PETSc finite element details

Geoffrey Irving
Otherlab

November 17, 2013

1 Weak form PDEs

Consider a strong form PDE with fields $u_0, u_1, \ldots$, each of which has components $u_{i0}, u_{i1}, \ldots$, and equations

$$f_{ab}\left(u_{ij}, \partial_k u_{ij}, \partial_k \partial_l u_{ij}\right) = 0$$

where we have one f_{ij} for each u_{ij}. We use the convention that an index occurring only inside a function call means that all values of the index are passed to the function. When we later discretize over space, u_i and u_j will be allowed to live in different discrete function spaces, while u_{ij} and u_{ik} come the same function space.

We pass to the weak form by integrating a smooth test function v_{ab} against f_{ab}, then integrating by parts to get

$$\int_\Omega v_{ab} f_{ab}\left(u_{ij}, \partial_k u_{ij}, \partial_k \partial_l u_{ij}\right) dx = 0$$
$$\int_\Omega v_{ab} f^0_{ab}\left(u_{ij}, \partial_k u_{ij}\right) dx + \int_\Omega \partial_c v_{ab} f^1_{abc}\left(u_{ij}, \partial_k u_{ij}\right) dx = 0$$

where f^0 and f^1 depend on u only to first derivatives. The pointwise functions f^0 and f^1 are exposed to petsc, with u_{ij} passed as an (i,j)-major array and $\partial_k u_{ij}$ as an (i,j,k)-major array. f^0_{ab} is split into functions f^0_a each returning a b-major array, and f^1_{abc} is split into functions f^1_a each returning a (b,c)-major array. In weird pseudocode signatures,

```
#define AUX a[i,j], da[i,j]/dx[k], x[k]
f0[a] : (u[i,j], du[i,j]/dx[k], AUX) -> [b];
f1[a] : (u[i,j], du[i,j]/dx[k], AUX) -> [b,c];
```

where x_k is position and a_{ij} are auxiliary fields.

2 Boundary conditions

With boundary conditions, the weak form of the PDE becomes

$$\int_{\Omega} v_{ab} f^{0}_{ab} dx + \int_{\Omega} \partial_c v_{ab} f^{1}_{abc} dx + \int_{\delta\Omega} v_{ab} f^{0\partial}_{ab} da + \int_{\delta\Omega} \partial_c v_{ab} f^{1\partial}_{abc} da = 0$$

where the boundary condition terms $f^{0\partial}$ and $f^{1\partial}$ additionally depend on outward pointing boundary normals n_k. The pseudocode signatures are the same as for f^0 and f^1 except for the dependence on normals:

```
f0b[a] : (..., n[k]) -> [b]
f1b[a] : (..., n[k]) -> [b,c]
```

3 Jacobians

The above is sufficient for computing residuals. For residual Jacobians, we additionally need first derivatives of f^0 and f^1 w.r.t. u and ∇u, for a total of four additional functions g^0, g^1, g^2, g^3. Theoretically we might also need derivatives of the boundary terms, but this is currently unsupported.

We split the 4-valued indexed g^{γ} into $g^{\alpha\beta}$ where $\gamma = 2\alpha + \beta$. $g^{\alpha\beta}$ represents the Jacobian of f^{α} w.r.t. the βth derivatives of u:

$$\begin{aligned} g^{00}_{aibj} &= \frac{d}{du_{ij}} f^{0}_{ab} \\ g^{01}_{aibjk} &= \frac{d}{d\partial_k u_{ij}} f^{0}_{ab} \\ g^{10}_{aibcj} &= \frac{d}{du_{ij}} f^{1}_{abc} \\ g^{11}_{aibcjk} &= \frac{d}{d\partial_k u_{ij}} f^{1}_{abc} \end{aligned}$$

Each $g^{\alpha\beta}$ is split into a two dimensional array of functions $g^{\alpha\beta}_{ai}$ giving function f^{α}_{a} differentiated against the βth derivatives of u_i. In signatures:

```
g^{00}[a,i] : (...) -> [bj];
g^{01}[a,i] : (...) -> [bjk];
g^{10}[a,i] : (...) -> [bcj];
g^{11}[a,i] : (...) -> [bcjk];
```