High fidelity RF wave solver for the fusion edge

M. Stowell, D. M. Copeland
Lawrence Livermore National Laboratory

D. L. Green, D. B. Batchelor, E. D'Azevedo
Oak Ridge National Laboratory

S. Frank, C. Migliore, S. Shiraiwa, J. Wright
Massachusetts Institute of Technology

M. Shepard, M. Hakimi, U. Riaz
Rensselaer Polytechnic Institute

RF-SciDAC : Center for Simulation of Fusion Relevant RF Actuators
http://rfscidac4.org

Stix Mini-App : Overview

- Plasma wave equation solver in pseudo-2D (work in progress) and pseudo-1D

$$\frac{1}{\mu_0}\nabla \times \nabla \times \vec{E} - \omega^2 \epsilon_0 \vec{E} + i\omega\vec{\sigma}\vec{E} = -i\omega \vec{J_{ext}}$$

- Current Boundary Conditions supported:
  - Conducting (1D, 2D)
  - Absorbing (1D, 2D)
  - Linear finite sheath (1D)

$$\vec{\sigma} = i\omega\varepsilon_0 \begin{pmatrix} S-1 & -iD & 0 \\ iD & S-1 & 0 \\ 0 & 0 & P-1 \end{pmatrix}$$

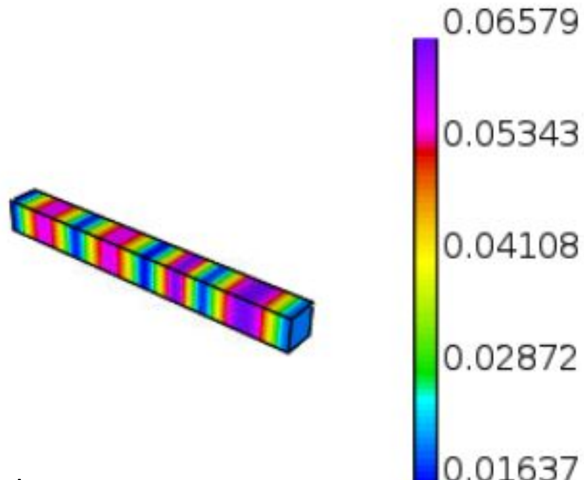
$$S = \tfrac{1}{2}(R+L),\ D = \tfrac{1}{2}(R-L)$$

$$R \equiv 1 - \textstyle\sum_i \frac{\omega_{pi}^2}{\omega(\omega+\Omega_i)}$$

$$L \equiv 1 - \textstyle\sum_i \frac{\omega_{pi}^2}{\omega(\omega-\Omega_i)}$$

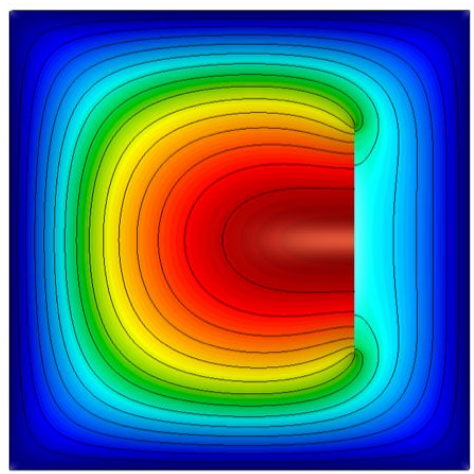
$$P \equiv 1 - \textstyle\sum_i \frac{\omega_{pi}^2}{\omega^2}$$

1D E Field Solution:



Plasma wave propagation

2D E Field Solution:

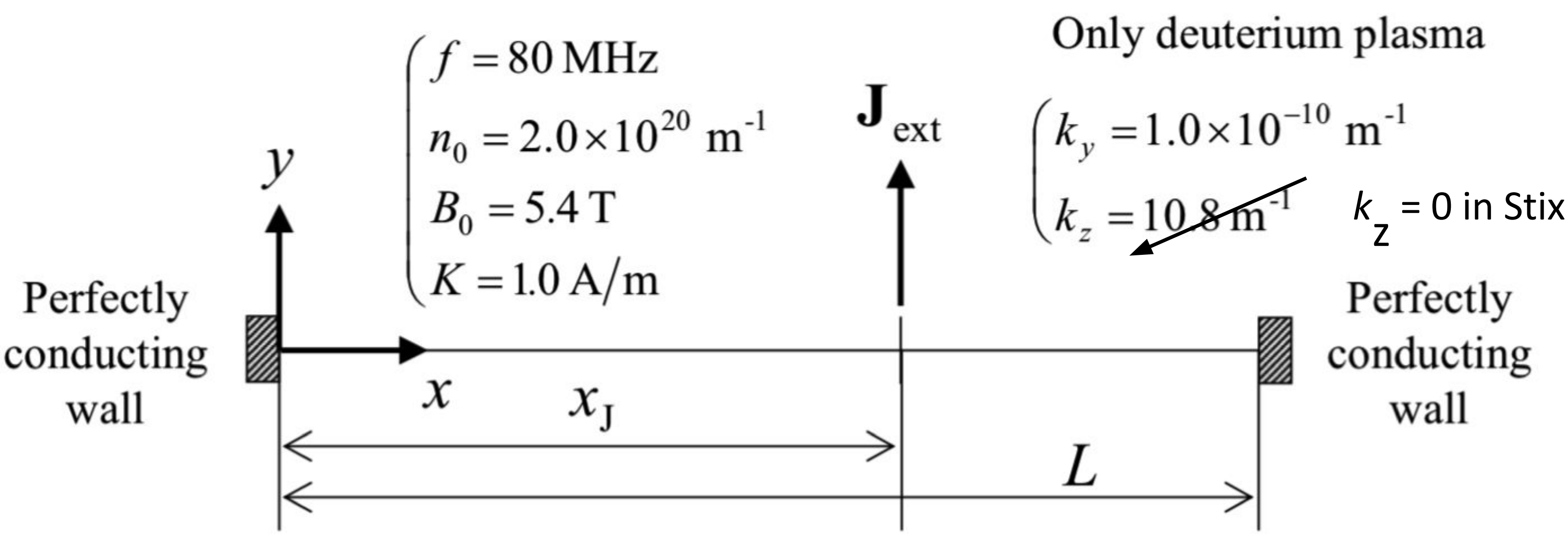


Diffusion

Stix Mini-App : 1D Model

(Based on Kohno 2017)

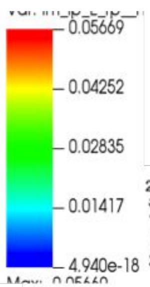
- Slab antenna
- B in z direction (can be changed)
- User specified BCs for all edge faces



[H. Kohno, et. al., Phys. Plasmas 19, 012508 (2012)]

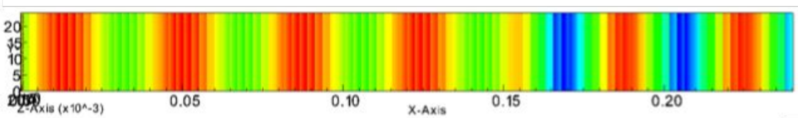
Stix Mini-App : Finite Sheath BC

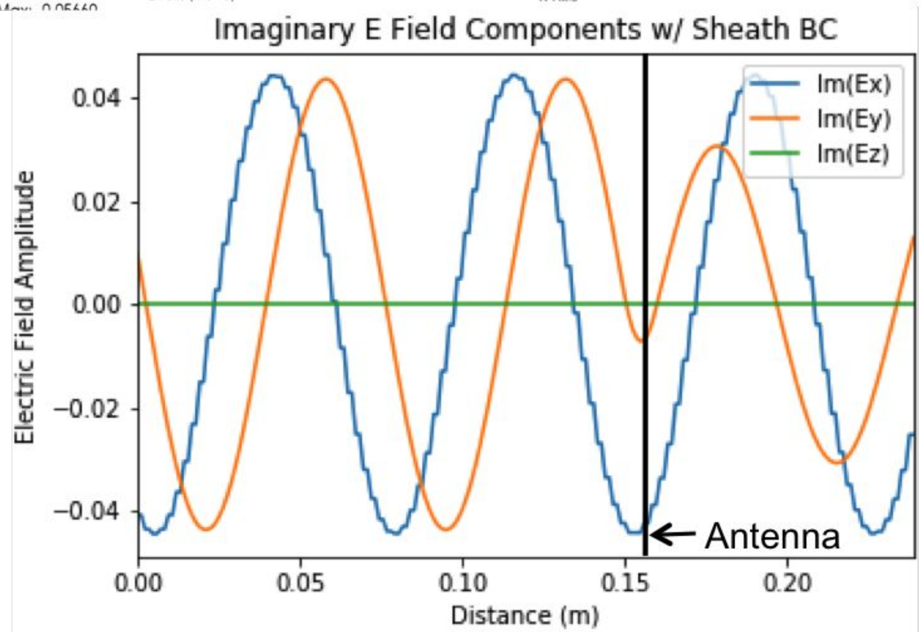
Example: Same plasma and setup parameters, different BCs



Linear z finite sheath BC:

$$E_t = \nabla_t \left(\frac{\omega}{i} D_n z \right)$$

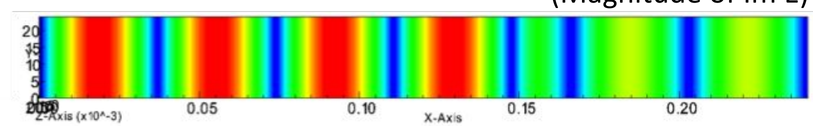


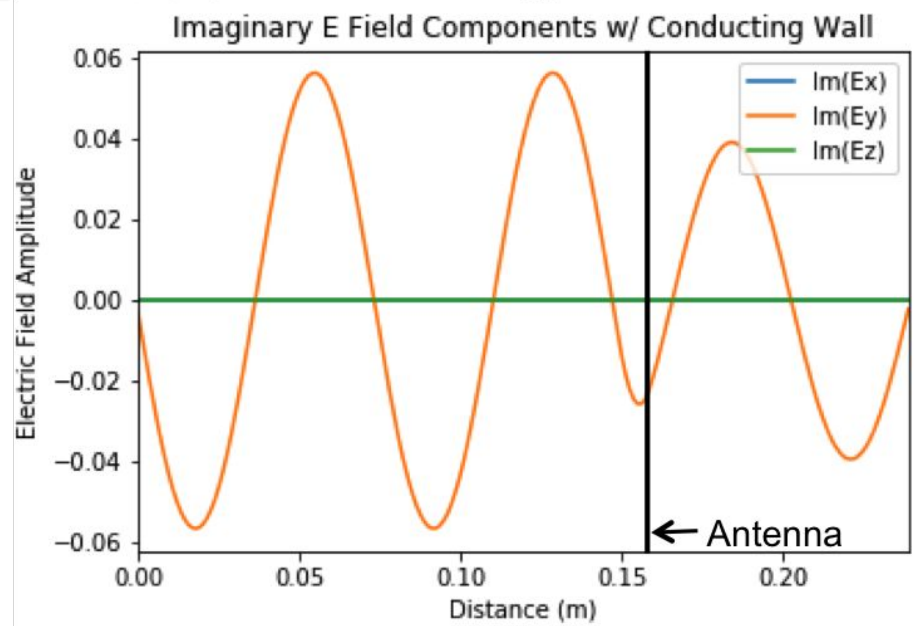


Conducting wall BC:

$$E_n, E_t = 0$$

(Magnitude of Im E)





Left plot shows sheath BC allows Ex to no longer be zero

RF-SciDAC

Stix Mini-App : Current and Future Work

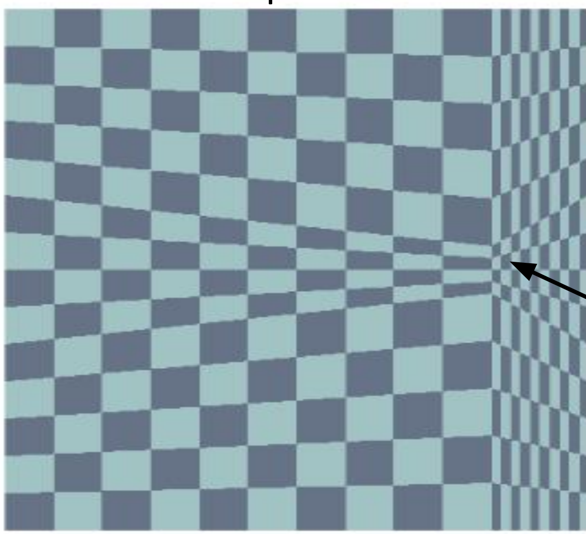
- In 1D, implementing non-linear finite sheath BC:

$$E_t = \nabla_t \left(\frac{\omega}{i} D_n z \right) \quad \text{where} \quad \frac{1}{z} = \frac{\langle J V_1 \rangle}{\langle V_1^2 \rangle} - \frac{i\omega \langle J \dot{V}_1 \rangle}{\langle \dot{V}_1^2 \rangle}$$

[J. Myra, et al., Phys. Plasmas 22, 062507 (2015)]

- In 2D Stix:
  - Incorporating current source in desired location

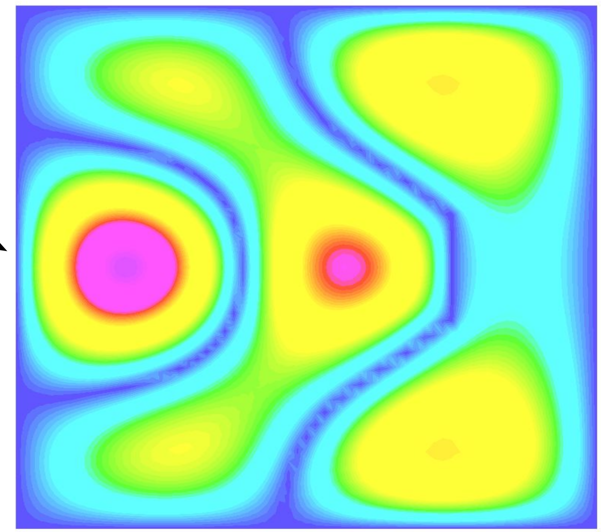
Example mesh



Antenna edge
(zero thickness)

Unsure whether this is
properly converged
(Work in progress)

E field solution of mesh

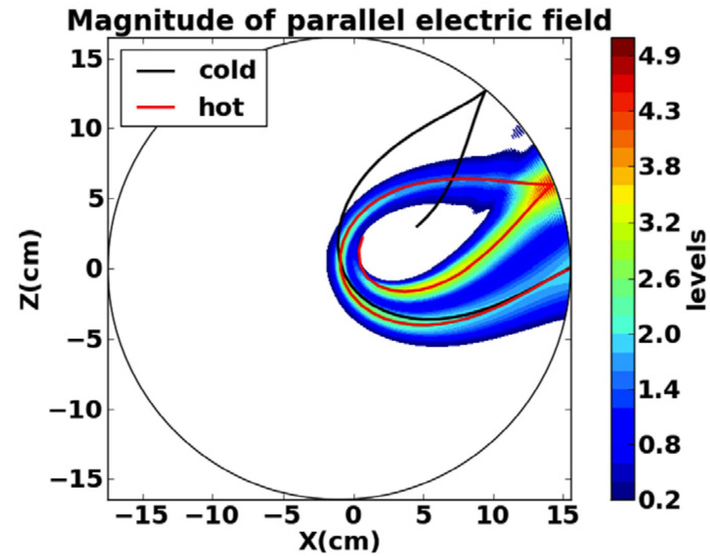


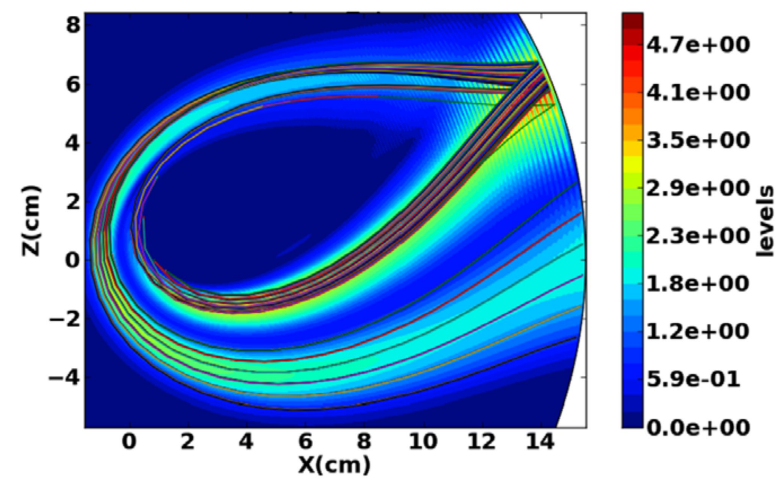
  - Linear finite sheath BC $\longrightarrow$ non-linear finite sheath BC
  - Create a more representative 2D mesh cross-section of a tokamak
- Non-uniform density in propagation direction
- Incorporating into PETRA-M

RF-SciDAC

Preconditioners : WKB I

- For large numbers of DOF (>100M) direct methods exhibit scaling problems which prevent utilizing of larger compute systems.
- Fast preconditioners are required to enable the problems of interest.
- WKB based preconditioner
  - Data parallel.
  - Can provide estimate of E field.
  - Requires volumetric source term to be used as a preconditioner.





Wright and Bertelli, Plasma Phys. Control. Fusion 56 (2014) 035006 (7pp)

RF-SciDAC

Preconditioners : WKB II

Solve Helmholtz equation in 1D

$$\frac{d^2Y}{dx^2} + k^2(x)Y = f(x)$$

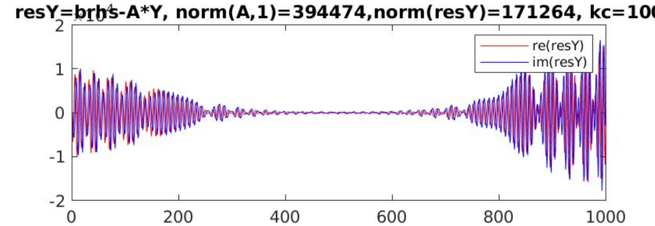
Homogeneous and particular solutions given as

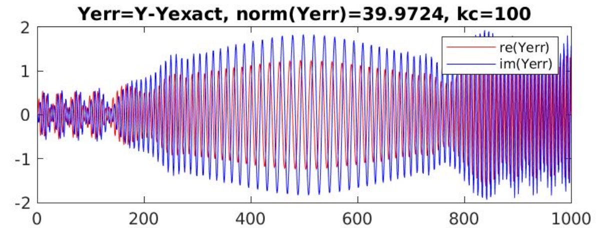
$$Y_1, Y_2 = \frac{1}{\sqrt{k(x)}} \exp(\pm i \int_0^x k(t)dt)$$

$$\begin{aligned} Y_p &= U_1(x)Y_1(x) + U_2(x)Y_2(x) \\ 0 &= U_1'Y_1 + U_2'Y_2 \\ f &= U_1'Y_1' + U_2'Y_2' \end{aligned}$$

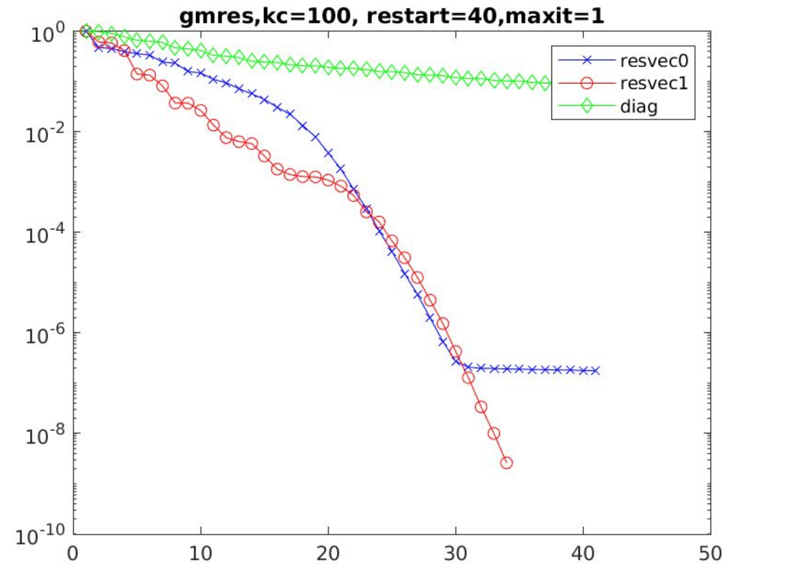
Solve for C1 and C2 to satisfy boundary conditions

$$Y = Y_p + C_1Y_1 + C_2Y_2$$





k(x) = 100 * (2 + cos(2*x)), numeric residual and analytic residual



Preconditioners : Algebraic I
Symmetric-Positive-Definite Preconditioner

We need a preconditioner for $\nabla \times (\mu^{-1}\nabla \times \vec{E}) - \omega^2 \epsilon \vec{E}$ where

$$\epsilon = \begin{pmatrix} S & -iD & 0 \\ iD & S & 0 \\ 0 & 0 & P \end{pmatrix}$$ Where $\vec{B} = |B|\hat{z}$

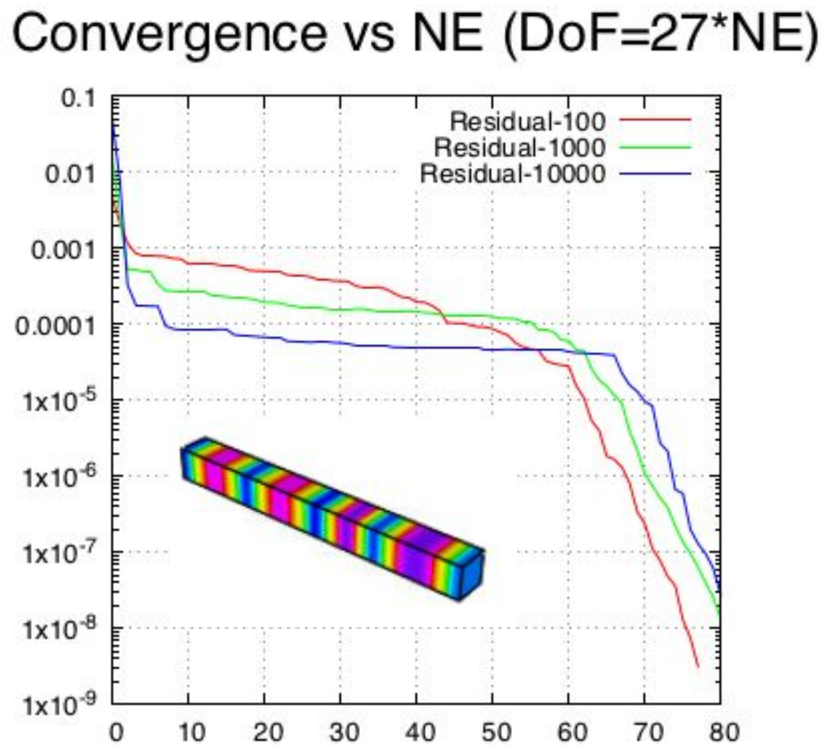
The obvious place to start is with $\nabla \times (\mu^{-1}\nabla \times \vec{E}) + \omega^2 \alpha \vec{E}$

Where $\alpha \equiv \begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & |P| \end{pmatrix}$ and $A \equiv \frac{1}{2}(|S+D| + |S-D|)$

Clearly this can easily be rotated to align with more general magnetic fields.

Preconditioners : Algebraic II
Applications in 1D

- Same 1D case as in miniapp, with sheath boundary conditions -> Non Hermitian, requires GMRES
- Using Euclid ILU preconditioner with symmetric positive definite precondition described on last slide
- Single core runs.
- Iteration count slowly grows as number of elements increases - likely because error correction has to propagate along wave characteristics
- Solver time increases linearly



NE	Memory(MB)	Time(s)
100	44	0.15
1000	170	1.6
100001411	17	

RF-SciDAC

Preconditioners : Domain Decomposition

- While the positive definite algebraic preconditioner works for 1D problems, the larger null space in higher dimensions becomes problematic.
- A domain decomposition preconditioner where the solution is solved exactly (SuperLU or SPARSEPACK) leaving error only at the interfaces should work better in 3D problems.
- Initial 3D study for model problem on the right:

h	k^2	kh	Iter	Re $\|E_h - E\|_{L^2}$	Re $\frac{\|E_h - E\|_{L^2}}{\|E\|_{L^2}}$	Im $\|E_h\|_{L^2}$
0.140308	2	0.20	33	0.174009	0.28416	3.35228e-09
0.140308	10	0.44	32	0.126879	0.20719	2.11609e-09
0.140308	51	1.00	39	0.135315	0.22097	1.16478e-08
0.0701539	2	0.10	38	0.0912488	0.14901	5.21567e-09
0.0701539	10	0.22	34	0.0625769	0.10219	1.00206e-09
0.0701539	51	0.50	45	0.0662409	0.10817	6.04593e-09
0.0350769	2	0.050	38	0.0912488	0.14901	5.21567e-09
0.0350769	51	0.25	49	0.0329131	0.053746905	4.29369e-08

Table 1: GMRES (relative tolerance 1e-8) tests on complex-valued DDM reduced interface system, with FEM order 1. Parameters from Rawat and Lee 2010 are used. L^2 norms are on one subdomain (results are nearly identical on the two subdomains). Exact solution (1).

Exact solution for model source with Exn=0:

$$E = (\sin(\pi y)\sin(\pi z), \sin(\pi x)\sin(\pi z), \sin(\pi x)\sin(\pi y)).$$

$$\nabla \times \nabla \times E - k^2 E = (2\pi^2 - k^2)E.$$

RF-SciDAC

Preconditioners : Reduced Precision I

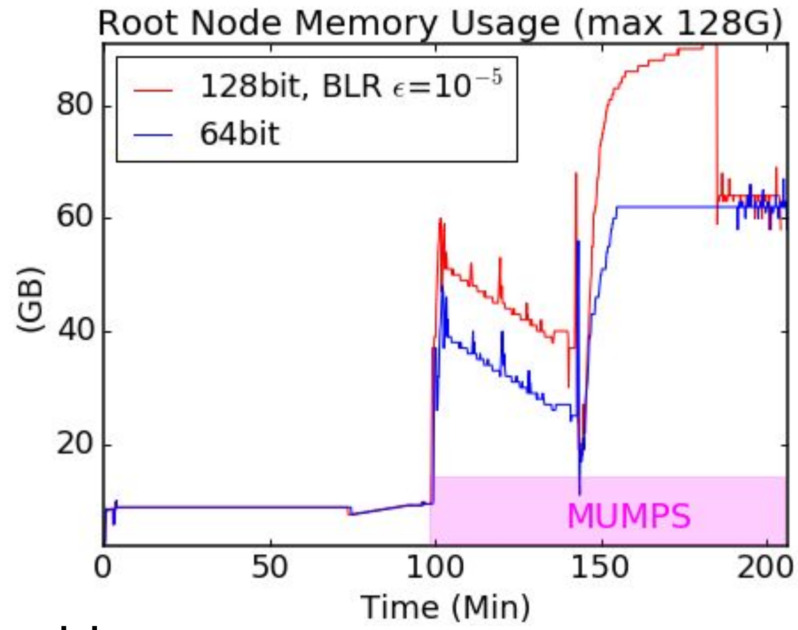
Using a direct solver to provide an approximate inverse of system matrix reduces memory and/or FLOPS requirement, thus allowing for solving a larger problem size.

Low rank approximations

- BLR (Block Low Rank)
- HSS (Hierarchically Semi-Separable)

Lowering floating point precision

- Single precision



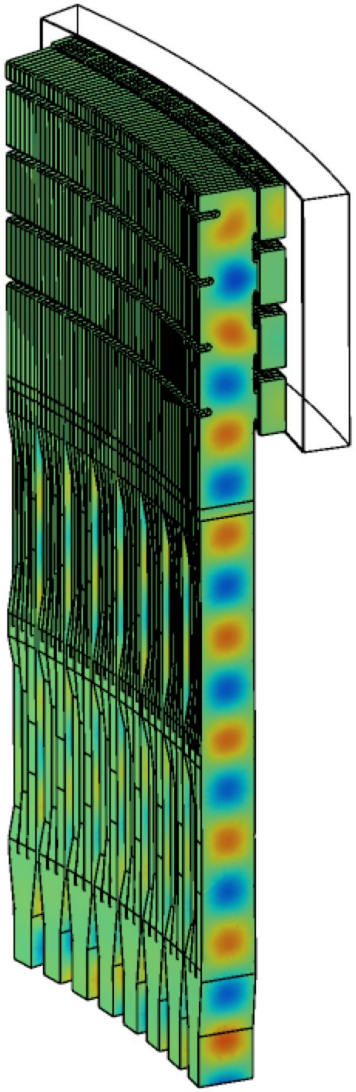
Advantages:

- Immediately applicable to realistic 3D problems.
- Potentially fit well for coarse grid solver in MG

Questions:

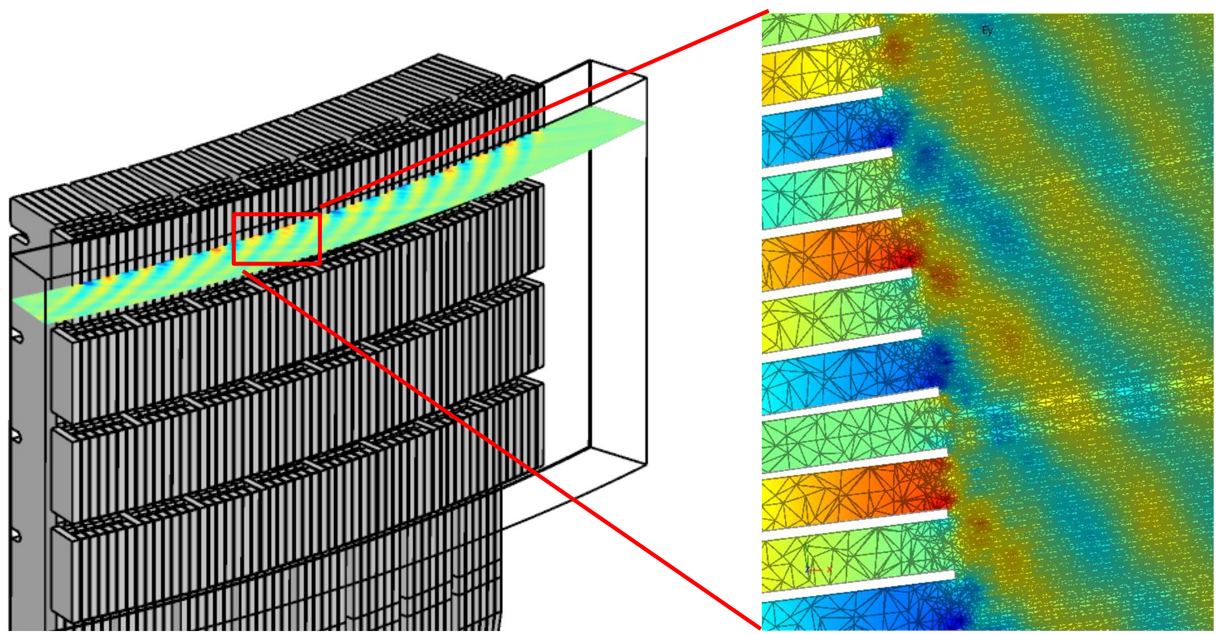
Does it work well for indefinite-Maxwell problems?

Preconditioners : Reduced Precision II



DIII-D high field silde lower hybrid launcher being designed/built by MIT

- 15-20 GMRES iteration
- Possible to resolve LH wave scattering in 3D
- 110 M DoFs (exceeding project 5th year milestone)



Code Acceleration : TORLH GPU Porting

- GPU Hackathon Small Scale (1 Summit Topology Node) Results
- NVBLAS – 2-3x speedup primarily in calls to SCALAPACK matrix multiplication routines with no code restructuring.

356 sec Wall Clock

torlh	was called	1 times, total	356.86 secs
gloini	was called	1 times, total	9.02 secs
fftl_ini	was called	1 times, total	[illegible].99 secs
equil	was called	1 times, total	0.03 secs
mhdcoe	was called	1741 times, total	0.00 secs
coords	was called	1107 times, total	0.07 secs
fftl_dir	was called	13515 times, total	0.06 secs
init zvec	was called	1 times, total	0.00 secs
allocGk	was called	1 times, total	0.05 secs
allocFk	was called	1 times, total	0.04 secs
mblock	was called	33 times, total	82.44 secs
mblock:inside plasma	was called	99 times, total	82.36 secs
operev	was called	[illegible] times, total	77.17 secs
fftl_inv	was called	[illegible] times, total	7.44 secs
dieltn	was called	[illegible]7172 times, total	14.68 secs
zetaRe	was called	19032064 times, total	8.14 secs
cgamswap	was called	132 times, total	9.66 secs
pzgetrf	was called	167 times, total	15.43 secs
pzgemm	was called	825 times, total	79.53 secs
cgamswap2	was called	99 times, total	1.79 secs
add_bloc[illegible]	was called	33 times, total	89.74 secs
form rhs from zk	was called	33 times, total	0.00 secs
pzgetrs,Gk=EkBk	was called	1 times, total	1.15 secs
pzgetrs,yk=Ekyk	was called	33 times, total	0.08 secs
yvec<-yk	was called	33 times, total	0.00 secs
putGk	was called	33 times, total	0.02 secs
pzgetrs,Gk=EkGk	was called	32 times, total	38.42 secs
ykm1<-yvec	was called	32 times, total	0.00 secs
pzgemm,yk=yk-Ck*ykm1	was called	32 times, total	0.07 secs
pzgetrs,T1=T2T1	was called	1 times, total	1.10 secs
pzgetrs,T3=T2T3 2	was called	1 times, total	1.18 secs
pzgetrs,ycyc=T2ycyc	was called	2 times, total	0.00 secs
ptri_solve_replicate	was called	1 times, total	0.27 secs
solve: xvec <- x_in	was called	1 times, total	0.00 secs
ptri_solve_distribut	was called	1 times, total	0.12 secs
xvec<-xk	was called	33 times, total	0.00 secs
getGk	was called	32 times, total	0.04 secs
yk<-yvec	was called	32 times, total	0.00 secs
pzgemm, xk=xk-Gk*xkp	was called	32 times, total	0.08 secs
solve: x_in <- xvec	was called	1 times, total	0.00 secs
deallocGk	was called	1 times, total	0.00 secs
deallocFk	was called	1 times, total	0.00 secs
elepwr	was called	1 times, total	19.10 secs
t0_powinit	was called	1 times, total	0.00 secs
t0_operev	was called	1 times, total	0.27 secs
t0_deffld	was called	1 times, total	0.05 secs
deffld	was called	33 times, total	1.65 secs
t0_surfpl1	was called	1 times, total	0.00 secs
t0_powtwd1	was called	1 times, total	0.00 secs
powtwd	was called	11 times, total	1.57 secs
elepwr-mpi	was called	1 times, total	0.01 secs
t0_elbfld	was called	1 times, total	0.01 secs
elbfld	was called	481 times, total	3.97 secs

215 sec Wall Clock

2-3x speedup in most time consuming routines!

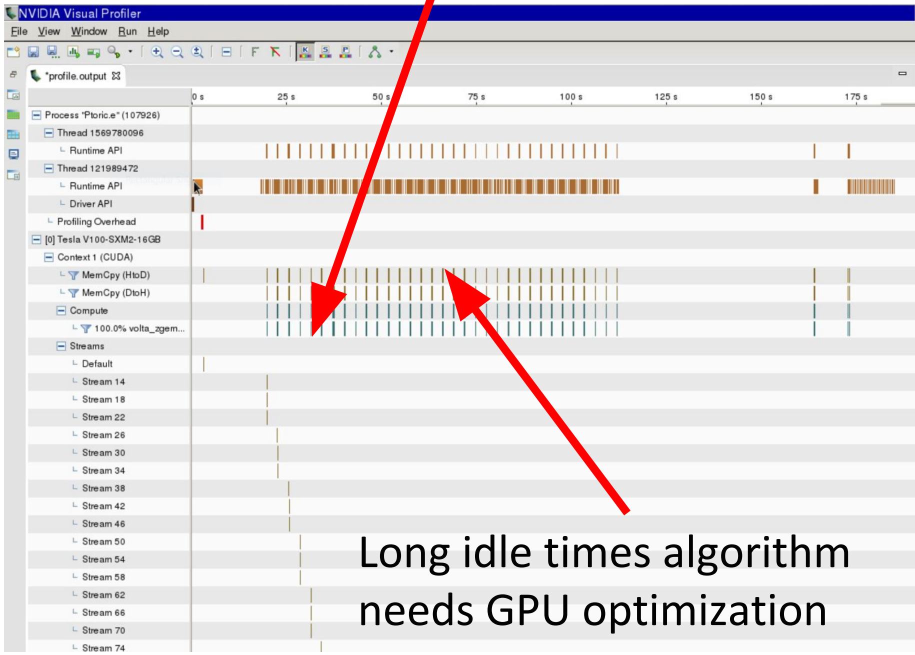
torlh	was called	1 times, total	215.75 secs
gloini	was called	1 times, total	9.09 secs
fftl_ini	was called	1 times, total	8.95 secs
equil	was called	1 times, total	0.15 secs
mhdcoe	was called	1741 times, total	0.00 secs
coords	was called	1107 times, total	0.08 secs
fftl_dir	was called	13515 times, total	0.07 secs
init zvec	was called	1 times, total	0.00 secs
allocGk	was called	1 times, total	0.05 secs
allocFk	was called	1 times, total	0.04 secs
mblock	was called	33 times, total	57.56 secs
mblock:inside plasma	was called	99 times, total	57.46 secs
operev	was called	99 times, total	51.33 secs
fftl_inv	was called	[illegible]631217 times, total	7.96 secs
dieltn	was called	37172 times, total	15.19 secs
zetaRe	was called	[illegible]032064 times, total	8.26 secs
cgamswap	was called	132 times, total	5.74 secs
pzgetrf	was called	167 times, total	31.75 secs
pzgemm	was called	825 times, total	27.97 secs
cgamswap2	was called	99 times, total	4.90 secs
add_block_row_distri	was called	33 times, total	53.29 secs
form rhs from zk	was called	33 times, total	[illegible]
pzgetrs,Gk=EkBk	was called	1 times, total	0.39 secs
pzgetrs,yk=Ekyk	was called	33 times, total	0.22 secs
yvec<-yk	was called	33 times, total	0.00 secs
putGk	was called	[illegible]	[illegible]
pzgetrs,Gk=EkGk	was called	32 times, total	12.54 secs
ykm1<-yvec	was called	32 times, total	0.00 secs
pzgemm,yk=yk-Ck*ykm1	was called	32 times, total	0.12 secs
pzgetrs,T1=T2T1	was called	1 times, total	0.38 secs
pzgetrs,T3=T2T3 2	was called	1 times, total	0.38 secs
pzgetrs,ycyc=T2ycyc	was called	2 times, total	0.01 secs
ptri_solve_replicate	was called	1 times, total	0.25 secs
solve: xvec <- x_in	was called	1 times, total	0.00 secs
ptri_solve_distribut	was called	1 times, total	0.14 secs
xvec<-xk	was called	33 times, total	0.00 secs
getGk	was called	32 times, total	0.04 secs
yk<-yvec	was called	32 times, total	0.00 secs
pzgemm, xk=xk-Gk*xkp	was called	32 times, total	0.10 secs
solve: x_in <- xvec	was called	1 times, total	0.00 secs
deallocGk	was called	1 times, total	0.00 secs
deallocFk	was called	1 times, total	0.00 secs
elepwr	was called	1 times, total	20.21 secs
t0_powinit	was called	1 times, total	0.00 secs
t0_operev	was called	1 times, total	0.28 secs
t0_deffld	was called	1 times, total	0.07 secs
deffld	was called	33 times, total	2.49 secs
t0_surfpl1	was called	1 times, total	0.00 secs
t0_powtwd1	was called	1 times, total	0.00 secs
powtwd	was called	11 times, total	1.63 secs
elepwr-mpi	was called	1 times, total	0.01 secs
t0_elbfld	was called	1 times, total	0.01 secs
elbfld	was called	481 times, total	4.01 secs

Internal TORLH profiling

GPU Calls

Code Acceleration : TORLH GPU Porting

Initial TORLH time profile with nvprof



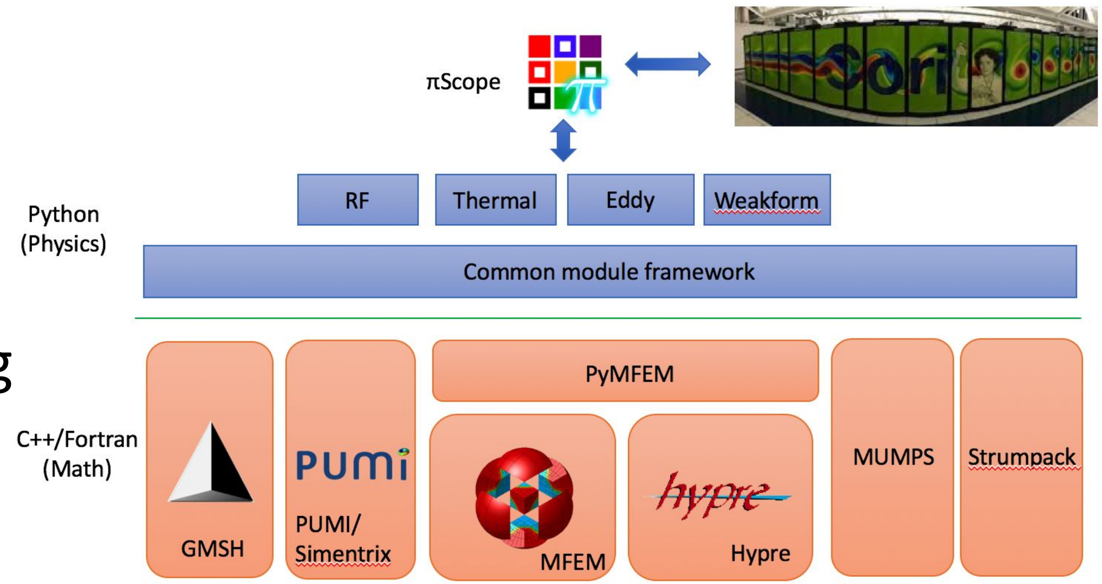
- OpenACC acceleration of loop-heavy matrix-build routines lead to 2-5x speedup of those routines.
- Communication, idle times, and parallelization changes for GPUs lead to overall slowdown of matrix build however.
- Performance at larger scales (10s to 100s nodes, 10x resolution) is next step.
  - We avoided code restructuring – long lived legacy code. Code restructuring to an effective, GPU based, algorithm with minimized CPU-GPU communication will be likely be necessary to achieve performance.
  - Lots to do on one node, but really need ~10 nodes to see how tuning is different at scale. Planning to request some Summit time to continue work.

RF-SciDAC

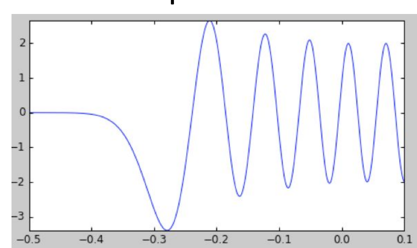
Petra-M : Physics Equation Translator for MFEM

FEM analysis platform

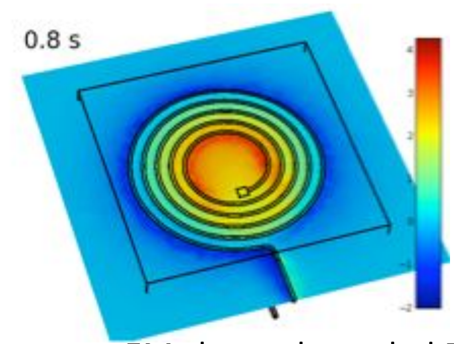
- Front-end interface to open source software
- Integrated FEM modeling from geometry to FEM assembly and solve.
- Deployment tool for our advanced physics model
  - Started as RF modeling tool and being applied in many RF problems in fusion experiments
  - Use case is even expanding outside RF waves



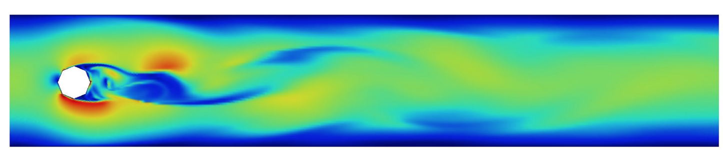
Petra-M examples



1D RF wave



EM-thermal coupled PDE



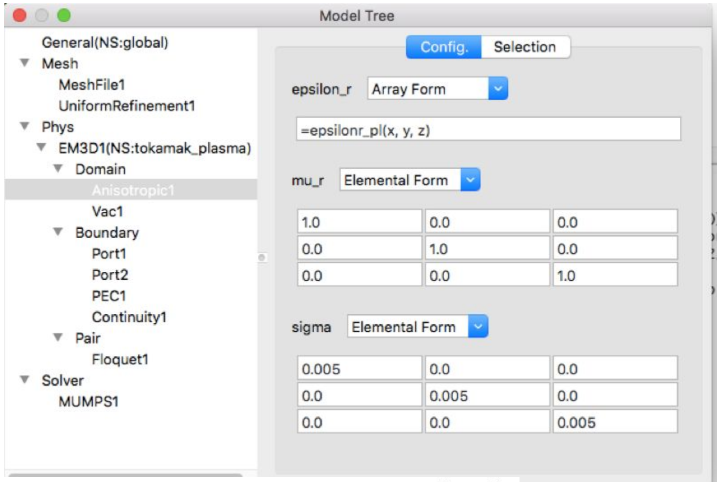
Navier-Stokes

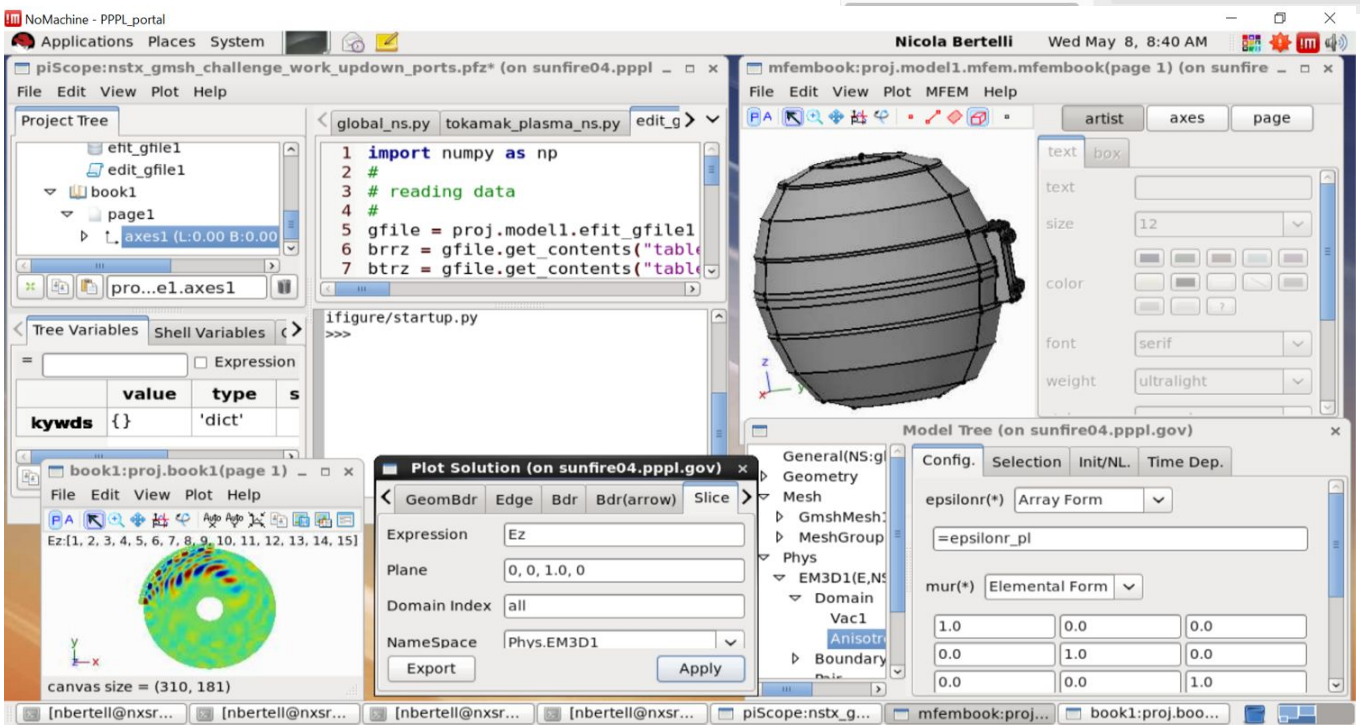
RF-SciDAC

Petra-M : RF physics layer on the screen

Solve inhomogeneous Maxwell eq. in frequency domain.

- Cold plasma with collisions
- RF port BCs (Coax, waveguides)
- 3D/2D/1D geometry

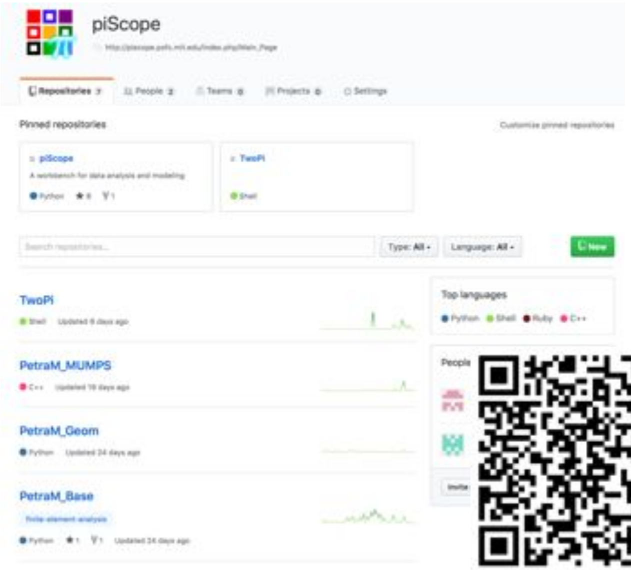




Petra-M : continues to improve under RF-SciDAC

The first version was released 2018 Aug and being updated constantly

- Support new libraries, such as HYPRE 2.16, MUMUP 5.2, STRUMPACK 3.1
- Add better interface for multi-physics coupling, pre-conditioner...
- Current focus is…
  - MFEM ver 4 support
  - Python 3.7 (migrating Python is trivial, but absorbing Python C-API difference is more challenging)



Physics goal in FY 2020

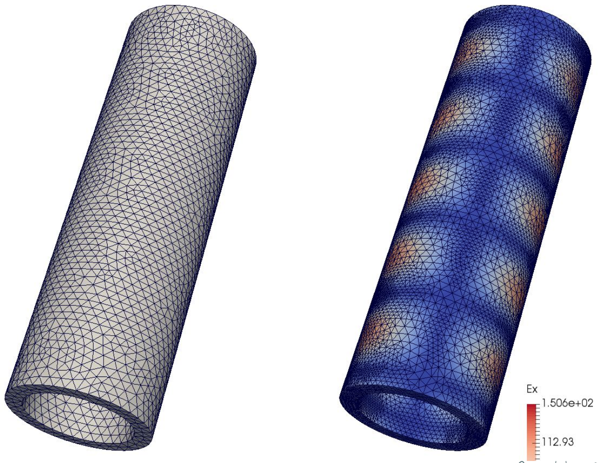
- Incorporate RF sheath BC to estimate impurity production on JET ILA.
- Incorporate adaptive refinement to allow resolution of slow and fast waves.

Meshing : Adaptive RF Simulation Workflow

- Workflow steps
  1. Obtain and clean-up antenna CAD models
  2. Combine antenna, reactor and physics geometries
  3. Associate analysis attributes
  4. Automatic mesh generation
  5. MFEM finite element analysis
  6. Estimate discretization errors. If below tolerance terminate
  7. Adapt mesh and return to step 5

Steps 1-3 are interactive
Steps 4-7 are automated

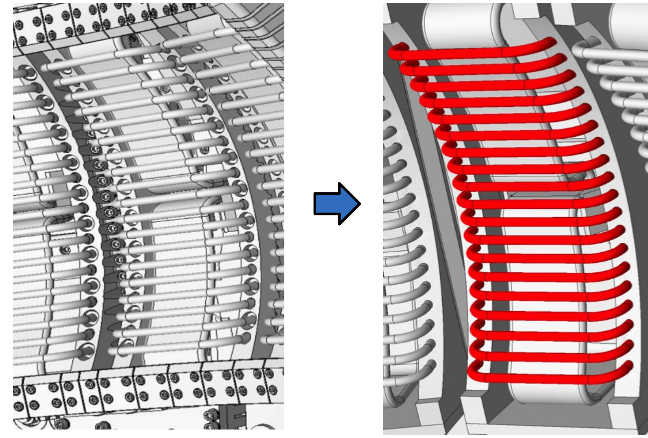
- Tools used
  - Geometry – Simmetrix and CAD systems
  - Meshing – Simmetrix
  - RF Simulation – PetraM and MFEM
  - Error estimation, mesh adaptation - PUMI



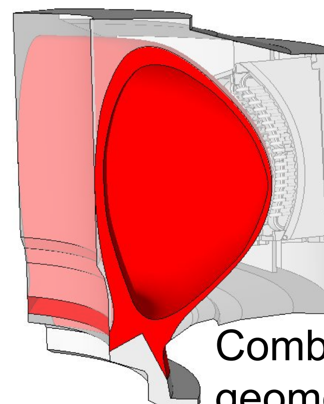
RF-SciDAC

Meshing : Defeaturing CAD, Combining Geometry

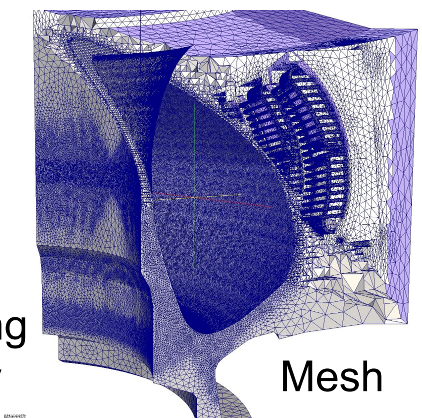
- Analysis geometry definition
  - Antenna defeatured using SimModeler GUI
  - Reactor cross section extruded
  - EFIT data used to construct flux surface
  - Model components combined
- Mesh generation
  - Mesh controls on analysis geometry
  - Automatic mesh generation
  - Curved mesh geometry inflation to higher than quadradic geometry (initial version – further efforts required)



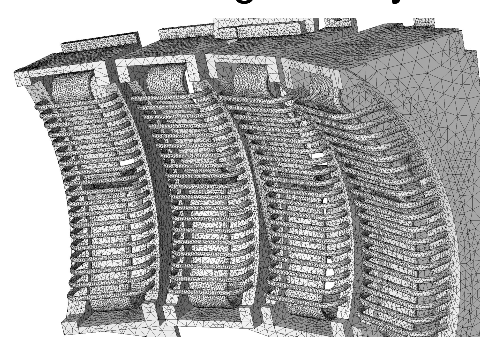
Defeaturing



Combining geometry



Mesh



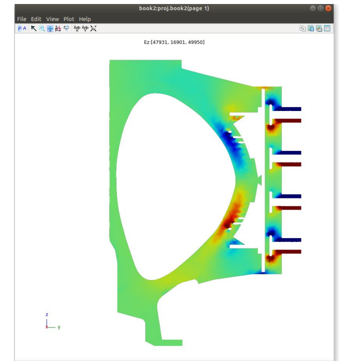
Close-up of 2.5M element quadratic mesh

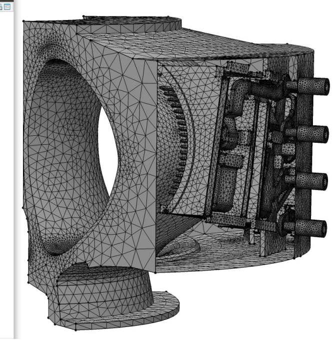
RF-SciDAC

Meshing : Adaptivity in PetraM/MFEM

- Integration with PetraM/MFEM
  - Parallel meshes integrated with MFEM – full in memory integration
  - Linkage with PetraM for attribute specification – current version has some file transfer – will be fully in-memory
- Adaptive Mesh Control
  - Patch recovery method for RF implemer
  - Conforming mesh adaptation
- Adaptive Example
  - Stratified D port antenna for C-Mod

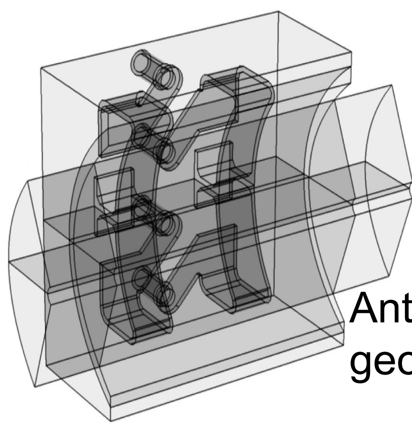
PetraM attribute specification



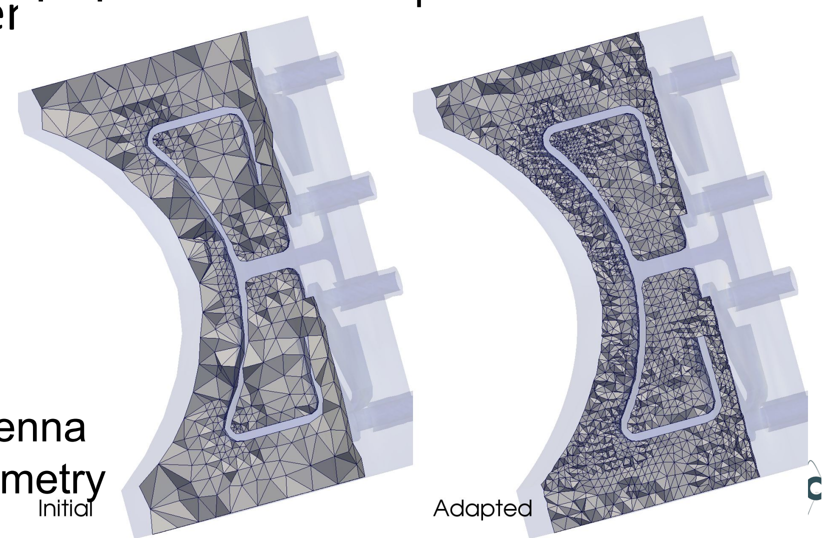


PetraM/MFEM result - electric field component on a cut plane

Initial and adapted meshes



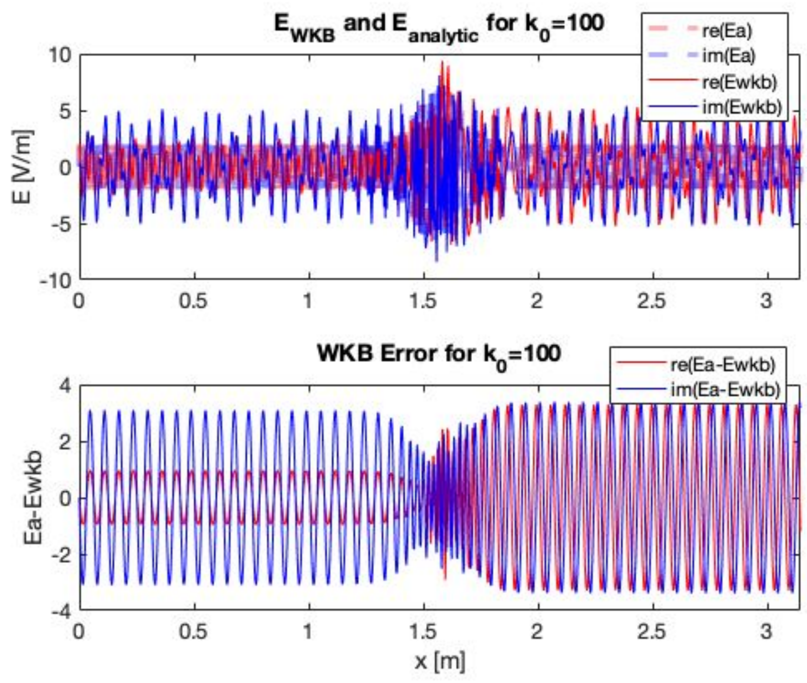
Antenna geometry

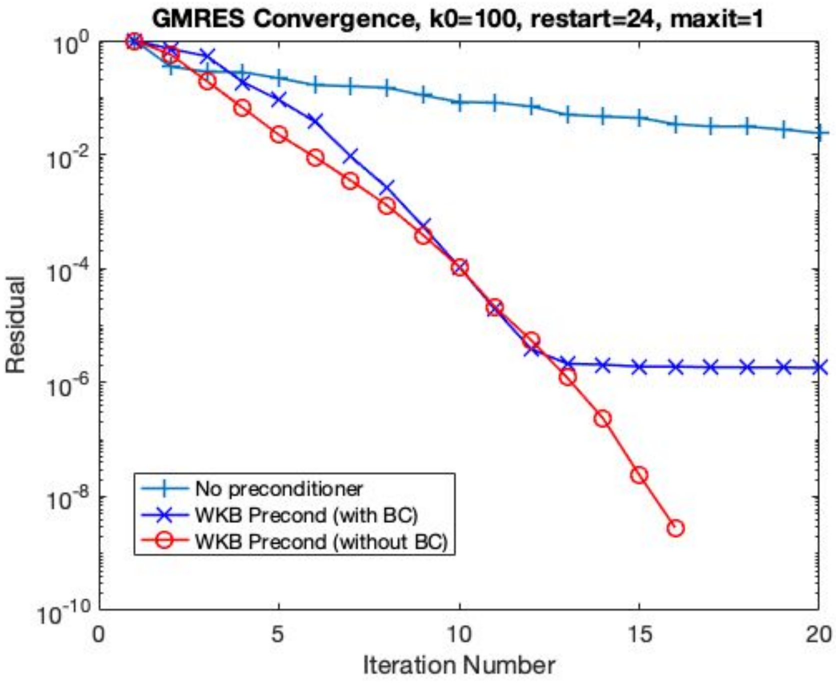


Backup / removed slides

Preconditioner : WKB Approach III

Comparison of iteration convergence using GMRES with WKB preconditioner for case where the WKB approximation has significant error - similar to a mode conversion surface.





Edge / far-SOL RF Wave Solver Component