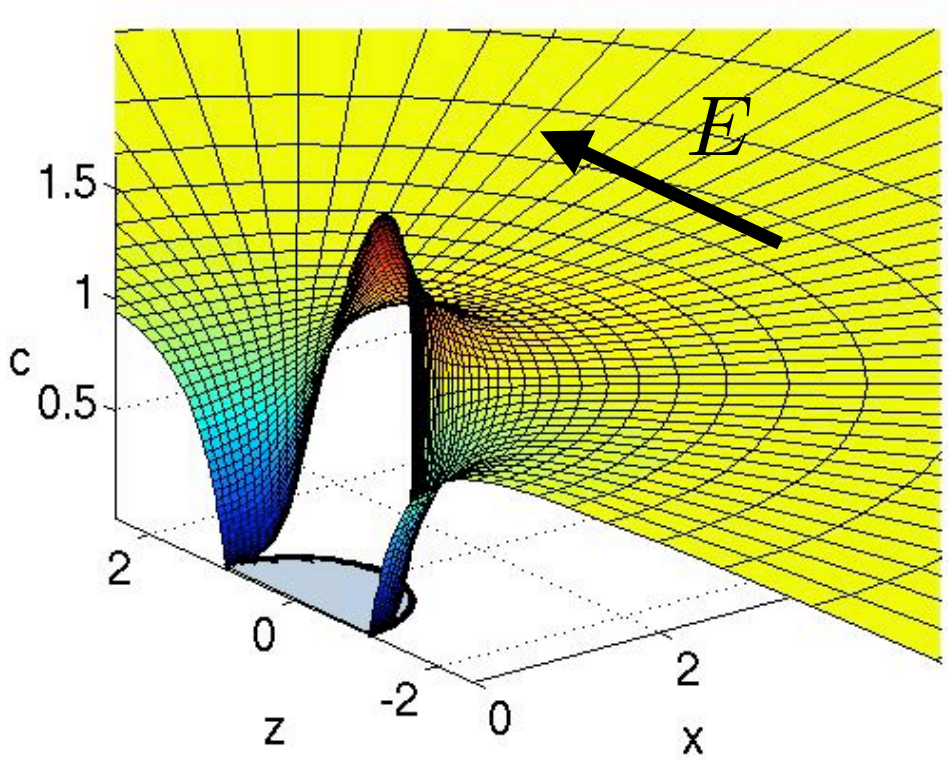
- Governing equations

$$0 = \nabla^2 c \qquad\qquad 0 = \nabla \cdot (c \nabla \phi)$$

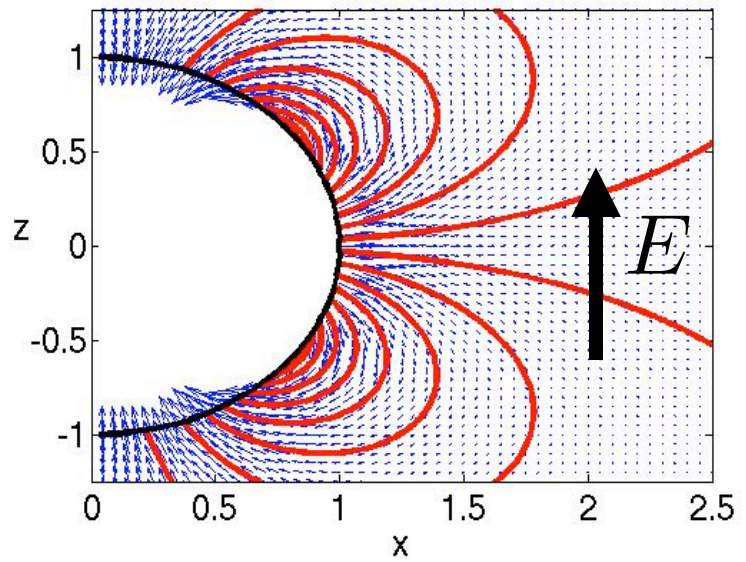
- Boundary conditions

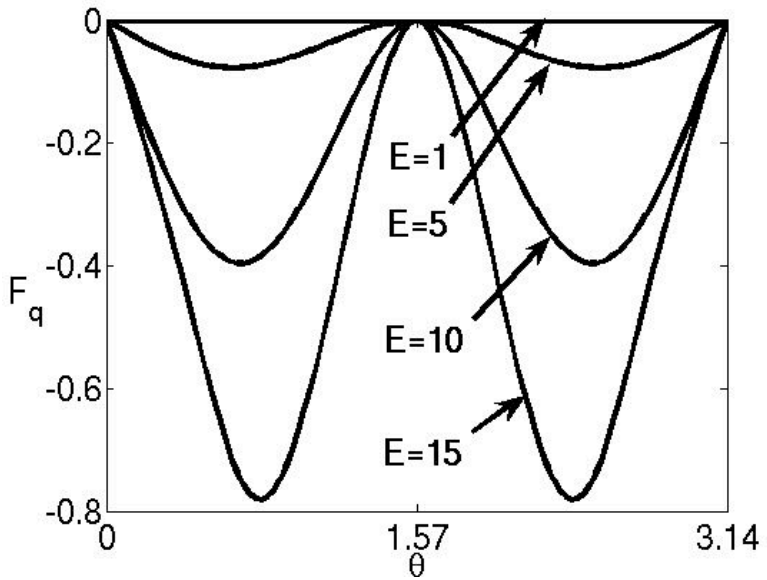
$$F_s^{\pm} = -\frac{\epsilon \left(e^{\mp \zeta/2} - 1\right)}{\sqrt{c}} \left(\nabla_s c \pm 2c \nabla_s \phi\right) \qquad\qquad \zeta + 2\delta(\zeta/2) = -\phi$$

Charging of Metal Colloid Sphere



$$E = 15$$
$$\epsilon = 0.01$$
$$\delta = 1$$

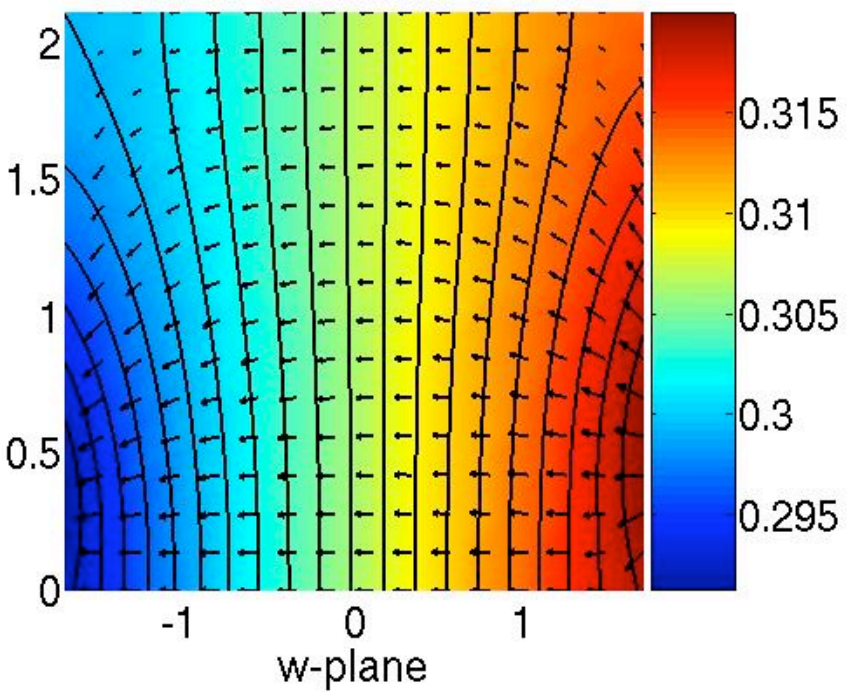


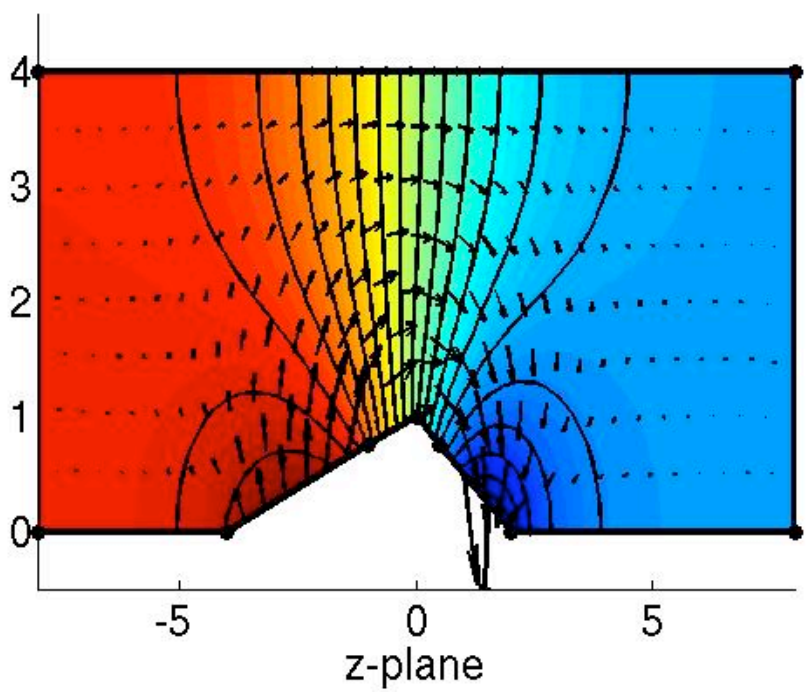


Conformal Mapping for Nernst-Planck Equations

- Steady, electroneutral Nernst-Planck equations are conformally invariant (Bazant 2004)

$$0 = \nabla \cdot (\nabla c_{\pm} \pm c_{\pm} \nabla \phi) \qquad \sum_i z_i c_i = 0$$





High-Performance ***Desktop*** Scientific Computing

- Spectral methods yield ***high*** accuracy at ***low*** cost
- ***Exact*** Jacobians via Direct Matrix Method
- MATLAB + C/Fortran = powerful tool for moderate-sized problems

Acknowledgements

- Advisor: Martin Z. Bazant

- Fellow graduate students
  - Jaehyuk Choi (Quant, BNP Paribas)
  - Per-Olof Persson (Instructor, Dept. of Math, MIT)
  - Boguk Kim

- Funding
  - Computational Science Graduate Fellowship, DoE
  - Center for Material Science and Engineering, MIT

THE END

EXTRA SLIDES

Desktop "Super"-Computers

- Fast CPUs (> 3 GHz)
- Large memories (2 – 4 GB)
- Large on-chip caches (1024 K)

Modern *serial* computers are extraordinarily powerful!