Domain decomposition techniques for interfacial discontinuities

Geordie McBain

Institut Jean le Rond d'Alembert

December 5, 2012

Connexion and substructuring

- Two ways to look at domain decomposition
  - Decomposing domains by iterative substructuring
  - Linking subdomains through interfacial conditions
- Small easy generalizations of techniques for the former mean they can be used for the latter.

Queries on `freefempp` mailing list

- **Electrical simulation with a voltage jump at an intermediate layer** (2011-05-20)
  My questions are:
  1) Is it possible with FreeFem++ to apply a constant voltage jump at the surface where both cubes are touching and how to do it?
  2) If possible, how to insert a resistive 2D sheet at this surface (current dependent voltage jump)?
- **Non linear heat transfer equation and Thermal Contact Resistance (TCR)** (2012-08-15)
  how could we simulate the Thermal contact resistance between two materials?

Steady thermal conduction

- For definiteness, consider the steady conduction of heat

$$-\nabla \cdot (\mathrm{K} \nabla T) = s$$

where

K conductivity
T temperature
s volumetric rate of generation

- Weak form $(\forall U)$:

$$\langle \nabla U, \mathrm{K} \nabla T \rangle - [U, \hat{\mathbf{n}} \cdot \mathrm{K} \nabla T] - \langle U, s \rangle = 0$$

Natural interfacial conditions

- basic interfacial conditions (like Kirchhoff's circuit laws):
  - equality of temperature (like potential at node)
  - zero sum of heat (like current into node)
- $$T_1 = T_2$$
  $$\hat{\mathbf{n}}_1 \cdot \mathsf{K}_1 \nabla T_1 + \hat{\mathbf{n}}_2 \cdot \mathsf{K}_2 \nabla T_2 = 0$$
- Often have $\mathsf{K}_1 \neq \mathsf{K}_2$ in applications.

Classical engineering heat transfer problem I

Given a two-layer furnace wall of thickness and conductivity L_i and K_i ($i = 1, 2$), if the inside is at T_1 and the outside at T_2, what is the temperature distribution?

Dirichlet–Dirichlet solutions, given T_{int}:

$$\text{subdomain solutions:}\quad \begin{aligned} T_1 - T_{\mathrm{int}} &= L_1 q_1 / K_1 \\ T_{\mathrm{int}} - T_2 &= L_2 q_2 / K_2 \end{aligned}$$

Poincaré–Steklov Dirichlet→Neumann operator:

$$q_1 - q_2 = \frac{T_1 K_1}{L_1} + \frac{T_2 K_2}{L_2} - \left(\frac{K_1}{L_1} + \frac{K_2}{L_2} \right) T_{\mathrm{int}}$$

$$\text{Solution:}\quad T_{\mathrm{int}} = \left(\frac{T_1 K_1}{L_1} + \frac{T_2 K_2}{L_2} \right) \div \left(\frac{K_1}{L_1} + \frac{K_2}{L_2} \right)$$

Classical engineering heat transfer problem II

Given a two-layer furnace wall of thickness and conductivity L_i and K_i ($i = 1, 2$), if the inside is at T_1 and the outside at T_2, how much heat q is lost?

Neumann–Neumann solutions, given q:

$$\text{subdomain solutions:} \quad \begin{aligned} T_1 - T_{\text{int}}^- &= L_1 q / K_1 \\ T_{\text{int}}^+ - T_2 &= L_2 q / K_2 \end{aligned}$$

Poincaré–Steklov Neumann→Dirichlet operator:

$$T_{\text{int}}^+ - T_{\text{int}}^- = T_2 - T_1 + \left(\frac{L_1}{K_1} + \frac{L_2}{K_2} \right) q$$

$$\text{Solution:} \quad q = \frac{T_1 - T_2}{\frac{L_1}{K_1} + \frac{L_2}{K_2}}$$

Neumann–Neumann domain decomposition

- This idea generalizes to finite element solutions.
- Guess the flux at the interface.
- Solve Neumann problem on either subdomain.
- Compute the temperature mismatch at interface.
- Iterate to find the flux eliminating the mismatch.

Generalized interfacial conditions

- various matching criteria:
  - specified jump: $T_1 = T_2 + \Delta T$
  - nonlinear algebraic: $f(T_1, T_2) = 0$
    - e.g. equilibrium chemical partitioning between phases
  - contact resistance: $T_1 = T_2 + Rq$
  - general nonlinear: $f(T_1, T_2, q) = 0$
- Any of these fit the Neumann–Neumann framework.
- Just 'solve' this equation for the interfacial flux.
- Noting that $T_1 = T_1(q)$ and $T_2 = T_2(q)$.

Generalized interfacial conditions (cont.)

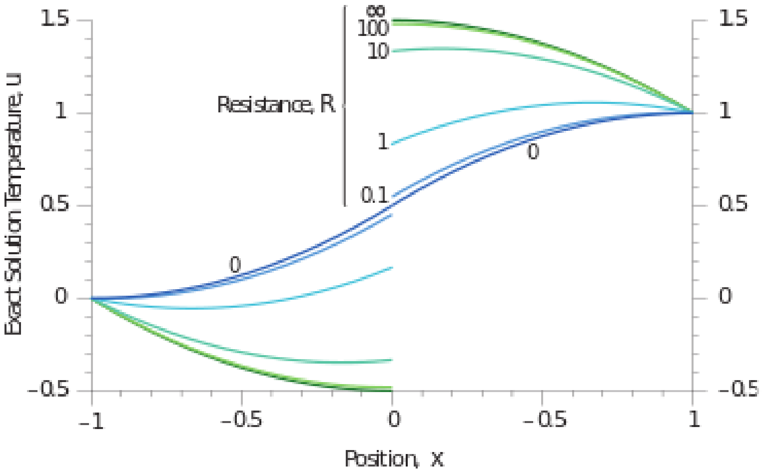
- Could also have a superficial power source.
- Then there's a jump in the flux.
- Method should still apply
  - guess the flux on one side
  - calculate the flux on the other from the condition
  - solve the two Neumann problems
  - etc.

Example of Carnes & Copps (2008)

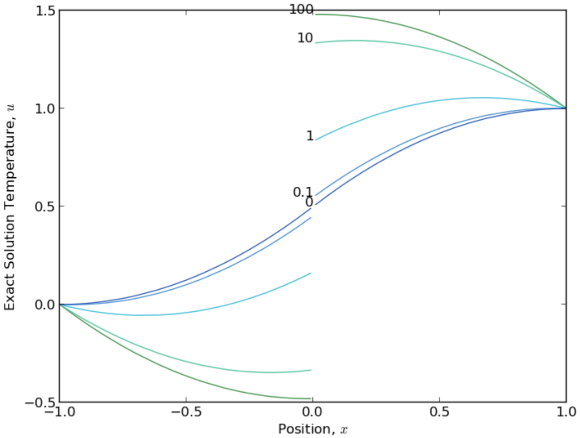
Carnes, B. R., & K. D. Copps (2008) Thermal contact algorithms. Sandia Report SAND2008-2607

$$
\begin{aligned}
\text{D.E.:} \quad & -\nabla^2 T = \operatorname{sgn} x \qquad (-1 < x < 1) \\
\text{Dirichlet conditions:} \quad & T(-1) = 0, \quad T(1) = 1 \\
\text{flux balance:} \quad & -T'(0^-) = -T'(0^+) \\
\text{interfacial resistance:} \quad & -RT'(0) = T(0^-) - T(0^+)
\end{aligned}
$$

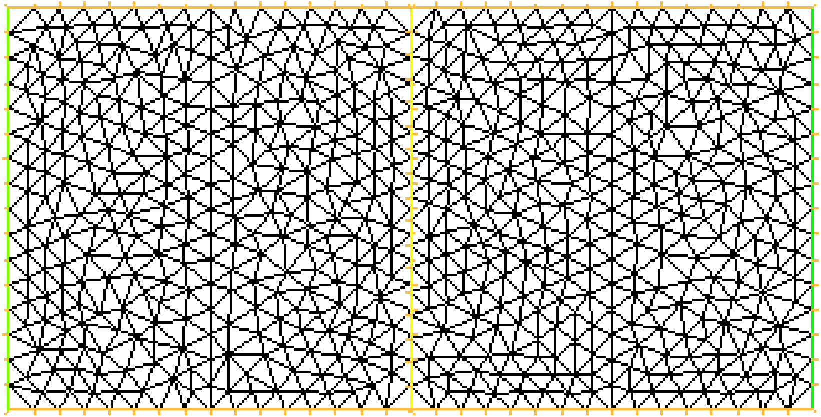
Results of Carnes & Copps (2008)



Reproduction in FreeFem++



Geometry and mesh



Solve in two dimensions (to investigate mismatched meshes).

Implementation in FreeFem++

- See § 9.8.3 Schwarz-gc.edp.
- define interfacial flux q on one whole subdomain
  - values meaningless off interface
- solve Neumann problem for each subdomain
- compute interfacial temperature mismatch
- iterate using conjugate gradients

Subdomain Neumann problems

```
problem Pb1 (u1, v1, init=i) =
  int2d (Th1) dot(grad(v1), grad(u1))
  + int2d (Th1) (-v1*s1)
  + int1d (Th1, inner) (+q*v1)
  + on (left, u1=0);
problem Pb2 (u2, v2, init=i) =
  int2d(Th2) dot(grad(v2), grad(u2))
  + int2d (Th2) (-v2*s2)
  + int1d (Th2, inner) (-q*v2)
  + on (right, u2=1);
```

Computing the mismatch $\Delta T - Rq$

```
varf b (u1, v1) = int1d (Th1, inner) (v1 * u1);
matrix B = b (Vh1, Vh1);

func real[int] BoundaryProblem (real[int] &l) {
   q[] = l;
   Pb1; Pb2; i++;
   v1 = u1 - u2; // temperature jump
   real[int] q1 = B * v1[];
   v1 = R * q; // resistive jump
   real[int] q2 = B * v1[];
   q[] = q1 - q2;
   return q[];
};
```

Finding the correct interfacial flux

```
Vh1 p=0;
LinearCG (BoundaryProblem, p[]);
BoundaryProblem (p[]);
```

Neumann–Neumann method

- A particular advantage of the Neumann–Neumann method:
  - We don't have to compute interfacial fluxes. Even with an interfacial resistance, for which the flux appears in the matching condition, we can use the guessed interfacial flux.
- Having to compute fluxes on each subdomain then transfer to where iteration variable was defined would have been fiddly.
- Thus a Dirichlet–Dirichlet method is less convenient.
  - not attempted here

Summary of extension on § 9.8.3

Instead of just calculating the temperature jump, also subtract resistance times the guessed flux.

An axially symmetric problem

- Instead of Carnes & Copps (2008) one-dimensional solution
- we propose an axially symmetric one.
- Still has a simple exact solution
- less trivial realization in two-dimensional finite elements
- Also tests curved interfaces.

The axially symmetric problem

$$\begin{aligned}
\text{D.E.:} \qquad & -\nabla\cdot(K_1\nabla T_1) = s_1 \qquad (r_0 < r < r_1) \\
& -\nabla\cdot(K_2\nabla T_2) = s_2 \qquad (r_1 < r < r_2) \\
\text{Dirichlet conditions:} \quad & T(r_0) = T_0 \\
& T(r_2) = T_2 \\
\text{flux balance:} \qquad & -T'(r_1^-) = -T'(r_1^+) \\
\text{interfacial resistance:} \qquad & -RT'(r_1) = T(r_1^-) - T(r_1^+)
\end{aligned}$$

General analytical solution of Neumann problem

- Dirichlet condition at one radius $T(r_i) = T_i$
- Guessed Neumann condition at another $-K_i T'(r_{\rm int}) = q_{\rm int}$
- source s_i and conductivity K_i constant per annular layer
- centre excluded: $r_i > 0$

$$T(r) = T_i + \frac{s_i}{2}\left(r_{\rm int}^2 \ln\frac{r}{r_i} + \frac{r_i^2 - r^2}{2}\right) - \frac{r_{\rm int}\, q_{\rm int}}{K_i} \ln\frac{r}{r_i}$$

General analytical solution for Poincaré–Steklov operator I

- flux q for specified contact resistance R
- two layers, $r_0 < r < r_{\text{int}}$ and $r_{\text{int}} < r < r_1$

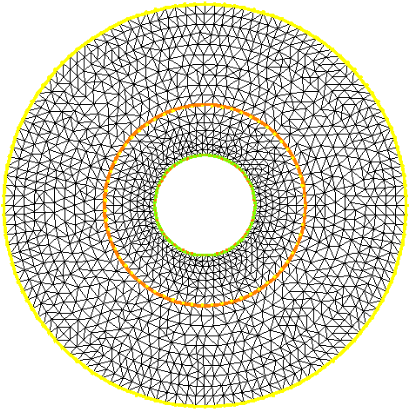
$$q = \frac{T_0 - T_1 - \frac{s_1}{2K_1}\left(r_{\text{int}}^2 \ln \frac{r_{\text{int}}}{r_1} + \frac{r_1^2 - r_{\text{int}}^2}{2}\right) + \frac{s_0}{2K_0}\left(r_{\text{int}}^2 \ln \frac{r_{\text{int}}}{r_0} + \frac{r_0^2 - r_{\text{int}}^2}{2}\right)}{R - r_{\text{int}}\left(\frac{\ln \frac{r_{\text{int}}}{r_1}}{K_1} - \frac{\ln \frac{r_{\text{int}}}{r_0}}{K_0}\right)}$$

General analytical solution for Poincaré–Steklov operator II

- flux q for specified temperature jump ΔT
- two layers, $r_0 < r < r_{\mathrm{int}}$ and $r_{\mathrm{int}} < r < r_1$

$$q = \frac{T_0 - T_1 - \Delta T - \frac{s_1}{2K_1}\left(r_{\mathrm{int}}^2 \ln \frac{r_{\mathrm{int}}}{r_1} + \frac{r_1^2 - r_{\mathrm{int}}^2}{2}\right) + \frac{s_0}{2K_0}\left(r_{\mathrm{int}}^2 \ln \frac{r_{\mathrm{int}}}{r_0} + \frac{r_0^2 - r_{\mathrm{int}}^2}{2}\right)}{r_{\mathrm{int}}\left(\frac{\ln \frac{r_{\mathrm{int}}}{r_1}}{K_1} - \frac{\ln \frac{r_{\mathrm{int}}}{r_0}}{K_0}\right)}$$

Geometry and mesh



Curved interfaces necessitate finer meshes.

Test problem:

- $s_0 = 1, s_1 = 0$
- $K_0 = 1, K_1 = 2$
- $r_0 = \frac{1}{2}, r_1 = 2$
- $r_{\text{int}} = 1$

Calculating mismatches

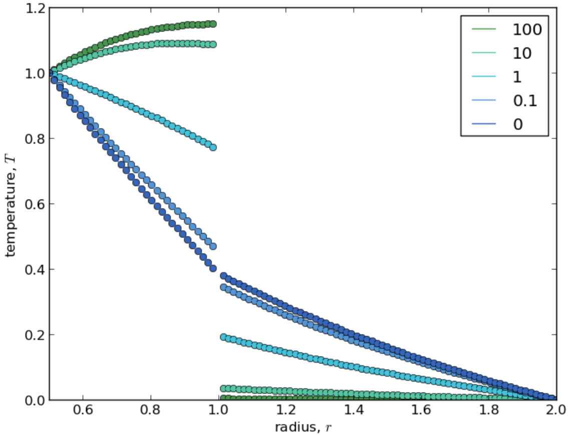
- contact resistance R:

```
v0 = u0 - u1; // temperature jump
real[int] q0 = B * v0[];
v0 = R * q; // resistive jump
real[int] q1 = B * v0[];
q[] = q0 - q1;
return q[];
```

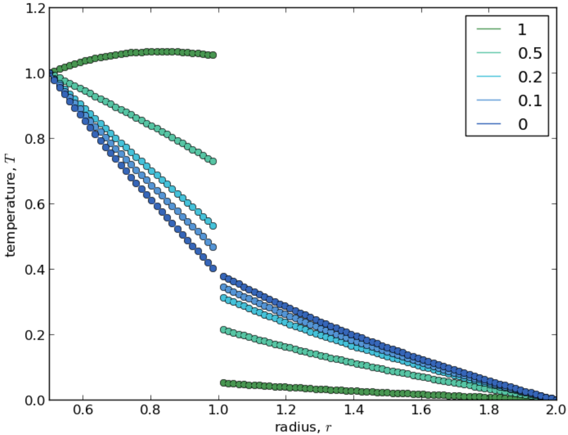
- jump ΔT:

```
v0 = u0 - (u1 + DT); // temperature jump shortfall
real[int] q0 = B * v0[];
return q0;
```

Results for various contact resistances



Results for various temperature jumps



Conclusion

- Many kinds of interfacial discontinuity can be handled easily in FreeFem++.
- Use domain decomposition.
- Guess the flux across the interface (Neumann–Neumann).
- Compute the mismatch (Neumann→Dirichlet Poincaré–Steklov).
- Iterate to find the flux giving the right mismatch.