Current and Planned AMR Support in PETSc

Toby Isaac
tisaac@ices.utexas.edu

The University of Chicago at Austin

June 18, 2015

motivation

why adapt?

1. Your non-adaptive calculations have reached the end of your resources (or the end of weak-scalability), and you want to push back.
2. You have a performance model that predicts it can help.

hp-FEM theory

predicts exponential convergence in N_{dof}:

- If we want zero error, it's worth it.
- If we have a nonzero tolerance, we must consider that hp systems require more resources per dof to solve than uniform, low-order systems. There is always a crossover.

wishlist

data structures

- a good interface for the user to specify local mesh sizes
  - " " " " " " " " " local metric tensors
- composable mesh formats
- configurability

numerical methods

- *a posteriori* mesh refinement and coarsening

The latter requires the former. The former clearly lives in DM, but the latter doesn't fit anywhere at the moment:

- KSP, SNES, TS, TAO: the dimension is fixed.
- PetscQoI_AMR (hypothetical): compile years of research into *a posteriori* methods.

mesh formats

DA

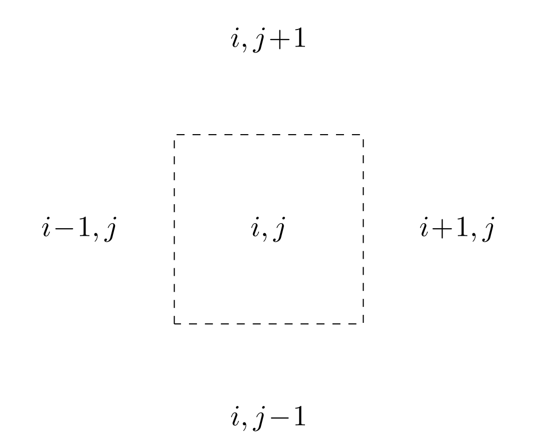
Plex

Performance ⟷ Generality

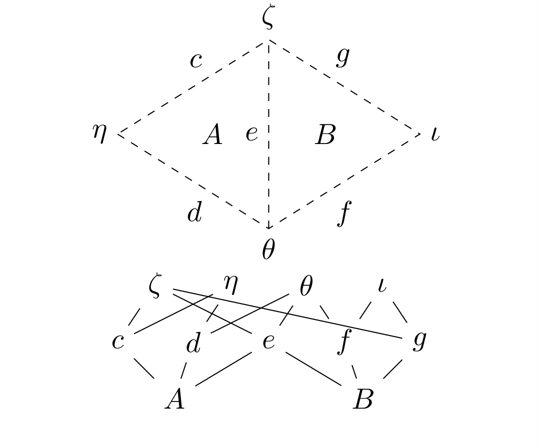
mesh formats

Who is my neighbor?

DA



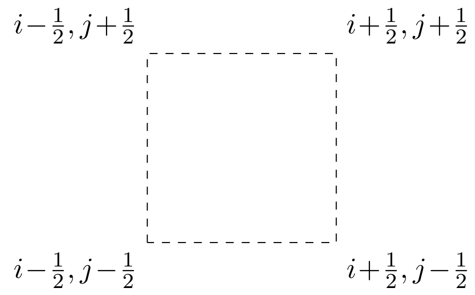
Plex



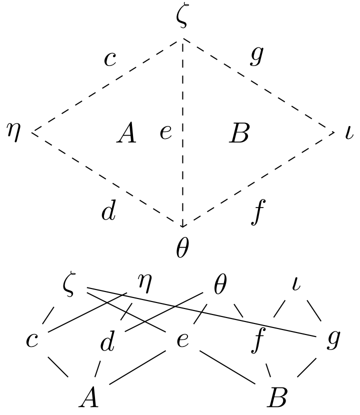
mesh formats

Who is my neighbor?

DA


$i-\frac{1}{2},j+\frac{1}{2}$
$i+\frac{1}{2},j+\frac{1}{2}$
$i-\frac{1}{2},j-\frac{1}{2}$
$i+\frac{1}{2},j-\frac{1}{2}$

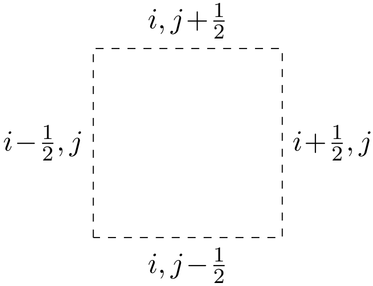
Plex



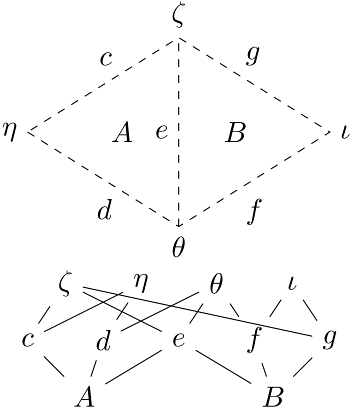
mesh formats

Who is my neighbor?

DA



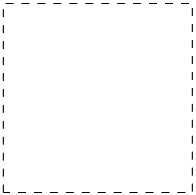
Plex



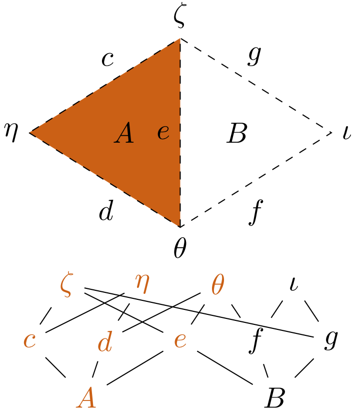
mesh formats

Who is my neighbor?

DA



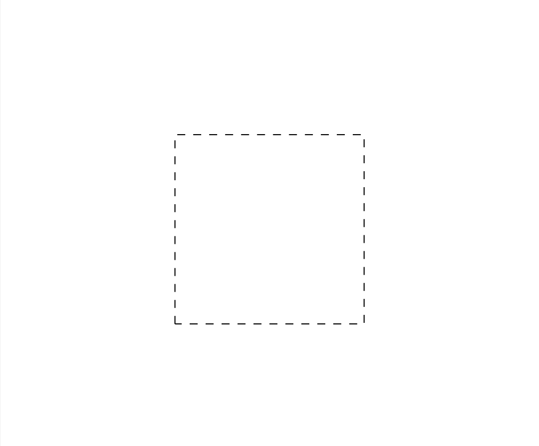
Plex



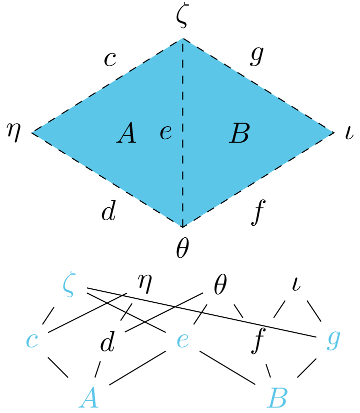
mesh formats

Who is my neighbor?

DA



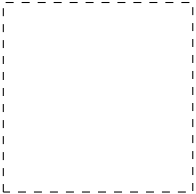
Plex



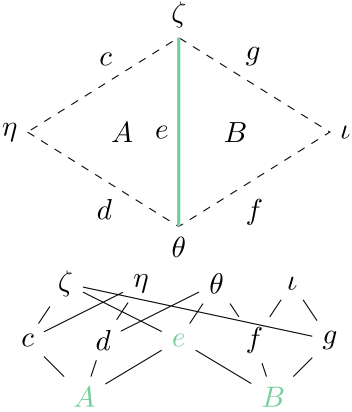
mesh formats

Who is my neighbor?

DA



Plex

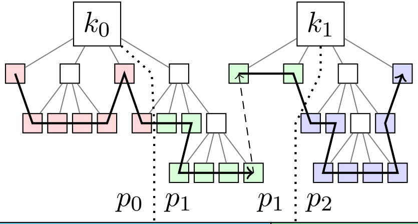


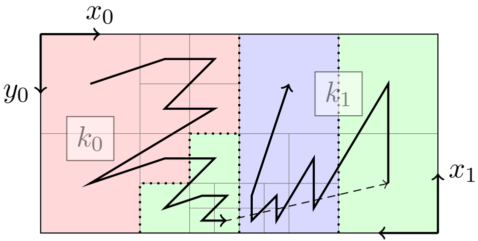
p4est

Overview

The p4est library (p4est.org, lead developer Carsten Burstedde) provides scalable AMR routines via a *forest-of-octrees/quadtrees*:

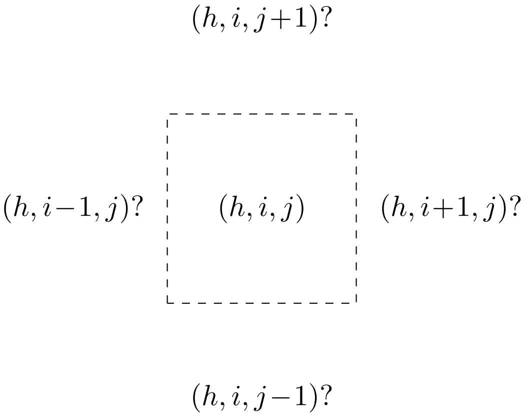
- a unstructured hexahedral mesh (“the forest”);
- where each hexahedron contains an arbitrarily refined octree;
- space-filling curve (SFC) orders elements;
- philosophy: as-simple-as-possible coarse mesh describes geometry, refinement captures all detail.





p4est

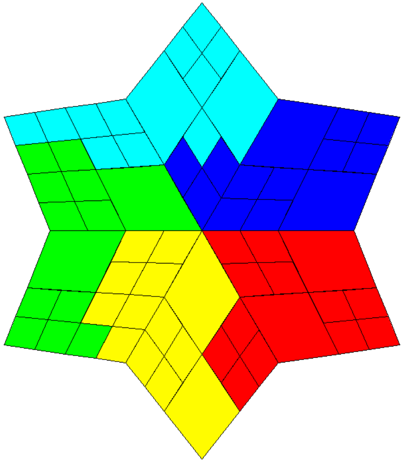
p4est: linear octree (pointerless octants)



$(h, i, j+1)?$

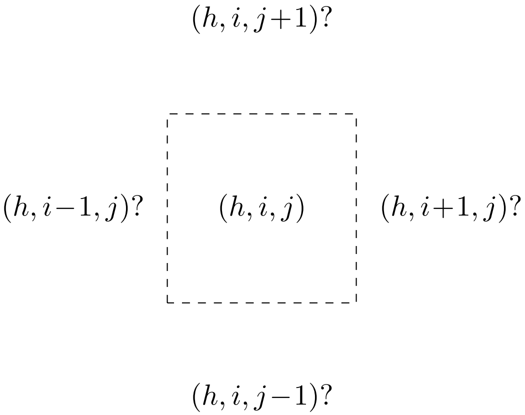
$(h, i-1, j)?$ (h, i, j) $(h, i+1, j)?$

$(h, i, j-1)?$



p4est

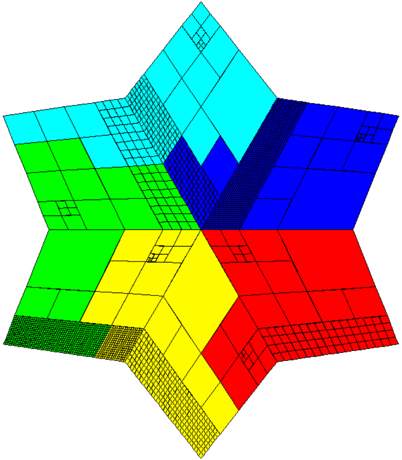
p4est: linear octree (pointerless octants)



$(h, i, j{+}1)?$

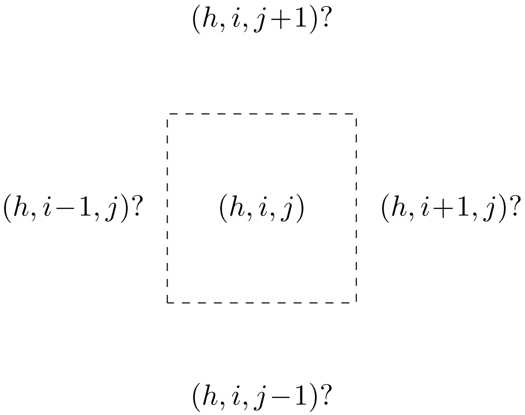
$(h, i{-}1, j)?$ (h, i, j) $(h, i{+}1, j)?$

$(h, i, j{-}1)?$



p4est

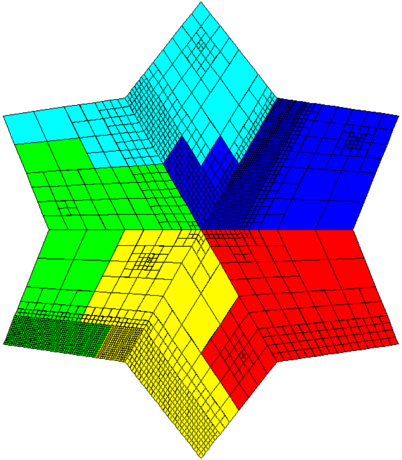
p4est: linear octree (pointerless octants)



$(h, i, j+1)$?

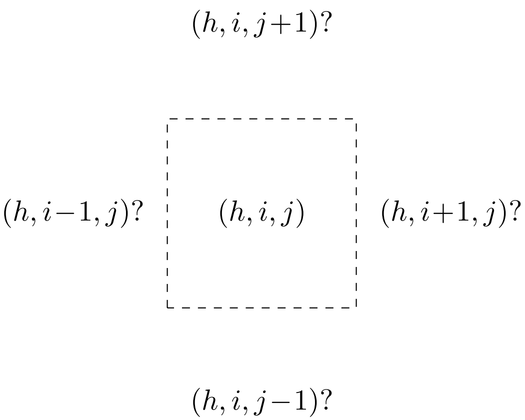
$(h, i-1, j)$? (h, i, j) $(h, i+1, j)$?

$(h, i, j-1)$?



p4est

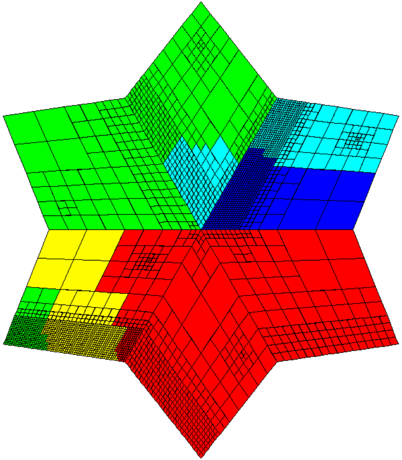
p4est: linear octree (pointerless octants)

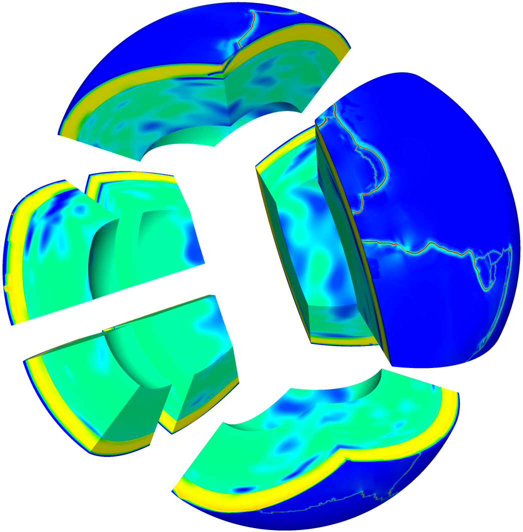


$(h,i,j+1)?$

$(h,i-1,j)?$ (h,i,j) $(h,i+1,j)?$

$(h,i,j-1)?$



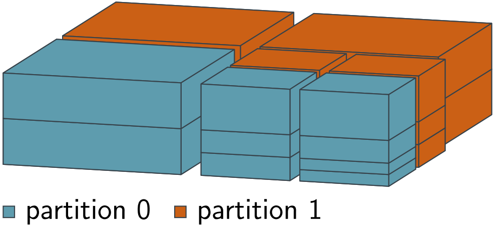


Mantle convection from (Stadler et al., 2010).

wanting an interface to solvers

Extension for thin domains (ice sheets)

(See my CSE15 slides, or arXiv:1406.6573)



GMG+AMG

(Sundar et al., SC12), (Rudi et al., best poster SC14, Rudi et al., paper submitted SC15): geometric-algebraic multigrid for mantle convection built on p4est.

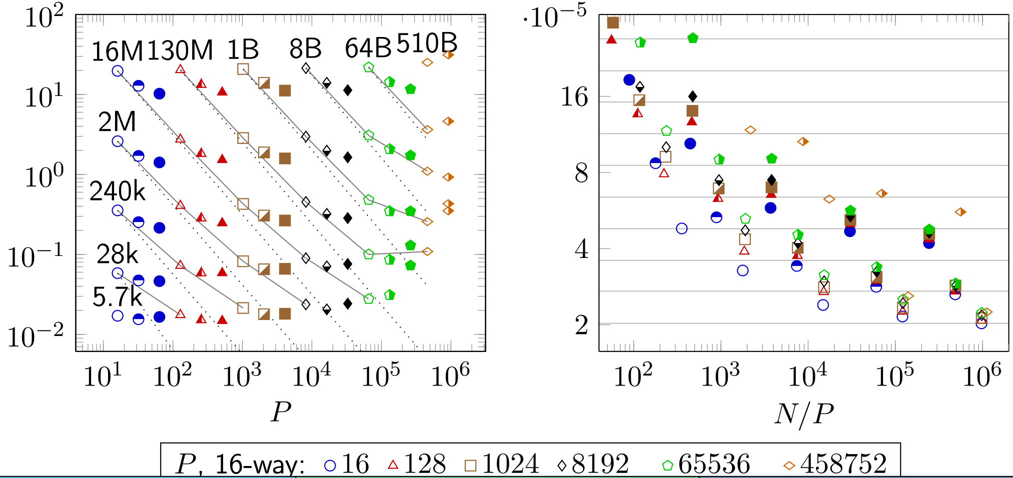
The implementation of GMG+AMG is very intrusive: it would've required more time than I had to adapt it to the thin-domain extension.

wanting extensibility, configurability

Scalability to 458K BG/Q cores of JUQUEEN from arXiv:1406.0089 (accepted SISC 2015).

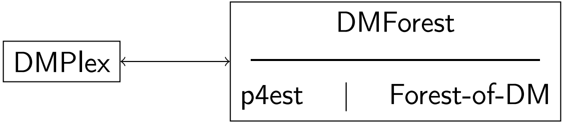
forest-to-mesh runtime in seconds

forest-to-mesh runtime in secs./(N/P)



DMForest

the plan



1. Support non-conforming cell interfaces in DMPlex. [done, in PETSc 3.6]
2. p4est-to-DMPlex conversion [done, in devel p4est, needs trivial update for PETSc 3.6]
3. DMForest interface, with p4est as first implementation. [stub]
   - Immediate solver support via backend conversion to DMPlex.
   - High-performance, native solver support if needed.
4. Runtime conversion of DMPlex to root forest (`-plex_convert_to_forest -forest_type X`):
   - user specifies coarse DMPlex that captures topology/geometry/X, has immediate access to all DMForest implementations.

Forest-of-DM

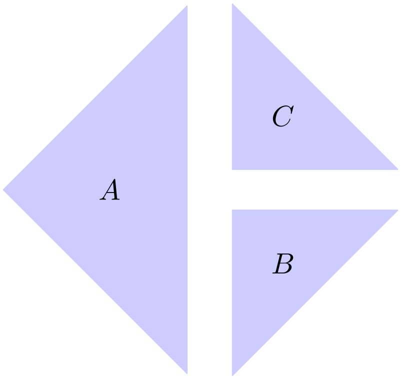
Design principle

Leaves on one level are roots on the next.

- All other aspects (coarse-mesh redundancy, partitioning, implicit vs. explicit tree) are implementation specifics.
- Other SFC approaches: Hilbert, Peano, Sierpinski
- Other external packages
- In particular, Forest-of-DA:
  - would allow a fair assessment of adaptivity overhead on domains other than boxes, i.e., compare locally-refined p4est to DA rather than to uniformly-refined p4est.

non-conforming cells in DMPlex

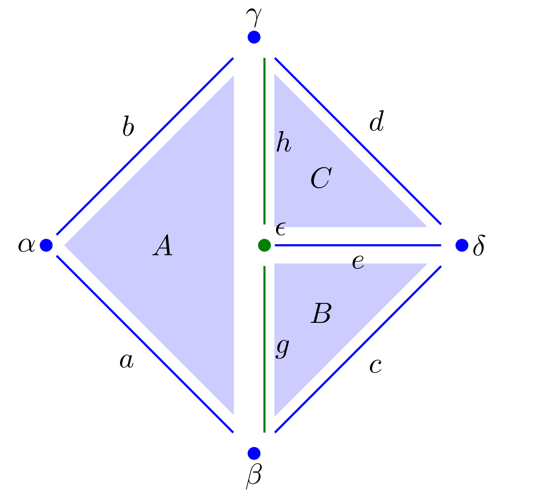
What is the proper completion of the DMPlex?



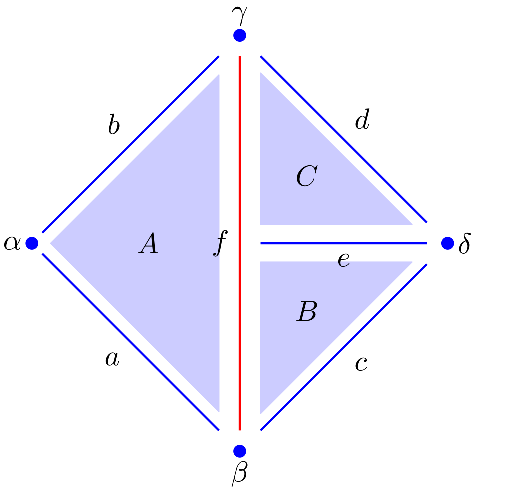
- We want contributors to write contributions meant for conforming meshes that still work for non-conforming meshes.

which is the proper partition completion?

FVM

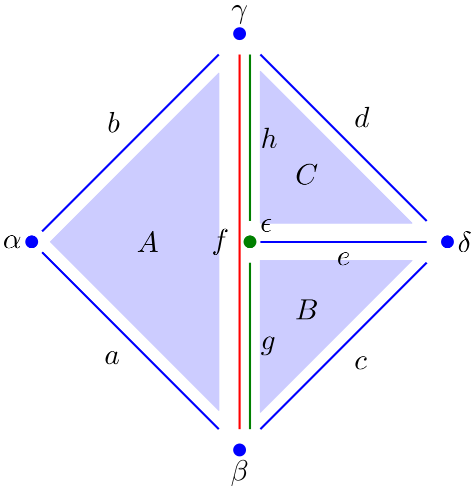


FEM



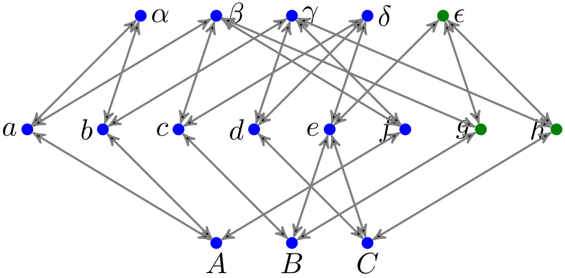
We add a tree structure to represent hierarchical points

DMPlex with hierarchical points



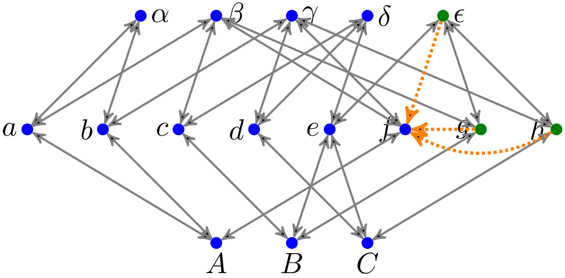
We add a tree structure to represent hierarchical points

DMPlex with hierarchical points



We add a tree structure to represent hierarchical points

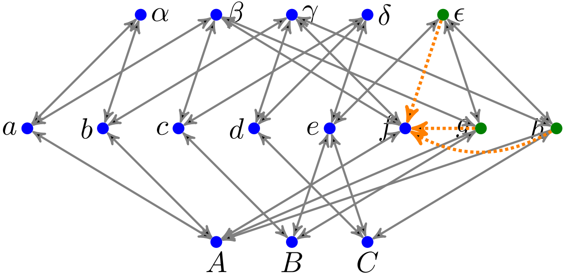
DMPlex with hierarchical points



DMPlexSetTree(...)

We add a tree structure to represent hierarchical points

DMPlex with hierarchical points



DMPlexSetTree(...)

- Supports automatically recalculated so adjacencies can be computed.
- Compatible with Michael Lange's mesh redistribution work.

We add a tree structure to represent hierarchical points

the two rules for FEM and FVM calculations

1. dofs on roots:
   - correct global sections automatically calculated.
2. quadrature on leaves:
   - not an issue for FEM if cells don't overlap.
   - FVM (and DG): only leaves in facet loops.

constraints

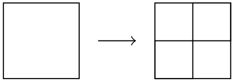
- Child points don't have global dofs (i.e., in global vectors), but they do have dofs in local vectors:
  - so that DMPlexVecGet/SetClosure() [get/set the closure of a function on a cell] always works.
- "Hanging-node" constraints must enforce $H^1/H^{\mathsf{curl}}/H^{\mathsf{div}}$ continuity, i.e., there is an interpolation operator I_a^c from *anchor* dofs to *constrained* dofs that must be applied.
  - Incorporated into DMGlobalToLocal()/DMLocalToGlobal(): constraints are always satisfied on local vectors/summed properly into global residuals.
  - DMPlexMatSetClosure() constructs the correct Jacobian.

automatic constraints

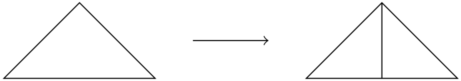
computing I_a^c

Most non-conforming meshes (such as p4est meshes) come from "reference" refinement rules:

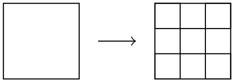
Uniform $1/2^d$ (Hilbert/Morton)



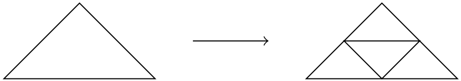
Bisection (Sierpinski)



Uniform $1/3^d$ (Peano)



Uniform $1/2^d$



- DMPlexSetReferenceTree(...):
  - (PetscFE) Hanging constraints computed on the reference refinement, copied into I_a^c (works even for curvilinear meshes).
  - Structured anisotropic refinement fits in this framework as well.

summary & current work

- DMForest will be a bridge from the successful, scalable p4est library to an extensible structured AMR interface:
  - Increase the availability of adaptivity to non-experts.
  - Develop (and fairly test) algorithms for $> 10^6$ processes.
- DMPlex can now serve as a general format for (hierarchically) non-conforming meshes, as it already has for conforming meshes.
  - Finite element support is complete (as complete as PetscFE).
  - Finite volume and discontinuous Galerking methods are in development, but the interface can be improved to hide complexity from the user.

Thank you!