The BEM++ boundary element library and applications

Timo Betcke

University College London

Supported by

EPSRC Grant EP/I030042/1

The BEM++ Library

- Core library in C++, complete interface via Python
- Support for Laplace, Helmholtz, Maxwell
- Shared-Memory parallelisation
- Integration of AHMED for H-Matrix assembly and algebra
- Extensive support for iterative solvers via interfaces to Trilinos (C++) and PyTrilinos (Python)
- BSD style open source license
- Currently, Mac and Linux supported

The BEM++ Project

UCL

Simon Arridge
Timo Betcke
Richard James (former member)
Nicolas Salles
Martin Schweiger
Woicjech Smigaj (former member)
New Postdoc (joining May ‘14)

Reading

Stephen Langdon
Chris Westbrook

Warwick

Andreas Dedner
Alastair Radcliffe

Many thanks to

Lars Kielhorn (ETH)
Olaf Steinbach (Graz)
Gerhard Unger (Graz)
Peter Monk (Delaware)

Layer Potentials

Single Layer Potential:

$$[\mathcal{V}\psi](x) = \int_{\Gamma} g(x,y)\psi(y)ds(y), \;\; x \in \Omega$$

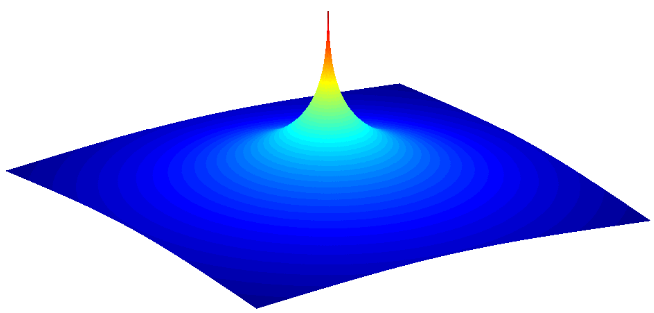
Double Layer Potential:

$$[\mathcal{K}\phi](x) = \int_{\Gamma} \gamma_{1,y} g(x,y)\phi(y)ds(y), \;\; x \in \Omega$$

$$\mathcal{V} : H^{-1/2}(\Gamma) \to H^{1}(\Omega), \quad \mathcal{K} : H^{1/2}(\Gamma) \to H^{1}(\Omega)$$

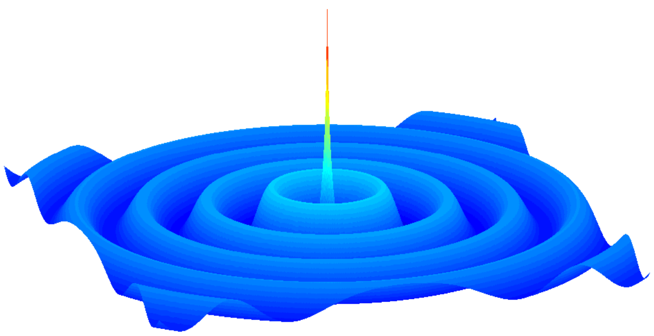
Green's function

Laplace



$$g(x,y) = \frac{1}{4\pi|x-y|}$$

Helmholtz



$$g(x,y) = \frac{e^{ik|x-y|}}{4\pi|x-y|}$$

Green's representation theorem and the Calderon projector

$$u(x) = [\mathcal{V}t](x) - [\mathcal{K}v](x), \;\; x \in \Omega$$

$$\gamma_0^{int} \qquad\qquad \gamma_1^{int}$$

$$v = \gamma_0^{int}[\mathcal{V}t] - \gamma_0^{int}[\mathcal{K}v] \qquad t = \gamma_1^{int}[\mathcal{V}t] - \gamma_1^{int}[\mathcal{K}v]$$

Calderon projection

$$\begin{bmatrix} v \\ t \end{bmatrix} = \begin{bmatrix} -\gamma_0^{int}\mathcal{K} & \gamma_0^{int}\mathcal{V} \\ -\gamma_1^{int}\mathcal{K} & \gamma_1^{int}\mathcal{V} \end{bmatrix} \begin{bmatrix} v \\ t \end{bmatrix} =: \begin{bmatrix} \frac{1}{2}I - K & V \\ D & \frac{1}{2}I + T \end{bmatrix} \begin{bmatrix} v \\ t \end{bmatrix}$$

Boundary Potential Operators

Single Layer Boundary Potential:

$$[V\psi](x) = \int_\Gamma g(x,y)\psi(y)ds(y),\ \ x \in \Gamma$$

Double Layer Boundary Potential:

$$[K\phi](x) = \int_\Gamma \gamma_{1,y}^{int} g(x,y)\phi(y)ds(y),\ \ x \in \Gamma$$

Adjoint Double Layer Boundary Potential

$$[T\psi](x) = \int_\Gamma \gamma_{1,x}^{int} g(x,y)\psi(y)ds(y),\ \ x \in \Gamma$$

Hypersingular Boundary Potential:

$$[D\phi](x) = -\gamma_{1,x}^{int} \int_\Gamma \gamma_{1,y}^{int} g(x,y)\phi(y)ds(y),\ \ x \in \Gamma$$

Traces and mapping properties

Traces:

$$[\gamma_0^{int}\mathcal{V}\psi](x) = [V\psi](x)$$

$$[\gamma_1^{int}\mathcal{V}\psi](x) = \frac{1}{2}\psi(x) + [T\psi](x)$$

$$[\gamma_0^{int}\mathcal{K}\phi](x) = -\frac{1}{2}\phi(x) + [K\phi](x)$$

$$[\gamma_1^{int}\mathcal{K}\phi](x) = -[D\phi](x)$$

Mapping properties:

$$V : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma)$$

$$T : H^{-1/2}(\Gamma) \to H^{-1/2}(\Gamma)$$

$$K : H^{1/2}(\Gamma) \to H^{1/2}(\Gamma)$$

$$D : H^{1/2}(\Gamma) \to H^{-1/2}(\Gamma)$$

Calderon projection in terms of boundary potentials

$$\begin{bmatrix} v \\ t \end{bmatrix} = \begin{bmatrix} \frac{1}{2}I - K & V \\ D & \frac{1}{2}I + T \end{bmatrix} \begin{bmatrix} v \\ t \end{bmatrix}$$

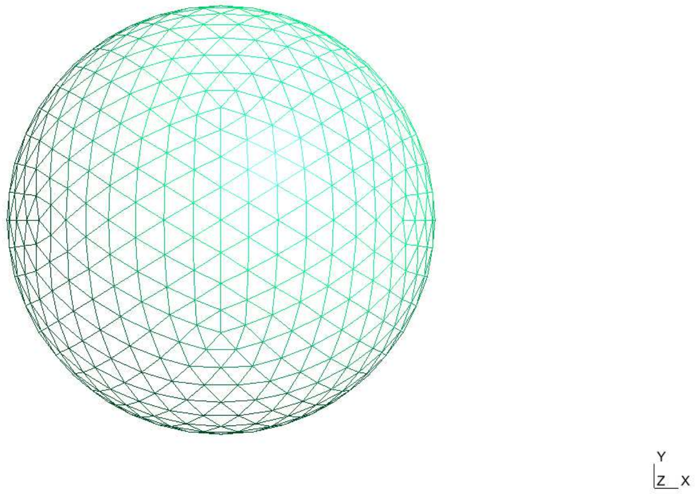
Operators in BEM++

Function	Weak form
`identityOperator()`	$\int_\Gamma \overline{\phi}(\mathbf{x})\,\psi(\mathbf{x})\,\mathrm{d}\Gamma(\mathbf{x})$
`maxwell3dIdentityOperator()`	$\int_\Gamma \overline{\boldsymbol{\phi}}(\mathbf{x})\cdot[\boldsymbol{\psi}(\mathbf{x})\times\boldsymbol{n}(\mathbf{x})]$
`laplaceBeltrami3dOperator()`	$\int_\Gamma \boldsymbol{\nabla}_\Gamma\overline{\phi}(\mathbf{x})\cdot\boldsymbol{\nabla}_\Gamma\psi(\mathbf{x})\,\mathrm{d}\Gamma(\mathbf{x})$
`laplace3dSingleLayerBoundaryOperator()`	$\int_\Gamma\int_\Sigma \overline{\phi}(\mathbf{x})\,g(\mathbf{x},\mathbf{y})\,\psi(\mathbf{y})\,\mathrm{d}\Gamma(\mathbf{x})\,\mathrm{d}\Sigma(\mathbf{y})$
`laplace3dDoubleLayerBoundaryOperator()`	$\int_\Gamma\int_\Sigma \overline{\phi}(\mathbf{x})\,\partial_{\mathbf{n}(\mathbf{y})}g(\mathbf{x},\mathbf{y})\,\psi(\mathbf{y})\,\mathrm{d}\Gamma(\mathbf{x})\,\mathrm{d}\Sigma(\mathbf{y})$
`laplace3dAdjointDoubleLayerBoundaryOperator()`	$\int_\Gamma\int_\Sigma \overline{\phi}(\mathbf{x})\,\partial_{\mathbf{n}(\mathbf{x})}g(\mathbf{x},\mathbf{y})\,\psi(\mathbf{y})\,\mathrm{d}\Gamma(\mathbf{x})\,\mathrm{d}\Sigma(\mathbf{y})$
`laplace3dHypersingularBoundaryOperator()`	$\int_\Gamma\int_\Sigma \mathbf{curl}_\Gamma\,\overline{\phi}(\mathbf{x})\cdot g(\mathbf{x},\mathbf{y})\ \mathbf{curl}_\Sigma\,\psi(\mathbf{y})\,\mathrm{d}\Gamma(\mathbf{x})\,\mathrm{d}\Sigma(\mathbf{y})$
`helmholtz3dSingleLayerBoundaryOperator()`	$\int_\Gamma\int_\Sigma \overline{\phi}(\mathbf{x})\,g_k(\mathbf{x},\mathbf{y})\,\psi(\mathbf{y})\,\mathrm{d}\Gamma(\mathbf{x})\,\mathrm{d}\Sigma(\mathbf{y})$
`helmholtz3dDoubleLayerBoundaryOperator()`	$\int_\Gamma\int_\Sigma \overline{\phi}(\mathbf{x})\,\partial_{\mathbf{n}(\mathbf{y})}g_k(\mathbf{x},\mathbf{y})\,\psi(\mathbf{y})\,\mathrm{d}\Gamma(\mathbf{x})\,\mathrm{d}\Sigma(\mathbf{y})$
`helmholtz3dAdjointDoubleLayerBoundaryOperator()`	$\int_\Gamma\int_\Sigma \overline{\phi}(\mathbf{x})\,\partial_{\mathbf{n}(\mathbf{x})}g_k(\mathbf{x},\mathbf{y})\,\psi(\mathbf{y})\,\mathrm{d}\Gamma(\mathbf{x})\,\mathrm{d}\Sigma(\mathbf{y})$
`helmholtz3dHypersingularBoundaryOperator()`	$\int_\Gamma\int_\Sigma g_k(\mathbf{x},\mathbf{y})[\mathbf{curl}_\Gamma\,\overline{\phi}(\mathbf{x})\cdot\mathbf{curl}_\Sigma\,\psi(\mathbf{y})$ $-k^2\,\overline{\phi}(\mathbf{x})\,\mathbf{n}(\mathbf{x})\cdot\psi(\mathbf{y})\,\mathbf{n}(\mathbf{y})]\,\mathrm{d}\Gamma(\mathbf{x})\,\mathrm{d}\Sigma(\mathbf{y})$
`maxwell3dSingleLayerBoundaryOperator()`	$\int_\Gamma\int_\Sigma g_k(\mathbf{x},\mathbf{y})[-\mathrm{i}k\,\overline{\boldsymbol{\phi}}(\mathbf{x})\cdot\boldsymbol{\psi}(\mathbf{y})$ $-\frac{1}{\mathrm{i}k}\,\mathrm{div}_\Gamma\,\overline{\phi}(\mathbf{x})\ \mathrm{div}_\Sigma\,\psi(\mathbf{y})]\,\mathrm{d}\Gamma(\mathbf{x})\,\mathrm{d}\Sigma(\mathbf{y})$
`maxwell3dDoubleLayerBoundaryOperator()`	$\int_\Gamma\int_\Sigma \boldsymbol{\nabla}_\mathbf{x} g_k(\mathbf{x},\mathbf{y})\cdot[\overline{\boldsymbol{\phi}}(\mathbf{x})\times\boldsymbol{\psi}(\mathbf{y})]\,\mathrm{d}\Gamma(\mathbf{x})\,\mathrm{d}\Sigma(\mathbf{y})$

Boundary element spaces

Assume triangular surface elements.

Nodes: x_i Elements: τ_ℓ



Space of piecewise constant functions:

$$S_h^0(\Gamma) := \operatorname{span}\{\psi_k^{(0)}\}$$

$$\psi_k^{(0)}(x) = \begin{cases} 1, & x \in \tau_k \\ 0, & \text{otherwise} \end{cases}$$

Space of continuous piecewise linear functions:

$$S_h^1(\Gamma) := \operatorname{span}\{\phi_j^{(1)}\}$$

$$\phi_j^{(1)}(x_i) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$$S_h^0 \subset H^{-1/2}(\Gamma)$$

$$S_h^1 \subset H^{1/2}(\Gamma)$$

Spaces in BEM++

Name	Description
`PiecewiseConstantScalarSpace`	Space $S_h^{(0)}$ of piecewise constant functions.
`PiecewiseConstantDualGridScalarSpace`	Space of piecewise constant functions defined on the dual grid.
`PiecewiseLinearContinuousScalarSpace`	Space $S_h^{(1)}$ of continuous piecewise linear functions.
`PiecewiseLinearDiscontinuousScalarSpace`	Space of element-wise linear functions.
`PiecewisePolynomialContinuousScalarSpace`	Space of continuous piecewise polynomial functions.
`PiecewisePolynomialDiscontinuousScalarSpace`	Space of element-wise polynomial functions.
`RaviartThomas0VectorSpace`	Space of lowest-order Raviart-Thomas basis functions.
`UnitScalarSpace`	Space of globally constant functions.

Formulating a Dirichlet problem

Inner product: $\langle f, g \rangle = \int_\Gamma \overline{f}(x) g(x) ds(x)$

Find $t \in H^{-1/2}(\Gamma)$, such that

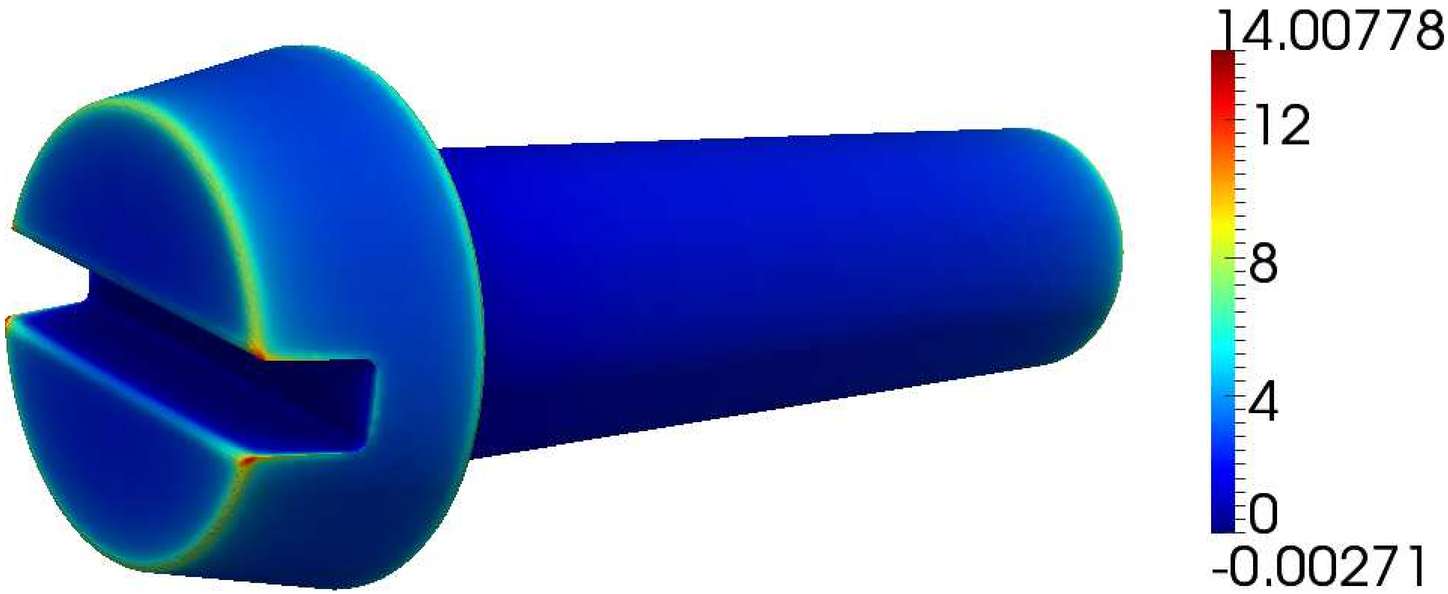
$$\langle \psi, Vt \rangle = \langle \psi, \left(\frac{1}{2}I + K\right) g \rangle$$

for all $\psi \in H^{-1/2}(\Gamma)$.

Note: $\psi \in H^{-1/2}(\Gamma),\ Vt \in H^{1/2}(\Gamma)$

Need to take pairings of dual spaces!

A more complex geometry.



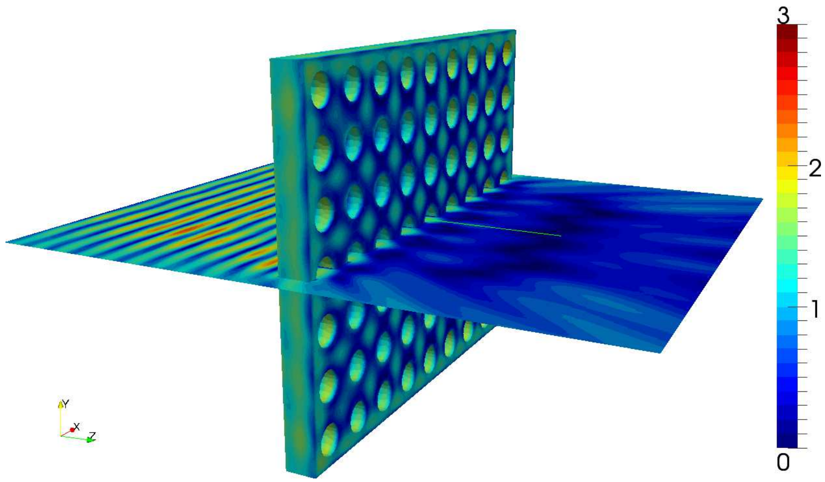
Charge density (in nC/m^2) on the surface of a 13 mm long bolt held at the potential u= 1V.

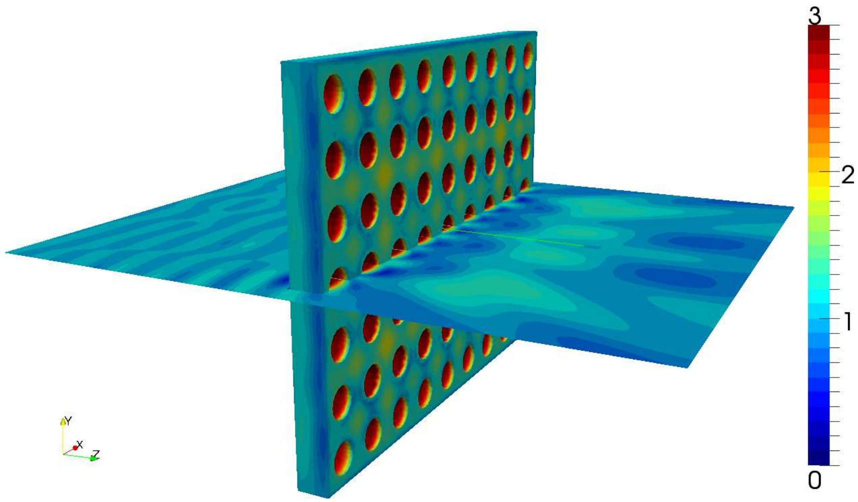
154,076 elements, 369 seconds in total for computation (12 core workstation)

Performance of Laplace Dirichlet Problems

#Elem.	ACA tol.	DLP op. Mem. (MB / %)	DLP op. t (s)	SLP op. Mem. (MB / %)	SLP op. t (s)	Solver #It.	Solver t (s)	Relative L^2 error	Relative $H^{-\frac{1}{2}}$ error
80	1E−2	0.0 / 100	0.1	0.0 / 100	0.0	15	0.0	8.56E−2	2.95E−2
80	1E−3	0.0 / 100	0.1	0.0 / 100	0.0	15	0.0	8.56E−2	2.95E−2
80	1E−4	0.0 / 100	0.1	0.0 / 100	0.0	15	0.0	8.56E−2	2.95E−2
320	1E−3	0.3 / 81	0.2	0.7 / 96	0.0	24	0.1	4.11E−2	9.34E−3
320	1E−4	0.3 / 88	0.1	0.8 / 97	0.0	23	0.1	4.11E−2	9.33E−3
320	1E−5	0.4 / 99	0.1	0.8 / 99	0.0	23	0.2	4.11E−2	9.33E−3
1280	1E−3	2.6 / 41	0.4	4.2 / 34	0.1	35	0.1	1.98E−2	3.17E−3
1280	1E−4	3.0 / 47	0.4	5.1 / 41	0.1	33	0.1	1.97E−2	3.12E−3
1280	1E−5	3.5 / 55	0.5	6.1 / 49	0.1	33	0.0	1.97E−2	3.12E−3
5120	1E−4	16.0 / 16	2.0	29.4 / 15	0.6	45	0.2	9.72E−3	1.09E−3
5120	1E−5	20.9 / 21	2.3	36.6 / 18	0.7	44	0.3	9.71E−3	1.08E−3
5120	1E−6	26.0 / 26	2.7	45.3 / 23	0.8	43	0.3	9.71E−3	1.08E−3
20480	1E−4	92.1 / 6	10.4	150.3 / 5	3.3	55	1.8	4.92E−3	4.04E−4
20480	1E−5	121.8 / 8	12.6	192.6 / 6	3.9	53	1.7	4.83E−3	3.81E−4
20480	1E−6	155.4 / 10	15.1	244.8 / 8	4.7	53	1.7	4.83E−3	3.81E−4
81920	1E−5	669.5 / 3	69.7	958.6 / 2	20.6	62	10.2	2.41E−3	1.35E−4
81920	1E−6	866.9 / 3	84.7	1238.7 / 2	24.9	61	10.3	2.41E−3	1.34E−4
81920	1E−7	1065.6 / 4	99.5	1541.3 / 3	29.8	61	11.2	2.41E−3	1.34E−4
327680	1E−6	4755.7 / 1	469.5	6004.8 / 1	125.5	68	41.6	1.21E−3	4.75E−5
327680	1E−7	5892.2 / 1	555.1	7515.7 / 1	150.7	68	46.2	1.21E−3	4.75E−5
327680	1E−8	7083.8 / 2	643.5	9124.5 / 1	178.0	68	46.6	1.21E−3	4.75E−5

A perforated sound-hard plate





Blocking of acoustic energy (490s)

Transmission of acoustic energy (450s)

31,966 elements

Operator algebra

BEM++ allows constructs such as:

$$t_2 = D * V * t$$

Galerkin formulation:

$$V_{Galerkin} : H^{-1/2}(\Gamma) \rightarrow \left[H^{1/2}(\Gamma)\right]', \; V_{Galerkin}\psi := \langle \cdot, V\psi \rangle$$

$$D_{Galerkin} : H^{1/2}(\Gamma) \rightarrow \left[H^{-1/2}(\Gamma)\right]', \; D_{Galerkin}\phi := \langle \cdot, D\phi \rangle$$

BEM++ automatically maps between spaces and their duals.

$t_2 = M_2^\dagger * D * M_1^\dagger * V * t$ (Internal BEM++ computation)

Mass matrices: M_1, M_2

H-Matrix Assembly of Operators

$$\langle D_k u, v\rangle_\Gamma = \frac{1}{4\pi}\int_\Gamma\int_\Gamma \frac{e^{ik|x-y|}}{|x-y|}\left(\mathrm{curl}u(y), \mathrm{curl}v(x)\right) ds(y)ds(x)$$
$$-\frac{k^2}{4\pi}\int_\Gamma\int_\Gamma \frac{e^{ik|x-y|}}{|x-y|} u(y)v(x)\left(n(x), n(y)\right) ds(y)ds(y),\ \ u,v\in H^{1/2}(\Gamma)$$

Global Assembly: Evaluate directly the weak form, dofs are vertices of mesh.

Local Assembly: Use representation in terms of sums of single layer potentials.

$$A \approx \sum_j Q_j H_j P_j$$

Hybrid assembly:

1. For admissible blocks assemble elementwise the strong form to obtain

$$A_{IJ} \approx Q_{IJ} H_{IJ} P_{IJ}$$

2. Coalesce each block to obtain H-Matrix representation.

Performance

Equation	Operator	ACA mode	Storage (MB)	Time (s)
Laplace	DLP	Global	120	8.0
		Hybrid	132	6.0
		Local	410	6.2
	Hyp	Global	94	29.0
		Hybrid	107	15.0
		Local	729	9.9
Helmholtz	DLP	Global	396	20.5
		Hybrid	438	15.2
		Local	1356	15.5
	Hyp	Global	282	73.0
		Hybrid	306	34.5
		Local	2246	26.4
Maxwell	SLP	Global	1698	56.3
		Hybrid	(unsupported)	
		Local	2246	26.2
Maxwell	DLP	Global	1659	50.7
		Hybrid	(unsupported)	
		Local	(unsupported)	

BIE for Maxwell

$$(\boldsymbol{\Psi}_{\mathrm{SL}}\boldsymbol{v})(\boldsymbol{x}) \equiv \mathrm{i}k\int_\Gamma G(\boldsymbol{x},\boldsymbol{y})\boldsymbol{v}(\boldsymbol{y})\Gamma(\boldsymbol{y}) - \frac{1}{\mathrm{i}k}\boldsymbol{\nabla}_{\boldsymbol{x}}\int_\Gamma G(\boldsymbol{x},\boldsymbol{y})(\boldsymbol{\nabla}_\Gamma\cdot\boldsymbol{v})(\boldsymbol{y})\Gamma(\boldsymbol{y})$$

$$(\Psi_{\mathrm{DL}}\boldsymbol{v})(\boldsymbol{x}) \equiv \boldsymbol{\nabla}_{\boldsymbol{x}}\times\int_\Gamma G(\boldsymbol{x},\boldsymbol{y})\boldsymbol{v}(\boldsymbol{y})\Gamma(\boldsymbol{y})$$

Boundary operator relations for exterior problems

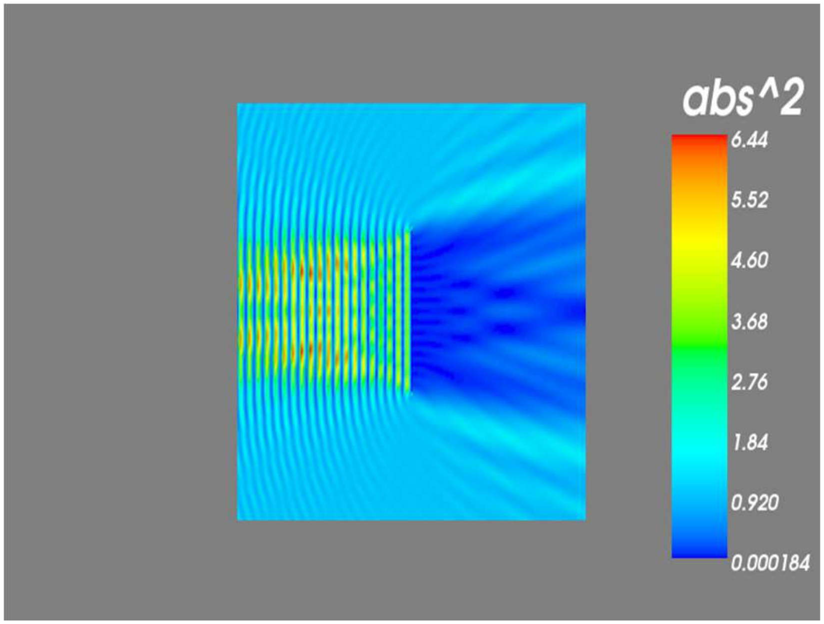
$$\begin{aligned}(\tfrac{1}{2}\boldsymbol{I}+\boldsymbol{C})\gamma_{\mathrm{D,ext}}\boldsymbol{u}+\boldsymbol{S}\gamma_{\mathrm{N,ext}}\boldsymbol{u}&=0\\ -\boldsymbol{S}\gamma_{\mathrm{D,ext}}\boldsymbol{u}+(\tfrac{1}{2}\boldsymbol{I}+\boldsymbol{C})\gamma_{\mathrm{N,ext}}\boldsymbol{u}&=0.\end{aligned}$$

$$(\gamma_{\mathrm{D,ext}}\boldsymbol{u})(\boldsymbol{x}) \equiv \boldsymbol{u}|_{\Gamma,\mathrm{ext}}(\boldsymbol{x})\times\boldsymbol{n}(\boldsymbol{x}),$$

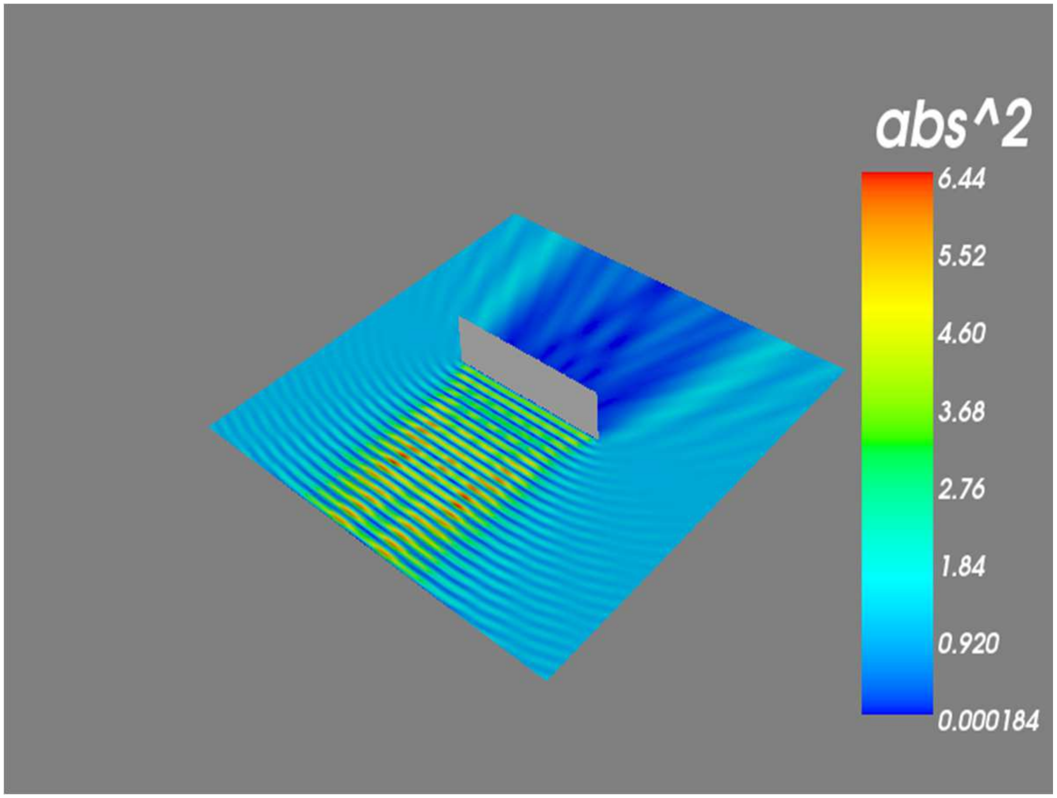
$$(\gamma_{\mathrm{N,ext}}\boldsymbol{u})(\boldsymbol{x}) \equiv (\mathrm{i}k)^{-1}(\boldsymbol{\nabla}\times\boldsymbol{u})|_{\Gamma,\mathrm{ext}}(\boldsymbol{x})\times\boldsymbol{n}(\boldsymbol{x}).$$

[Buffa/Hiptmair ’03]

Scattering from an electromagnetic screen

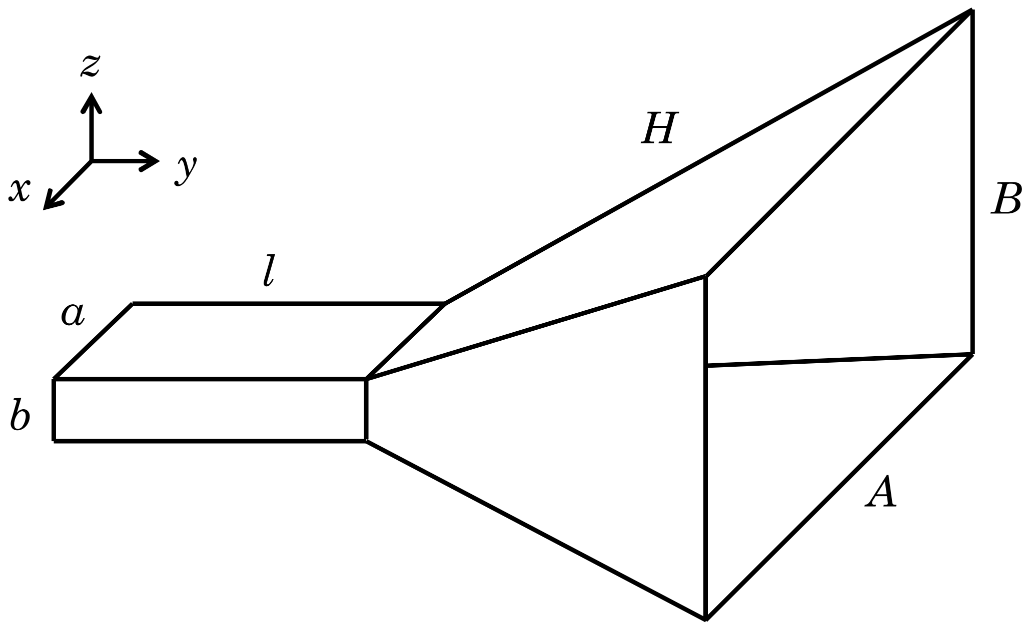


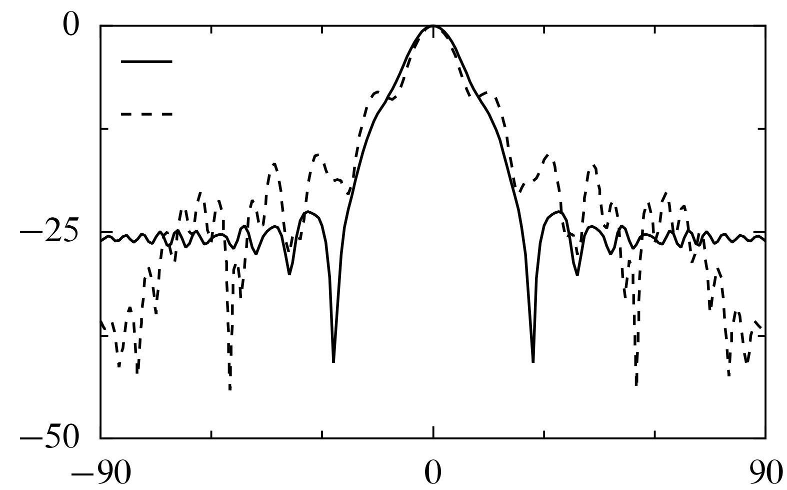
- Electric Field Integral Equation
- Automatic handling of edge dofs on boundary



- 12 Core Workstation
- 3.6s for operator assembly
- 5.9s for HLU preconditioner
- 0.1s solve time
- 7 GMRES iterations

An open horn antenna





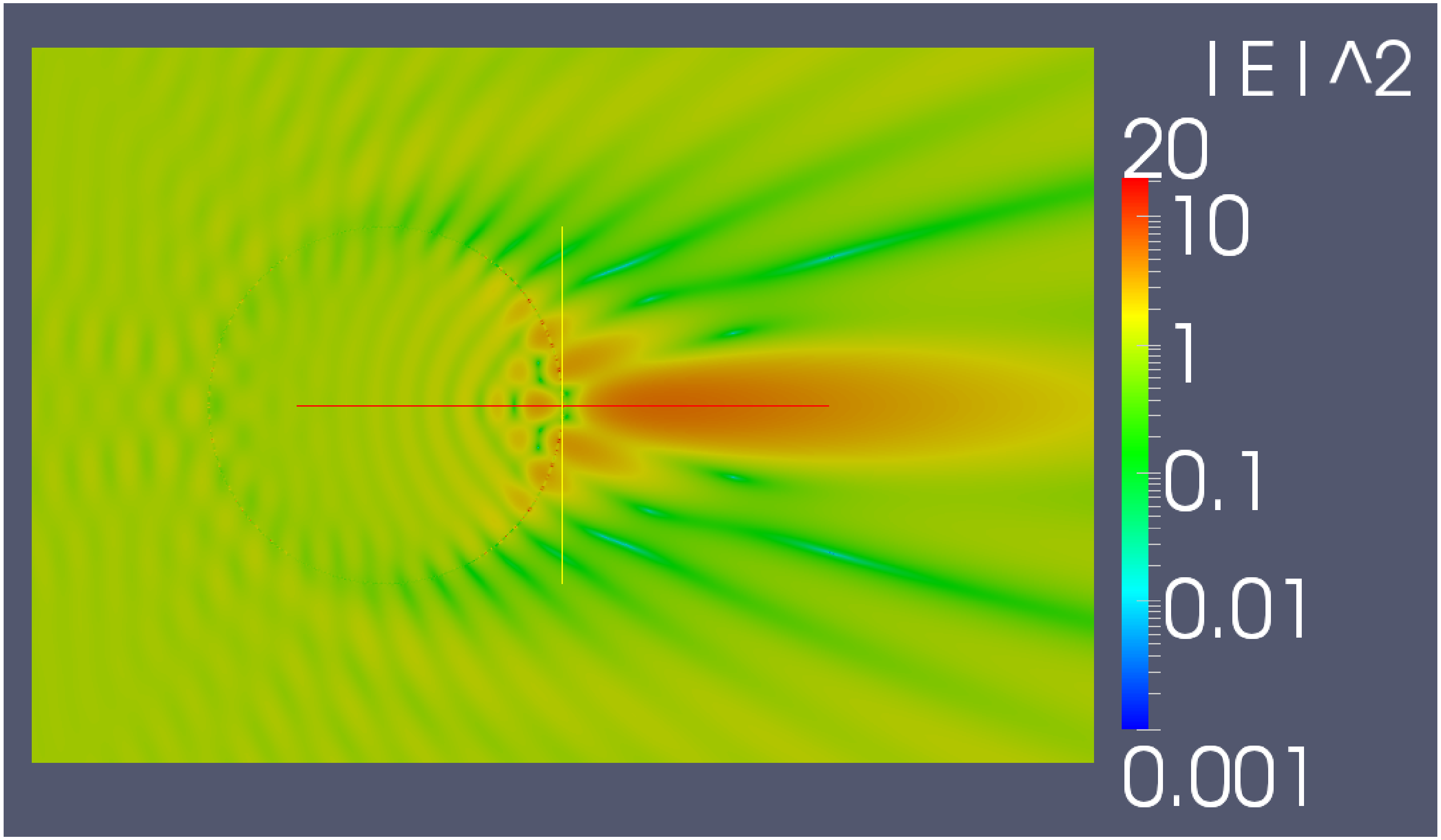
2

- 82,393 dofs
- 160s for operator assembly
- 198s for HLU preconditioner
- 56s solve time
- 198GMRES iterations

1

0

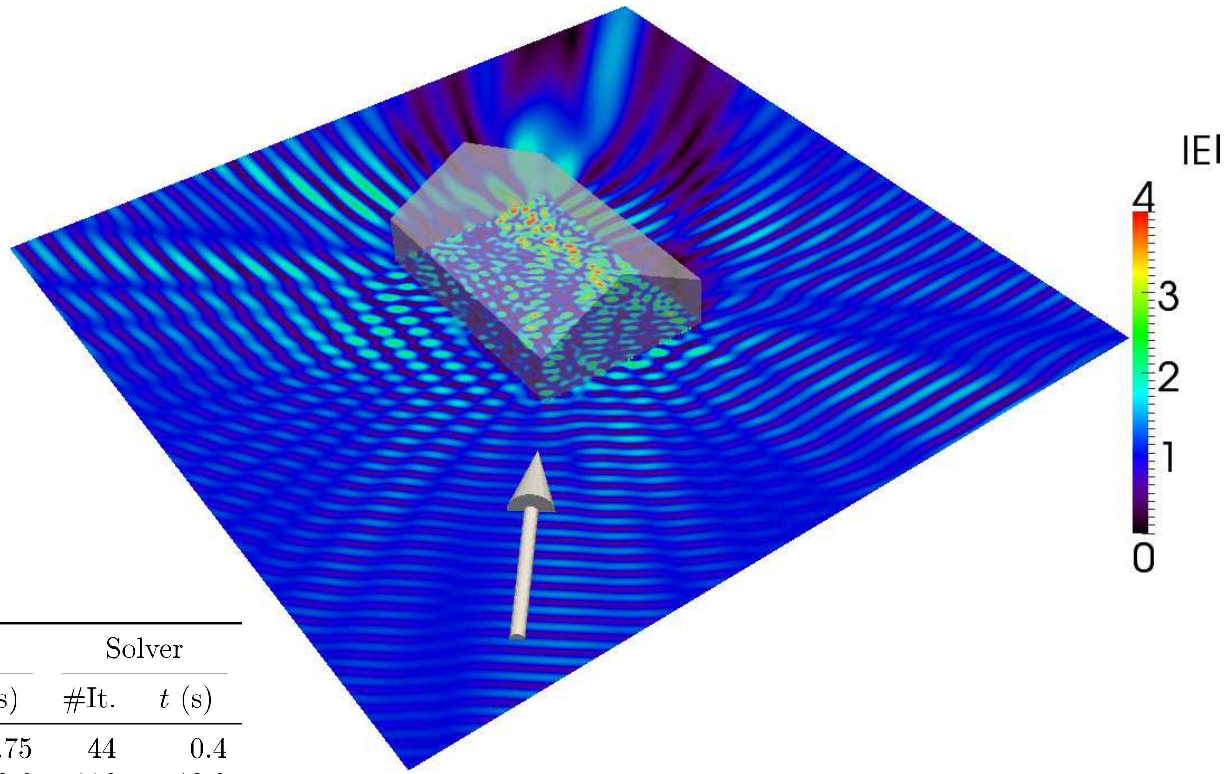
Simulating photonic jets



[Devilez et. al., 2008]

Scattering from an ice crystal

- Maxwell transmission
- H-Matrix compression
- HLU preconditioner



		$\boldsymbol{S}_{\text{int}}$		Preconditioner		Solver	
H/λ	#Elem.	Mem. (MB / %)	t (s)	Mem. (MB)	t (s)	#It.	t (s)
1	638	9.9 (100 %)	0.5	4.66	0.75	44	0.4
2	2718	123.6 (57 %)	7.5	49.1	8.6	116	12.0
4	10740	986.7 (25 %)	45.7	471.9	91.6	192	93.1
8	46016	7541 (11 %)	325.6	4605.6	1293.2	472	1463.0

$$\begin{bmatrix} \boldsymbol{S}_{\text{ext}} + \boldsymbol{S}_{\text{int}} & -(\frac{1}{2}(1-\rho^{-1})\boldsymbol{I} + \boldsymbol{C}_{\text{ext}} + \rho^{-1}\boldsymbol{C}_{\text{int}}) \\ \frac{1}{2}(1-\rho)\boldsymbol{I} + \boldsymbol{C}_{\text{ext}} + \rho\boldsymbol{C}_{\text{int}} & \boldsymbol{S}_{\text{ext}} + \boldsymbol{S}_{\text{int}} \end{bmatrix} \begin{bmatrix} \gamma_{\text{D,ext}}\boldsymbol{E} \\ \gamma_{\text{N,ext}}\boldsymbol{E} \end{bmatrix} = \begin{bmatrix} -\gamma_{\text{N,ext}}\boldsymbol{E}_{\text{inc}} \\ \gamma_{\text{D,ext}}\boldsymbol{E}_{\text{inc}} \end{bmatrix}$$

Maxwell – Crossing the Scales

- Loop-tree/Loop-star decompositions to overcome low-frequency instabilities [Andriulli ‘12]
- Advanced preconditioning techniques: Calderon preconditioners [Andriulli et al. 2008], OSRC [Bouajaji, Antoine, Geuzaine ‘14]
- High-Frequency Fast Multipole Methods [Darve, Greengard, Rohklin, Ying, …]
- High-Frequency fast direct solvers [Michielssen ‘13]

Convolution Quadrature for Time-Domain

[Banjai/Sauter '08]

$$\partial_t^2 u - \Delta u = 0 \quad \text{in } \Omega \times (0,T)$$
$$u(\cdot,0) = \partial_t u(\cdot,0) = 0 \quad \text{in } \Omega$$
$$u = g \quad \text{on } \Gamma \times (0,T)$$

Forward transform data boundary data (FFT):

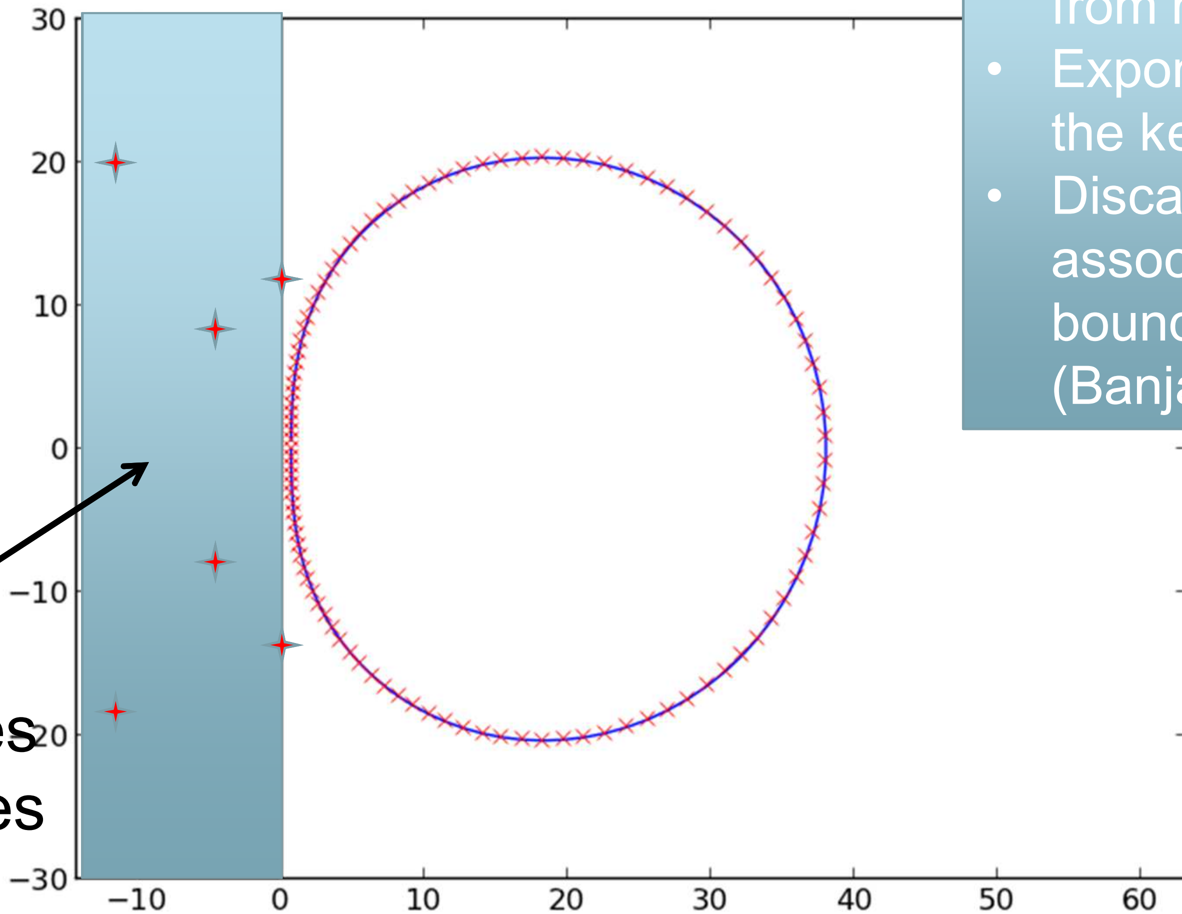
$$\hat{g}_l(x) = \sum_{n=0}^{N} \lambda^n g_n(x) \zeta_{N+1}^{-\ell n}$$

Solve in Laplace domain: $\mathbb{V}(s_\ell)\hat{\phi}_\ell(x) = \hat{g}_\ell(x)$

Inverse transform solution (IFFT):

$$\phi_n = \frac{\lambda^{-n}}{N+1} \sum_{\ell=0}^{N} \hat{\phi}_\ell \zeta_{N+1}^{n\ell}$$

Laplace domain wavenumbers

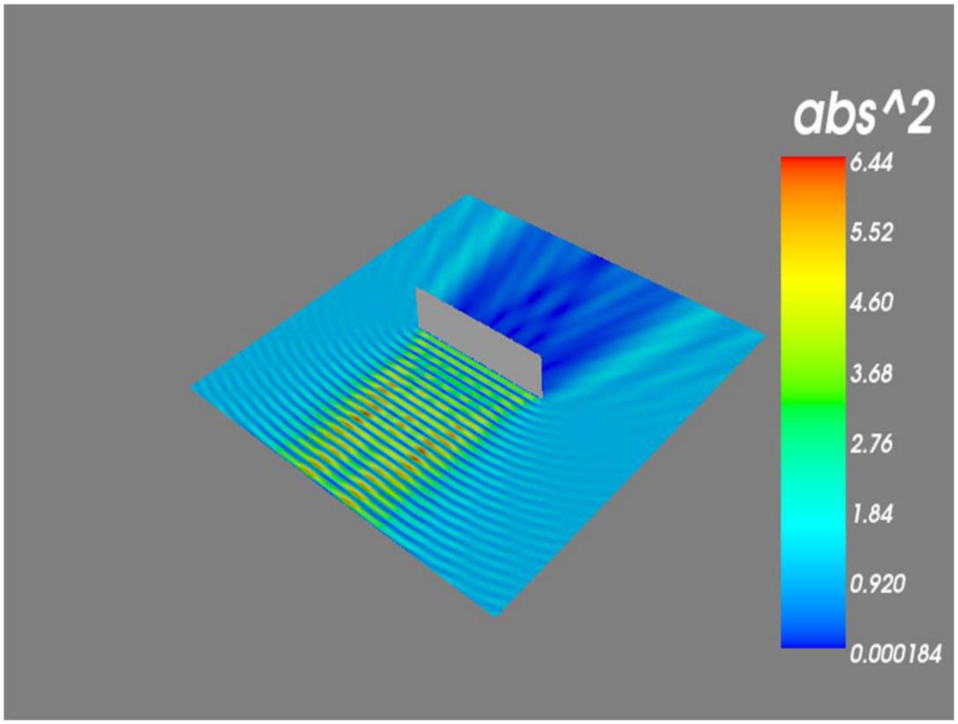


Scattering poles and resonances

- Wavenumbers away from resonances
- Exponential decay of the kernel
- Discard wavenumbers associated with small boundary data (Banjai/Sauter '08)

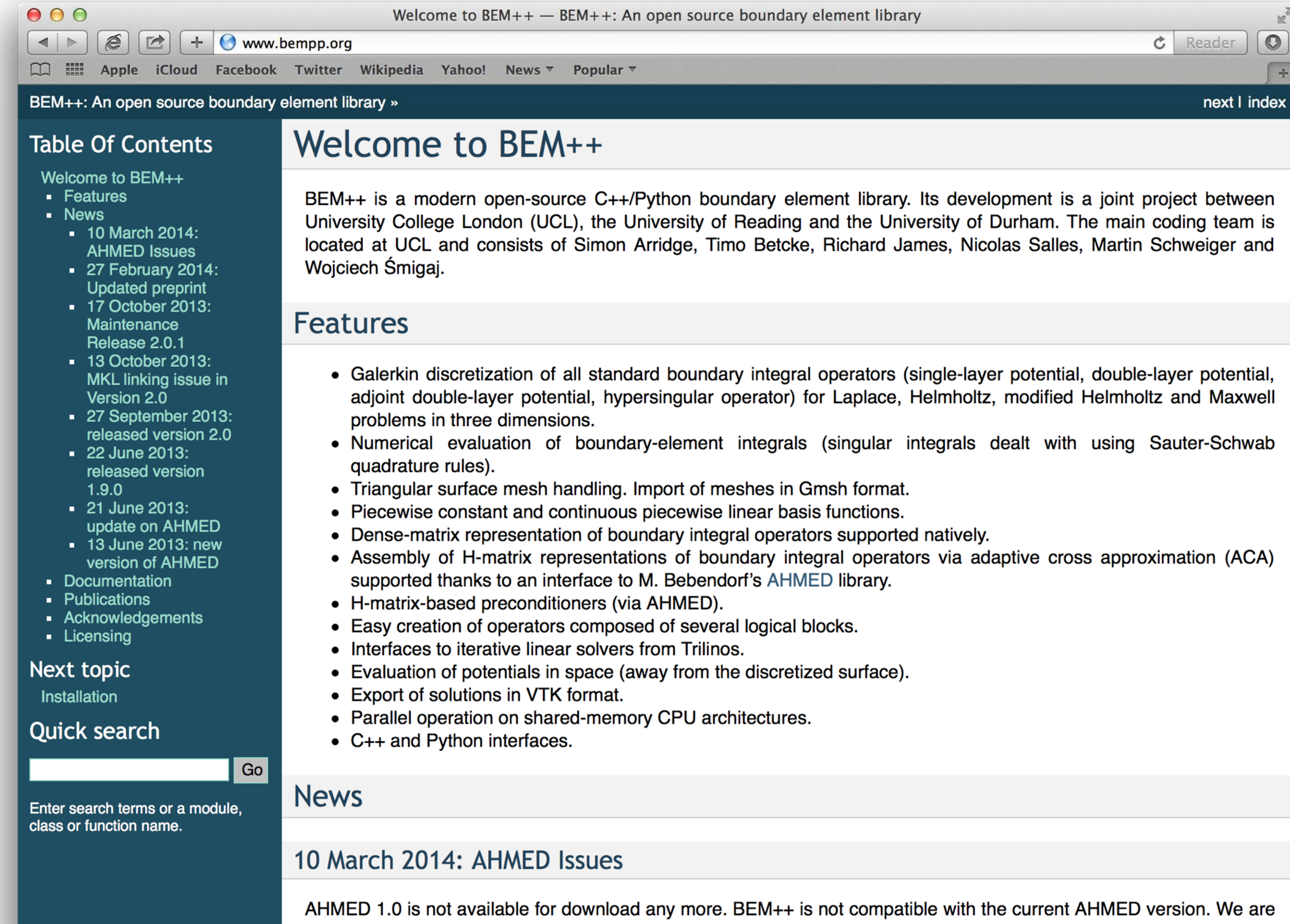
$$N = 100,\ \lambda = 0.93, \Delta t = 0.1$$

Time-Domain diffraction from a screen



Frequency Domain

Time Domain



Welcome to BEM++ — BEM++: An open source boundary element library

www.bempp.org

Apple iCloud Facebook Twitter Wikipedia Yahoo! News Popular

BEM++: An open source boundary element library » next | index

Table Of Contents

Next topic

Installation

Quick search

Go

Enter search terms or a module, class or function name.

Welcome to BEM++

BEM++ is a modern open-source C++/Python boundary element library. Its development is a joint project between University College London (UCL), the University of Reading and the University of Durham. The main coding team is located at UCL and consists of Simon Arridge, Timo Betcke, Richard James, Nicolas Salles, Martin Schweiger and Wojciech Śmigaj.

Features

- Galerkin discretization of all standard boundary integral operators (single-layer potential, double-layer potential, adjoint double-layer potential, hypersingular operator) for Laplace, Helmholtz, modified Helmholtz and Maxwell problems in three dimensions.
- Numerical evaluation of boundary-element integrals (singular integrals dealt with using Sauter-Schwab quadrature rules).
- Triangular surface mesh handling. Import of meshes in Gmsh format.
- Piecewise constant and continuous piecewise linear basis functions.
- Dense-matrix representation of boundary integral operators supported natively.
- Assembly of H-matrix representations of boundary integral operators via adaptive cross approximation (ACA) supported thanks to an interface to M. Bebendorf's AHMED library.
- H-matrix-based preconditioners (via AHMED).
- Easy creation of operators composed of several logical blocks.
- Interfaces to iterative linear solvers from Trilinos.
- Evaluation of potentials in space (away from the discretized surface).
- Export of solutions in VTK format.
- Parallel operation on shared-memory CPU architectures.
- C++ and Python interfaces.

News

10 March 2014: AHMED Issues

AHMED 1.0 is not available for download any more. BEM++ is not compatible with the current AHMED version. We are