

Ongoing developments for variational forms: coupled problems, parallelization

Xavier Claeys Axel Fourmont Frédéric Hecht Pierre Jolivet
Pierre Marchand Jacques Morice Pierre-Henri Tournier

FreeFEM Days, 14th Edition

December 8, 2022

- 1 vector BEM equation: Maxwell
- 2 Composite FE Spaces
- 3 Examples
- 4 FEM-BEM coupling example
- 5 Parallelization

Quick recap on the Boundary Element Method

Model problem

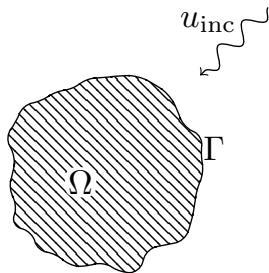
Volume form of the problem:

$$\begin{cases} -\Delta u - k^2 u = 0 & \text{in } \mathbb{R}^3 \setminus \Omega \\ u = -u_{\text{inc}} & \text{on } \Gamma \\ + \text{ radiation condition} \end{cases}$$

Green kernel: $\mathcal{G}(\mathbf{x}) = \exp(\imath k|\mathbf{x}|)/(4\pi|\mathbf{x}|)$

Single Layer Potential SL : $\forall q \in H^{-1/2}(\Gamma)$,

$$\text{SL}(q)(\mathbf{x}) = \int_{\Gamma} \mathcal{G}(\mathbf{x} - \mathbf{y}) q(\mathbf{y}) d\sigma(\mathbf{y}), \quad \forall \mathbf{x} \in \mathbb{R}^3 \setminus \Gamma$$



SL produces solutions of the PDE which satisfy the necessary conditions at infinity (here the Helmholtz equation and the Sommerfeld radiation condition)

\Rightarrow look for $p \in H^{-1/2}(\Gamma)$ such that $\text{SL}(p)(\mathbf{x}) = u(\mathbf{x})$ with $u = -u_{\text{inc}}$ on Γ

A variational formulation of the integral equation can be obtained by imposing the Dirichlet condition in a weak manner: find $p : \Gamma \rightarrow \mathbb{C}$ such that

$$\int_{\Gamma \times \Gamma} \frac{\exp(\imath k|\mathbf{x} - \mathbf{y}|)}{4\pi|\mathbf{x} - \mathbf{y}|} p(\mathbf{y}) q(\mathbf{x}) d\sigma(\mathbf{x}, \mathbf{y}) = - \int_{\Gamma} u_{\text{inc}}(\mathbf{x}) q(\mathbf{x}) d\sigma(\mathbf{x}) \quad \forall q : \Gamma \rightarrow \mathbb{C}$$

Quick recap on the Boundary Element Method

BEMTool library

BEMTool is a general purpose BEM library written by Xavier Claeys (LJLL). It is written in C++ and handles:

- Laplace, Yukawa, Helmholtz, Maxwell
- both in 2D and in 3D
- 1D, 2D and 3D triangulations (not necessarily flat)
- \mathbb{P}_k -Lagrange $k = 0, 1, 2$ and surface \mathbb{RT}_0

BEMTool is interfaced with FreeFEM.

It is available on GitHub  <https://github.com/xclaeys/BemTool>

Hierarchical matrices

Hierarchical block structure, Low-rank approximation

Let $\mathbf{B} \in \mathbb{C}^{N \times N}$ be a dense BEM matrix

quadratic cost in storage and complexity of the matrix-vector product

Low-rank approximation ?

BEM matrices do not have fast decreasing singular values,

BUT *near* the diagonal : *near-field* interactions

away from the diagonal : *far-field* \Rightarrow Green function very regularizing

Idea: build a hierarchical representation of the blocks of the matrix

identify and compress admissible blocks using low-rank approximation

\Rightarrow Build a low-rank approximation $\mathbf{B} \approx \sum_{j=1}^r \mathbf{u}_j \mathbf{v}_j^T$ (\sim approx. truncated SVD)

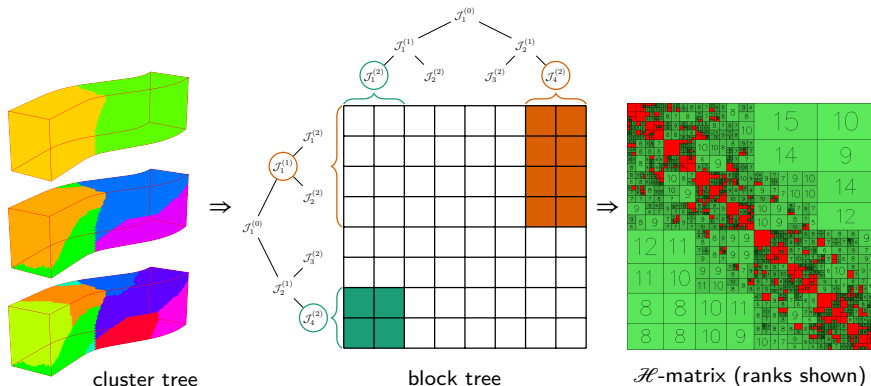
for each admissible block \mathbf{B} , using e.g. *Partially pivoted Adaptive Cross Approximation*, which needs only $2r$ rows/columns

Hierarchical matrices

Summary


- build a hierarchical, geometric clustering of the degrees of freedom
- traverse the block tree recursively
- geometric *admissibility condition*:

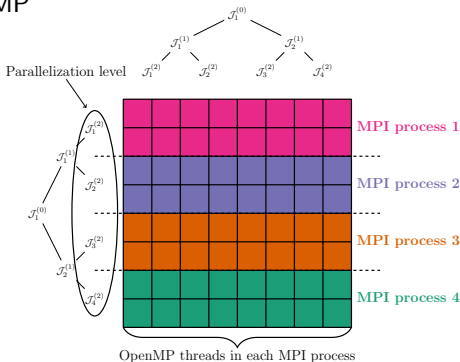
$\max(\text{diam}(X), \text{diam}(Y)) \leq \eta \text{dist}(X, Y) \Rightarrow$ compress the block



Hierarchical matrices

Htool library

- C++ library available on GitHub 
<https://github.com/PierreMarchand20/htool>
by Pierre Marchand and P.-H. T.
- interfaces with BEMTool for BEM kernels
- Parallel assembly, \mathcal{H} -matrix/vector and \mathcal{H} -matrix/matrix products using MPI and OpenMP



BEM variational forms in FreeFEM

Define the type of operator

$$-\Delta u - k^2 u = 0, \quad k \in \mathbb{C}$$

$k = 0$	Laplace
$k \in \mathbb{R}_+^*$	Helmholtz
$k \in i\mathbb{R}_+^*$	Yukawa

NEW Maxwell EFIE:

$k \in \mathbb{R}_+^*$ and surface \mathbb{RT}_0 space (RT0S)

Operators

BemKernel Ker ("SL", k=2*pi);

"SL"	Single Layer
"DL"	Double Layer
"HS"	Hyper Singular
"TDL"	Transpose Double Layer

Potentials

BemPotential Pot ("SL", k=2*pi);

"SL"	Single Layer
"DL"	Double Layer

BEM variational forms in FreeFEM

Define the problem

- Bilinear form on 3D surface mesh :

```
BemKernel Ker( "SL", k=2*pi);  
varf vbem(u,v) = int2dx2d(ThS) (ThS) (BEM(Ker,u,v));
```

or directly:

```
varf vbem(u,v) =  
int2dx2d(ThS) (ThS) (BEM(BemKernel( "SL", k=2*pi), u,v));
```

- Bilinear form on 2D curve mesh :

```
varf vbem(u,v) = int1dx1d(ThL) (ThL) (BEM(Ker,u,v));
```

- Assemble the HMatrix with *BEMTool* and *Htool* :

```
load "bem"  
HMatrix<complex> H = vbem(Uh,Uh);
```

BEM variational forms in FreeFEM

Solve the problem

```
fespace Uh(ThS,P1);  
Uh<complex> p, b;  
  
HMatrix<complex> H=vbem(Uh,Uh); // assemble the HMatrix
```

```
varf vrhs(u,v) = int2d(ThS)(finc*v);  
b[] = vrhs(0,Uh); // assemble the right-hand side
```

Solve the linear system with GMRES, with Jacobi preconditioner:

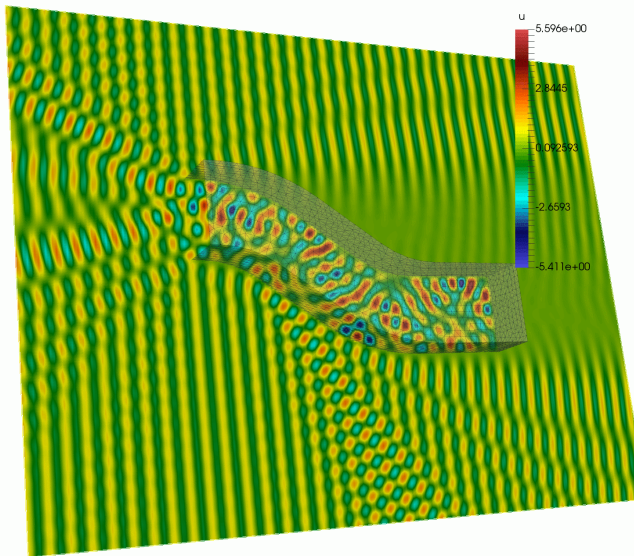
```
p[] = H^-1*b[];
```

Reconstruct the solution on an output mesh using the Potential:

```
varf vpot(u,v) = int2d(ThS)(POT(BemPotential("SL", k=2*pi),u,v));  
  
meshS ThOut = square3(50,50);  
fespace UhOut(ThOut,P1);  
UhOut<complex> u;  
  
HMatrix<complex> HP = vpot(Uh,UhOut);  
// reconstruct the field on every node of ThOut:  
u[] = HP*p[]; // p is the BEM solution  
plot(u);
```

Plane wave scattering by the COBRA cavity

- P1, 10 points per wavelength
- 33K dofs
- Dirichlet B.C.
- First kind formulation
- 94.4% compression
- two-level DD precondition, coarse mesh w/ 3.3 p.p. wavelength (3.7K dofs)
- assembly: 29.5s on 192 cores
- 7 gmres it (0.5s)
- radiation: 6.8s



Time-harmonic Maxwell's equations

Scattered field of a perfect electric conductor (PEC) object

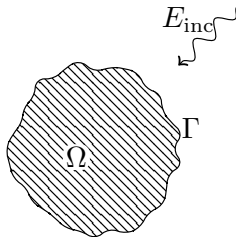
The scattered fields \mathbf{E} and \mathbf{H} for an homogeneous medium verify:

$$\operatorname{curl} \mathbf{E} + i\omega\mu_0 \mathbf{H} = 0, \quad \text{in } \mathcal{R}^3 \setminus \Omega$$

$$\operatorname{curl} \mathbf{H} + i\omega\epsilon_0 \mathbf{E} = 0, \quad \text{in } \mathcal{R}^3 \setminus \Omega$$

$$\mathbf{E} \wedge \mathbf{n} = -\mathbf{E}_{\text{inc}} \wedge \mathbf{n}, \quad \text{on } \Gamma$$

$$\lim_{r \rightarrow +\infty} r \left\| \sqrt{\epsilon_0} \mathbf{E} - \sqrt{\mu_0} \mathbf{H} \nabla \frac{r}{|r|} \right\| = 0.$$



Time convention: $\exp(i\omega t)$.

We introduce the total magnetic trace:

$$\mathbf{j} = i\kappa Z_0 \mathbf{n} \wedge (\mathbf{H} + \mathbf{H}_{\text{inc}}).$$

The total magnetic trace:

$$\mathbf{j} = i\kappa Z_0 \mathbf{n} \wedge (H + H_{\text{inc}})$$

Electric field integral equation (EFIE)

$$\begin{aligned} \int_{\Gamma \times \Gamma} \mathcal{G}_\kappa(\mathbf{x} - \mathbf{y}) (\mathbf{j}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{y}) - \kappa^{-2} \operatorname{div}_\Gamma \mathbf{j}(\mathbf{x}) \operatorname{div}_\Gamma \mathbf{v}(\mathbf{y})) d\sigma(\mathbf{x}) d\sigma(\mathbf{y}) \\ = - \int_\Gamma \mathbf{E}_{\text{inc}}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) d\sigma(\mathbf{x}) \end{aligned}$$

Green kernel: $\mathcal{G}_\kappa(\mathbf{x}) = \exp(i\kappa|\mathbf{x}|)/(4i\pi|\mathbf{x}|)$

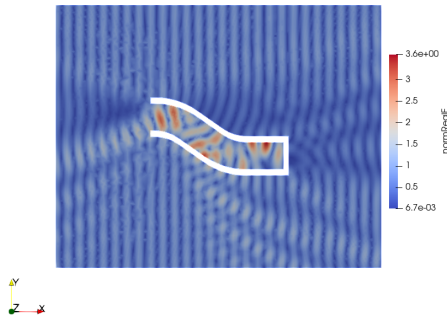
The scattered field \mathbf{E} is obtained with the Stratton-Chu Formula:

$$\mathbf{E}(\mathbf{y}) = \int_\Gamma \mathcal{G}_\kappa(|\mathbf{y} - \mathbf{x}|) \mathbf{j}(\mathbf{x}) d\sigma(\mathbf{x}) + \frac{1}{\kappa^2} \nabla_y \left(\int_\Gamma \mathcal{G}_\kappa(|\mathbf{y} - \mathbf{x}|) \operatorname{div}_\Gamma \mathbf{j}(\mathbf{x}) d\sigma(\mathbf{x}) \right).$$

vector BEM equation: Maxwell

Cobracavity

- Thickness around the cavity.
- $f = 5\text{GHz}$, RT0, 21204 dofs
- PEC material
- EFIE :
- assembly : 426 s
- 84 % compression
- Jacobi preconditioner,
8 MPI processes.
- solution j : 2743 gmres it (410s)
- potential E : 44 s
- total time 881s



Composite FE Space

Composite FE Space

Definition and example 1/2

- FE space is defined with a Finite Element (P1, P2, ...) and a mesh.

- Vector FE Space :

```
fespace Xh(Th, [P2,P2,P1]); // depends on one mesh
fespace XhPeriodic(Th, [P2,P2,P1], periodic=[[1,x],[3,x]]);
XhPeriodic [u1,u2,p]; // periodic for all components
```

- Definition of composite FE space :

$$X_h = U_h^1(T_h^1, FE_1) \times U_h^2(T_h^2, FE_2) \cdots \times U_h^n(T_h^n, FE_n),$$

where T_h^i and T_h^j can be different meshes (\times = cartesian product).

- Rewriting Vector FE space:

```
fespace Uh1(Th,P2);
fespace Uh2(Th,P2,periodic=[[1,x],[3,x]]);
fespace Ph(Th,P1);

fespace Xh(<Uh1,Uh2,Ph>); // Xh=Uh1 x Uh2 x Ph
```

- Remark: The notation with the angular bracket $\langle \cdot \rangle$ is not decided yet and can be changed in the future release of Freefem.

Composite FE Space

Definition and example 2/2

- Definition of composite FE space:

$$X_h = U_h^1(T_h^1, FE_1) \times U_h^2(T_h^2, FE_2) \cdots U_h^n(T_h^n, FE_n),$$

where T_h^i and T_h^j can be different meshes (\times = cartesian product).

- Rewriting Vector FE space:

```
fespace Uh1(Th,P2);  
fespace Uh2(Th,P2,periodic=[[1,x],[3,x]]);  
fespace Ph(Th,P1);  
  
fespace Xh(< Uh1,Uh2,Ph> ); // Xh = Uh1 x Uh2 x Ph
```

- FEM-BEM coupling:

```
fespace Xfem(Th,P1); // Th is a 2d volume mesh  
fespace Xbem(ThL,P1); // ThL is a curve mesh (meshL)  
  
fespace Xh(<Xfem,Xbem>); // Xh = Xfem x Xbem
```

- Remark: The notation with the angular bracket $\langle \cdot \rangle$ is not decided yet and can be changed in the future release of Freefem.

Composite FE space

problem/solve keyword

- Current way of defining a problem:

```
// definition of the FE spaces
fespace Uh1 (Th1,P2); fespace Uh2 (Th2,P1);
fespace Vh1 (Th3,P2); fespace Vh2 (Th4,P1);
// definition of FE functions
Uh1 u1; Uh2 u2;
Vh1 v1; Vh2 v2;

problem myPB ([u1,u2],[v1,v2]) = int2d(Th1) (...)
                                + int2d(Th2) (...)
                                - int1d(Th1) (...) + on (...);

myPB;
```

Limitations:

- All FE spaces are defined on the *same* mesh ($Th1=Th2=\dots=Th4$).
- Uh1 and Vh1 (resp. Uh2 and Vh2) are the *same* FE space.
- *same* = computer memory

⇒ Rewrite of problem/solve function in the kernel with composite FE space to bypass these limitations.

Composite FE space

Alternative: using varf

- In current FreeFEM, we can bypass these limitations by using the **varf** keyword.
- Mixed variational form: $U_h1=V_h1$, $U_h2=V_h2$ are different FE spaces.

// definition of the FE spaces

fespace Uh1(Th1,P2);

fespace Uh2(Th2,P1); *// Th1 and Th2 are different meshes*

Uh1 u1,v1; *// definition of FE functions*

Uh2 u2,v2;

varf vfA11(u1,v1) = **int2d**(Th1) (**dx**(u1)***dx**(v1) + **dy**(u1)***dy**(v1))
+ **int2d**(Th1) (f1*v1) + **on** (1,u1=0);

matrix A11 = vfA11(Uh1,Uh1); *// Interaction between Uh1 and Uh1*

matrix A21 = vfA21(Uh1,Uh2); *// Interaction between Uh1 and Uh2*

matrix A12 = vfA12(Uh2,Uh1); *// Interaction between Uh2 and Uh1*

matrix A22 = vfA22(Uh2,Uh2); *// Interaction between Uh2 and Uh2*

matrix A = [[A11,A12], [A21,A22]]; *// global matrix*

real[int] rhs = [rhs1,rhs2]; *// where real[int] rhs1=vfA11(0,Vh1);*

real[int] sol = A⁻¹*rhs; *// solution of the system*

Limitations:

- A variational form for each block of the system for mixed variational form.
- ⇒ The goal of composite FE space is to write one varf in this case.

Composite FE space

problem/solve keyword

- New formulation with composite FE spaces:

```
// definition of the FE spaces
fespace Uh(Th1, [P2,P2]);
fespace Ph(Th2,P1);
// definition of FE functions
Uh [u1,u2]; Uh [v1,v2];
Ph p; Ph q;
// use angle bracket "<" ">" for composite
// "[" "]" define a FE function
// Remark: this notation is a placeholder so that we can easily
switch between old and composite during development
// ⇒ So it may change in the future release of Freefem
problem myCompositePB(<[u1,u2],[p]>,<[v1,v2],[q]>) = ...
myCompositePB;

plot( [u1,u2] ); // u1 and u2 contain the solution of the problem
```

- allows for different meshes, and even different mesh types (mesh, meshS, meshL, ...)
- Uh1 and Vh1 (resp. Uh2 and Vh2) are the same FE space for the moment.

Example with composite FE Space

Thermal contact

Consider the following thermal contact problem:

$$\left\{ \begin{array}{l} \nabla(k_i \nabla u_i) = f_i \quad \text{in } \Omega_i, \\ k_i \nabla u_i \cdot \mathbf{n} = 0, \text{ on upper and lower boundary of } \Omega_i, \\ u_1 = T_1 \text{ for } x = 0, \\ u_2 = T_2 \text{ for } x = L. \end{array} \right.$$

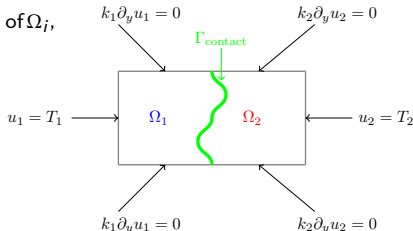
- Γ_{contact} is the boundary between the two domains:

$$k_1 \nabla u_1 \cdot \mathbf{n}_1 = -k_2 \nabla u_2 \cdot \mathbf{n}_2 \quad \text{in } \Gamma_{\text{contact}}.$$

- hypothesis on the thermal flux in the contact boundary: $k_1 \nabla u_1 \cdot \mathbf{n}_1 = c(u_2 - u_1)$

- variational form:

$$\forall (v_1, v_2), \left\{ \begin{array}{l} \int_{\Omega_1} k_1 \nabla u_1 \cdot \nabla v_1 dX + c \int_{\Gamma_{\text{contact}}} (u_1 - u_2) v_1 d\sigma(X) = 0, \\ \int_{\Omega_2} k_2 \nabla u_2 \cdot \nabla v_2 dX + c \int_{\Gamma_{\text{contact}}} (u_2 - u_1) v_2 d\sigma(X) = 0. \end{array} \right.$$



Example with composite FE Space

Thermal contact

- variational form:

$$\forall (v_1, v_2), \begin{cases} \int_{\Omega_1} k_1 \nabla u_1 \cdot \nabla v_1 dX + c \int_{\Gamma_{\text{contact}}} (u_1 - u_2) v_1 d\sigma(X) = 0, \\ \int_{\Omega_2} k_2 \nabla u_2 \cdot \nabla v_2 dX + c \int_{\Gamma_{\text{contact}}} (u_2 - u_1) v_2 d\sigma(X) = 0. \end{cases}$$

```
fespace Vh1(Th1,P1); // definition of FE space
```

```
fespace Vh2(Th2,P1);
```

```
Vh1 u1,v2; // definition of the FE function
```

```
Vh2 u2,v2;
```

```
// use angle bracket "<" ">" for composite
```

```
problem compositeThermal(<[u1],[u2]>,<[v1],[v2]>) =
```

```
int2d(Th1) (k1*dx(u1)*dx(v1) + k1*dy(u1)*dy(v1))
```

```
+ int2d(Th2) (k2*dx(u2)*dx(v2) + k2*dy(u2)*dy(v2))
```

```
+ int1d(Th1,10) (c*(u1-u2)*v1)
```

```
+ int1d(Th2,10) (c*(u2-u1)*v2)
```

```
+ on(3,u1=T0) + on(4,u2=T1);
```

```
compositeThermal;
```

```
// u1 and u2 contains the solution of the problem
```

Example with composite FE Space

Thermal contact

conclusion for problem/solve:

- composite FE spaces allow u_1 and u_2 to lie in different domains Ω_1 and Ω_2 .
 - composite FE spaces allow to use non conformal meshes for Ω_1 and Ω_2 .
⇒ The only relation between Ω_1 and Ω_2 is due to the coupled interface problem.
 - composite FE spaces allow to have different kind of physics in different domains.
- ⇒ all above points also work with composite FE spaces using [varf](#).

Example with composite FE Space

Stokes example

Consider the following Stokes problem:

$$\begin{cases} -\Delta u + \nabla p = f & \text{in } \Omega \\ -\operatorname{div} u = 0 & \text{in } \Omega \\ u = u_D & \text{on } \Gamma_D \end{cases}$$

- The variational form is given by:

$$\forall (v, q), \begin{cases} \int_{\Omega} \nabla u \cdot \nabla v \, dX - \int_{\Omega} p \operatorname{div}(v) \, dX = \int_{\Omega} f \cdot v \, dX, \\ - \int_{\Omega} \operatorname{div}(u) q \, dX - \int_{\Omega} \epsilon p q = 0. \end{cases}$$

- (u, p) is solution $\Rightarrow (u, p + C)$ is also a solution.
- stabilized term (blue term) allows to have only one solution $\epsilon = 1e-10$.

Composite FE Space

Stokes example

- variational form:

$$\forall (v, q), \begin{cases} \int_{\Omega} \nabla u \cdot \nabla v \, dX - \int_{\Omega} p \operatorname{div}(v) \, dX = \int_{\Omega} f \cdot v \, dX, \\ - \int_{\Omega} \operatorname{div}(u) q \, dX - \int_{\Omega} \epsilon p q = 0. \end{cases}$$

- stabilized term (blue term) allows to have only one solution $\epsilon = 1e-10$.
- The choice of the fespace for u and p must be chosen carefully (LBB condition)

$$\forall u \in U_h, \quad \sup_{p \in P_h} \frac{\int_{\Omega} \nabla p \cdot \nabla u \, dX}{\|p\|_{P_h}} \geq \beta \|u\|_{U_h},$$

- $T_h^u = T_h^p$ and P1(u)-P1(p) FE doesn't work.
- $T_h^u = T_h^p$ and P2-P1 FE (Taylor-Hood Element)

Composite FE Space

Stokes Example : driven cavity

- The driven cavity is a standard test. It is a box full of liquid with its lid moving horizontally at speed one.
- upper boundary $u = (1,0)$ and otherwise $u = (0,0)$.
- we have also added a hole inside with zero velocity at the boundary.

`composite_stokes_cavite_entraine.e``dp`

Conclusion:

- Composite FE spaces allow to use different meshes for the same domain.

Composite FE Space

Stokes Example : periodic in one direction.

$$\begin{cases} -\Delta u + \nabla p = f & \text{in } \Omega \\ -\operatorname{div} u = 0 & \text{in } \Omega \end{cases}$$

- In this test, we suppose that we know the solution:
- $u = (\sin(x)\cos(y), -\cos(x)\sin(y))$
- $p = 2\cos(x)\cos(y)$
- $f = (0, -4\cos(x)\sin(y))$

`composite_stokes_periodic_problem.edp`

Conclusion:

- Composite FE spaces allow to use a periodic condition on one component.

FEM-BEM coupling

Model problem

Consider the following Helmholtz problem:

$$\left\{ \begin{array}{ll} -\Delta u - n(\mathbf{x})k^2 u = f & \text{in } \Omega_{\text{int}} \\ -\Delta u_{\text{ext}} - n_{\text{ext}}k^2 u_{\text{ext}} = 0 & \text{in } \Omega_{\text{ext}} \\ u = u_{\text{ext}} & \text{on } \Gamma \\ \nabla u \cdot \mathbf{n} = \nabla u_{\text{ext}} \cdot \mathbf{n} & \text{on } \Gamma \\ + \text{ radiation condition} \end{array} \right.$$

Variational form in Ω_{int} : $\forall v$,

$$\int_{\Omega_{\text{int}}} \nabla u \nabla v - \int_{\Omega_{\text{int}}} n k^2 u v - \int_{\Gamma} \nabla u \cdot \mathbf{n} v = \int_{\Omega_{\text{int}}} f v$$

FEM-BEM coupling

Bielak-MacCamy formulation

$$\int_{\Omega_{\text{int}}} \nabla u \nabla v - \int_{\Omega_{\text{int}}} n k^2 u v - \int_{\Gamma} \nabla u_{\text{ext}} \cdot \mathbf{n} v = \int_{\Omega_{\text{int}}} f v$$

Bielak-MacCamy indirect coupling: find ansatz ϕ s.t. $u_{\text{ext}} = SL(\phi)$ in Ω_{ext}

We have

$$\begin{cases} \gamma_D(u) = \gamma_D(u_{\text{ext}}) & = \gamma_D \circ SL(\phi) \\ \nabla u \cdot \mathbf{n} = \nabla u_{\text{ext}} \cdot \mathbf{n} & = -\gamma_N \circ SL(\phi) \end{cases}$$

with γ_D and γ_N the Dirichlet and Neumann trace operators.

Thus, the coupled problem is: find (u, ϕ) such that $\forall (v, \psi)$

$$\begin{cases} \int_{\Omega_{\text{int}}} \nabla u \nabla v - \int_{\Omega_{\text{int}}} n k^2 u v + \int_{\Gamma} \gamma_N \circ SL(\phi) v = \int_{\Omega_{\text{int}}} f v \\ \int_{\Gamma} \gamma_D(u) \psi - \int_{\Gamma} \gamma_D \circ SL(\phi) \psi = 0 \end{cases}$$

FEM-BEM coupling

Bielak-MacCamy formulation

$$\begin{cases} \int_{\Omega_{\text{int}}} \nabla u \nabla v - \int_{\Omega_{\text{int}}} n k^2 u v + \int_{\Gamma} \gamma_N \circ SL(\phi) v = \int_{\Omega_{\text{int}}} f v \\ \int_{\Gamma} \gamma_D(u) \psi - \int_{\Gamma} \gamma_D \circ SL(\phi) \psi = 0 \end{cases}$$

In the current version of Freefem, no mixed formulation in FreeFEM: assemble blocks separately

⇒ With Composite FE spaces: only one coupled variational form

```
varf Lenses([uf,ub],[vf,vb]) =  
  int2d(Th)((-k^2*uf*vf+Grad(uf)'*Grad(vf))) // Fem  
+ int1dx1d(ThL)(ThL)(BEM(-1*BemKernel("SL",k=k),ub,vb)) // -SL  
+ int1d(ThL)(uf*vb) // Mass  
+ int1d(ThL)(0.5*ub*vf) + int1dx1d(ThL)(ThL)(BEM(BemKernel(  
  "TDL",k=k),ub,vf)); // TDL
```

Note: the bilinear form associated to the *transpose double layer operator* is

$$a_{K'}(\phi, v) = \frac{1}{2} \int_{\Gamma} \phi v + \int_{\Gamma} K'(\phi) v := \int_{\Gamma} (\gamma_N \circ SL)(\phi) v$$

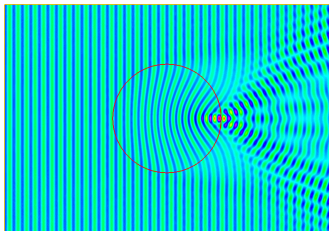
FEM-BEM coupling

Application: gradient index lenses

Wave focusing and bending using gradient index lenses

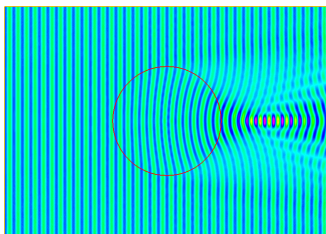
Maxwell's fish-eye lens (1854)

$$n(r) = \frac{2}{1 + \left(\frac{r}{R}\right)^2}$$



Luneburg lens (1944)

$$n(r) = \sqrt{2 - \left(\frac{r}{R}\right)^2}$$



Helmholtz-2d-FEM-BEM-coupling-MUMPS.edp

Composite FE space

Variational form

Helmholtz-2d-FEM-BEM-coupling-MUMPS.edp

- 35 lines for the definition of the matrix and the rhs.
- Matrix system is

$$\begin{pmatrix} F & TDL \\ mass & -SL \end{pmatrix} \begin{pmatrix} u_{fem} \\ u_{bem} \end{pmatrix} = \begin{pmatrix} F_{rhs} \\ 0 \end{pmatrix} \quad \text{wheresol} = \begin{pmatrix} u_{fem} \\ u_{bem} \end{pmatrix}.$$

```
fespace Uh(Th,P1); // Th is a 2d volume mesh
fespace UhL(ThL,P1); // ThL is a meshL
fespace CUh(<Uh,UhL>); // CUh = Uh x UhL

varf Lenses([ufem,ubem],[vfem,vbem]) =
int2d(Th)((-ind*k^2*ufem*vfem+Grad(ufem)'*Grad(vfem))) // F
+ int1dx1d(ThL)(ThL)(BEM(BemKernel("TDL",k=k),ubem,vfem)) // TDL
+ int1d(ThL)(0.5*ubem*vfem) // TDL
+ int1d(ThL)(ufem*vbem) // mass
+ int1dx1d(ThL)(ThL)(BEM(-1*BemKernel("SL",k=k),ubem,vbem)) // -SL
+ int2d(Th)(finc*vfem) + on(waveguide,ufem=0); // RHS

matrix<complex> A = Lenses(CUh,CUh);
complex[int] rhs = Lenses(0,CUh);
```


Composite FE space

Variational form

```
fespace Uh(Th,P1); // Th is a 2d volume mesh
fespace UhL(ThL,P1); // ThL is a meshL
fespace CUh(<Uh,UhL>); // CUh = Uh × UhL
...
varf Lenses([ufem,ubem],[vfem,vbem]) = ...

matrix<complex> A = Lenses(CUh,CUh);
complex[int] rhs = Lenses(0,CUh);

complex[int] sol = A^-1*rhs;
```

- In the version in development, FE function of composite FE space is not defined:

```
CUh [ufem,ubem]; // compile error
```

- As $CUh = Uh \times UhL$, we have $sol = \begin{pmatrix} u_{fem} \\ u_{bem} \end{pmatrix}$.

```
Uh<complex> ufem; // ufem
UhL<complex> ubem; // ubem
// method 1: extract solution
ufem[] = sol(0:(Uh.ndof)-1); // extract ufem
ubem[] = sol(Uh.ndof:(sol.n-1)); // extract ubem
// method 2: dispatch solution
[ufem[], ubem[]] = sol;
```

Ultimate ~~Dream~~ Goal:

run any sequential script in parallel on N processes \Rightarrow same result, fully distributed data structures (completely hidden to the user), speedup of N

What we are working on for now:

- Hide the parallelization of the assembly and solution steps (main bottlenecks) as much as possible
- Generalize the matrix format, using PETSc for now ; useful for handling composite operators stemming from coupled problems (e.g. sparse + \mathcal{H} -matrices for FEM-BEM)

Parallel FreeFEM

Parallel assembly and MUMPS distributed solver

Hide parallel assembly, give distributed matrix to MUMPS:

```
load "MUMPS"  
solve pb(u,v,solver=sparsec_solver,master=-1) = int3d(  
    Th) (Grad(u)'*Grad(v)) -int3d(Th) (f*v) +on(1,u=2);
```

or

```
load "MUMPS"  
varf pb(u,v) = int3d(Th) (Grad(u)'*Grad(v)) +int3d(Th  
    ) (f*v) +on(1,u=2);  
  
matrix A = pb(Uh,Uh,solver=sparsec_solver,master=-1);  
real[int] rhs = pb(0,Uh);  
  
u[] = A^-1*rhs;
```

Parallel FreeFEM

PETSc matrices

Also works with composite FE spaces for coupled problems, e.g. FEM-BEM lenses:

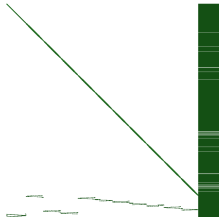
```
load "MUMPS"
varf Lenses([ufem,ubem],[vfem,vbem]) =
int2d(Th)((-ind*k^2*ufem*vfem+Grad(ufem)'*Grad(vfem))) // F
+ int1dx1d(ThL)(ThL)(BEM(BemKernel("TDL",k=k),ubem,vfem)) // TDL
+ int1d(ThL)(0.5*ubem*vfem) // TDL
+ int1d(ThL)(u1*v2) // mass
+ int1dx1d(ThL)(ThL)(BEM(-1*BemKernel("SL",k=k),ubem,vbem)) // -SL
+ int2d(Th)(finc*vfem) + on(waveguide,ufem=0) ; // RHS

matrix<complex> A = Lenses(CUh,CUh,solver=sparsesolver, master=-1);
complex[int] rhs = Lenses(0,CUh);

complex[int] sol = A^-1*rhs;
```

BUT for now, densifies the compressed BEM blocks (\mathcal{H} -matrices)

⇒ use PETSc instead, construct a distributed *nested* Mat, each block has its own matrix type



We can then take advantage of the wide range of available solvers in PETSc. In particular, we can easily use *fieldsplit* preconditioners for coupled problems, each variable naturally defining its own *field*:

```
load "PETSc-complex"
varf Lenses([ufem,ubem],[vfem,vbem]) =
int2d(Th)((-ind*k^2*ufem*vfem+Grad(ufem)'*Grad(vfem))) // F
+ int1dx1d(ThL)(ThL)(BEM(BemKernel("TDL",k=k),ubem,vfem)) // TDL
+ int1d(ThL)(0.5*ubem*vfem) // TDL
+ int1d(ThL)(u1*v2) // mass
+ int1dx1d(ThL)(ThL)(BEM(-1*BemKernel("SL",k=k),ubem,vbem)) // -SL
+ int2d(Th)(finc*vfem) + on(waveguide,ufem=0); // RHS

Mat<complex> A = Lenses(CUh,CUh);
set(A,sparams="-ksp_view_-ksp_monitor_-ksp_type_fgmres
-ksp_view_final_residual_-ksp_gmres_restart_200_-pc_type_fieldsplit_
-fieldsplit_0_pc_type_asm_-fieldsplit_0_sub_pc_type_lu
-fieldsplit_0_ksp_type_gmres_-fieldsplit_1_ksp_type_gmres
-fieldsplit_1_ksp_max_it_20");

complex[int] rhs = Lenses(0,CUh);
complex[int] sol = A^-1*rhs;
```

- use PETSc as a solver directly in problem/solve:

```
solve pb(u,v,solver=petsc,sparams="...");
```

- C++ implementation of Domain Decomposition macros (macro_ddm.idp) to hide difficulties to the user \Rightarrow will be able to use DD solvers such as GenEO transparently
- developments in Htool: \mathcal{H} -LU factorization, more compression techniques (Block ACA, HCA, randomized SVD, ...)