Mappings between centering points

I believe that POOMA needs to provide better support for expressions involving fields with different centerings.

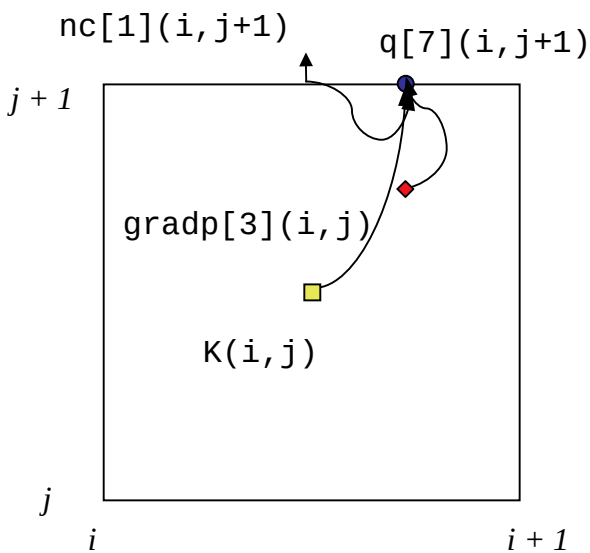
Consider the deceptively simple flux equation:

$$q = \vec{\vec{\mathbf{K}}} \cdot \nabla P \cdot \mathbf{n}_c$$

Each of the fields in this expression has a different centering.The flux is face-centered at multiple positions, the permeability is cell-centered, the coordinate normals are face centered, and the pressure gradient is cell-centered at multiple positions.

We know how to write out the POOMA discretization in terms of indices. Assume the top, right flux point is orientation 7 and the top right pressure gradient point is orientation 3:

```
q[7](i,j+1) = dot(dot(K(i,j), gradp[3](i,j)), nc[1](i,j+1))
```



If we were evaluating at different orientations of *q*, we would find that we would need to re-use field values. For example, the same permeability is used to evaluate all orientations of *q* associated with cell *i,j*. Similarly, a single value coordinate normal contributes to 2 *q* orientations as does a single value of the pressure gradient.Getting this to work right in a data-parallel form is a trick.

Jeffrey realized that values need to be replicated and interpolated. I agree with him that these are the right abstractions. However, as I mentioned before, I do argue that the replication must depend on centering. This is clear when trying to map cell-centered data to faces in 3D: the counting just doesn't work out. Nor is there any easy way to assure that the right points are mapped: for example, the upper-right `gradp` point maps to the *q* centering point on the right hand side of the top face and the top of the right hand face.

To me this seems like a job for conversion functions:

```
q = dot(dot(cellToMultiFace(K), multiCellToMultiFace(gradp)),
       faceToMultiFace(nc))
```

where the counting obviously works out right if the outputs of these three functions are fields on *q*'s centering. Moreover, I think that we already have most of the technology to implement this in the form of Field stencil. in fact, the "stencils" I'm talking about to do this mapping may be sufficiently simple that we can provide a simpler user interface.

Field stencil is already designed to do interpolations like accumulating the total flux in a cell. I like the notion that these two abstractions—replication and interpolation—could be implemented using the same technology.

What do people think about this?