Improved Resistive Wall (R-wall) Algorithm for NIMROD

D. C. Barnes, C. Akcay, J. M. Finn

NIMROD SPRING MEETING
21 April, 2018
• Thanks to Jake for tour of developer res-wall stuff
• We have been working w UW version
  • NIM-TIB?
• Re-did Andie Beccera’s work for our app
• External solver based on DCB’s resistive wall stuff from 2D
• Tested on cylinder
  • Corners
  • R = 0
  • Etc.
• NIMROD implementation for n=1 completed
  • Testing now
External Solver

\[ B_x = \nabla \Phi_M , \nabla^2 \Phi_M = 0 \]
continuous \( B_n = \partial_n \Phi_M \Rightarrow \Phi_M \)

\[ \Rightarrow B_{Tx} = \left( \Phi_M ' , \frac{in}{R} \Phi_M \right) \]

\[ \Rightarrow [ [ B_T ]] \Rightarrow J^* \Rightarrow E_T \Rightarrow \dot{B}_n \]

\[ \Phi_M = \int dS \frac{\sigma(S)}{\| r - r_S \|} \approx D\sigma \]

\[ \partial_n \Phi_M = \int dS \sigma(S) \partial_n \frac{1}{\| r - r_S \|} \approx N\sigma \]

\[ S = DN^{-1} \]
Convergence of boundary solution

Function

Derivative

$y = 69.802x^{1.98}$

$y = 71.909x^{1.618}$

$y = 56.045x^{1.736}$
Resistive wall transformer solution
Improve R-wall algorithm

• Desired – $n$th-order convergence ($n$ up to 6 or so)

  or

• Round-off-ish accuracy

• Components of boundary integral approach
  – Free-space Green’s function
  – Approximate integrals (logarithmic singularity)
  – Matrix inversion and algebra
  – Remainder of algorithm

Nimset time

Nimrod time
R-wall project

- TechX + Tibbar
  - Jake King – PI
  - Dan Barnes – Vacuum solver
  - Cihan Akcay – NIMdevel implementation
  - John Finn – Applied math & benchmarking
PREFICE AND FAST COMPUTATION OF THE GENERAL COMPLETE ELLIPTIC INTEGRAL OF THE SECOND KIND

TOSHIO FUKUSHIMA

Table 7. Coefficients of Taylor Expansion Polynomials of $B(m)$ and $D(m)$: $0.4 < m \leq 0.5$

<table>
<thead>
<tr>
<th>$j$</th>
<th>$B_j$</th>
<th>$D_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.839479570270612971</td>
<td>0.97440243665463869673</td>
</tr>
<tr>
<td>1</td>
<td>0.149916440366396336</td>
<td>0.6132246805394160910</td>
</tr>
<tr>
<td>2</td>
<td>0.099331925819428838</td>
<td>0.67190966685962169966</td>
</tr>
<tr>
<td>3</td>
<td>0.080347033483341786</td>
<td>0.870727620185996140</td>
</tr>
<tr>
<td>4</td>
<td>0.085638440500470454</td>
<td>1.229542231202690701</td>
</tr>
<tr>
<td>5</td>
<td>1.00954725932000372</td>
<td>1.826005967544425059</td>
</tr>
<tr>
<td>6</td>
<td>1.30574181535616015</td>
<td>2.80634530697762740</td>
</tr>
<tr>
<td>7</td>
<td>1.76105067658849928</td>
<td>4.41879329084928134</td>
</tr>
<tr>
<td>8</td>
<td>2.246835162402255547</td>
<td>7.083239057475765325</td>
</tr>
<tr>
<td>9</td>
<td>3.36424476867718855</td>
<td>11.5158812055758294</td>
</tr>
<tr>
<td>10</td>
<td>5.2700256230102743</td>
<td>18.93151118599927464</td>
</tr>
<tr>
<td>11</td>
<td>7.91389634259204750</td>
<td>31.4119969382049688</td>
</tr>
<tr>
<td>12</td>
<td>1.216702532429718021</td>
<td>52.52072945457382854</td>
</tr>
<tr>
<td>13</td>
<td>88.38485473506529806</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>149.5663744928804784</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>254.3179084310411743</td>
<td></td>
</tr>
</tbody>
</table>
Vacuum solver – Green’s functions

- $n = 0 \& n = 1$ use elliptic integrals
- $n > 1$ by recursion

$$s(x, x') = \frac{1}{4\pi|x - x'|} = \frac{1}{4\pi^2\sqrt{rr'}} \sum_{n\in\mathbb{Z}} e^{i n(\theta - \theta')} Q_{n-1/2}(\chi)$$

\begin{align*}
\frac{\partial\chi}{\partial r'} &= \frac{(r')^2 - r^2 - (z - z')^2}{2r(r')^2}, \\
\frac{\partial\chi}{\partial z'} &= \frac{z' - z}{rr'}, \\
Q_{-1/2}(\chi) &= \mu K(\mu), \\
Q_{1/2}(\chi) &= \chi \mu K(\mu) - \sqrt{2(\chi + 1)} E(\mu), \\
Q_{-n-1/2}(\chi) &= Q_{n-1/2}(\chi), \\
Q_{n-1/2}(\chi) &= 4 \frac{n-1}{2n-1} \chi Q_{n-3/2}(\chi) - \frac{2n-3}{2n-1} Q_{n-5/2}(\chi), \\
\frac{\partial Q_{n-1/2}(\chi)}{\partial \chi} &= \frac{2n-1}{2(\chi^2 - 1)} (\chi Q_{n-1/2} - Q_{n-3/2}),
\end{align*}
A Direct Solver for the Rapid Solution of Boundary Integral Equations on Axisymmetric Surfaces in Three Dimensions

Patrick Young and Per-Gunnar Martinsson
Dept. of Applied Mathematics, Univ. of Colorado at Boulder, Boulder, CO 80309-0526

Abstract: A scheme for rapidly and accurately computing solutions to boundary integral equations (BIEs) on rotationally symmetric surfaces in \( \mathbb{R}^3 \) is presented. The scheme uses the Fourier transform to reduce the original BIE defined on a surface to a sequence of BIEs defined on a generating curve for the surface. It can handle loads that are not necessarily rotationally symmetric. Nyström discretization is used to discretize the BIEs on the generating curve. The quadrature used is a high-order Gaussian rule that is modified near the diagonal to retain high-order accuracy for singular kernels. The reduction in dimensionality, along with the use of high-order accurate quadratures, leads to small linear systems that can be inverted directly via, e.g., Gaussian elimination. This makes the scheme particularly fast in environments involving multiple right hand sides. It is demonstrated that for BIEs associated with Laplace’s equation, the kernel in the reduced equations can be evaluated very rapidly by exploiting recursion relations for Legendre functions. Numerical examples illustrate the performance of the scheme; in particular, it is demonstrated that for a BIE associated with Laplace’s equation on a surface discretized using 320,000 points, the set-up phase of the algorithm takes 2 minutes on a standard desktop, and then solves can be executed in 0.5 seconds.

APPENDIX OF QUADRATURE NODES AND WEIGHTS

<table>
<thead>
<tr>
<th>( N_p )</th>
<th>2( N_p + 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1.93869e-04 4.10935e-07 5.37883e-08 5.37880e-08 5.37880e-08</td>
</tr>
<tr>
<td>20</td>
<td>1.93869e-04 4.10513e-07 3.30601e-12 1.66132e-13 1.66132e-13</td>
</tr>
<tr>
<td>40</td>
<td>1.93869e-04 4.10513e-07 3.23162e-12 8.28568e-14 8.28568e-14</td>
</tr>
<tr>
<td>80</td>
<td>1.93869e-04 4.10513e-07 2.92918e-12 2.92901e-13 2.92901e-13</td>
</tr>
</tbody>
</table>

Table 3. Error in internal Dirichlet problem solved on domain (a) in Figure 6.1.
Vacuum solver – Integrals

• Testing in stand-alone code
  – $n = 0$ & $n = 1$ so far
  – Manufacture solution using several random sources inside torus
  – Give $\partial_n \phi$, solve for $\phi$ compare

• Issues
  – Young & Martinsson use Nystrom method (collocation)
  – We need (eventually) FE method
  – Interpolation between various meshes
Manufactured solution
Vacuum solver – Integrals

- **Status**
  - Progress
  - Problems (bugs)
  - Higher-order not yet

Lagrange interpolation on G-L nodes

Lagrange interpolation on deg 5 regular
Work to date

• Implemented Young & Martinsson quadrature
• Tested FE direct integration version

\[ M_{ii'} = \int d\xi Jb_i \int d\xi' J' b_i' \]

  – Symmetrized showed jaggies
  – Asymmetric showed 1\textsuperscript{st} order convergence
• Testing Nystrom method on G-L mesh
  – Found description of interpolation vague
  – So – used different method near ends of each segment
  – Still buggy code, so need week or two more work
• Reproduce Y & M high-accuracy results
• Interpolate 2 ways to FE?
• Understand issues with direct FE formulation
NIMdevel interface

- Cihan built on Tibbar machine (almost)
- Available on NERSC (of course)
- Studying Andie’s implementation of rest of algorithm

\[
\begin{align*}
E_t &= \frac{R_w}{\mu_0} n \times [ [B_t] ] \\
B_t &= \nabla \phi \\
\dot{B}_n &= -n \cdot \nabla \times E_t
\end{align*}
\]