The Impact of Structured AMR Representation on Software Design in PETSc

https://arxiv.org/abs/1508.02470

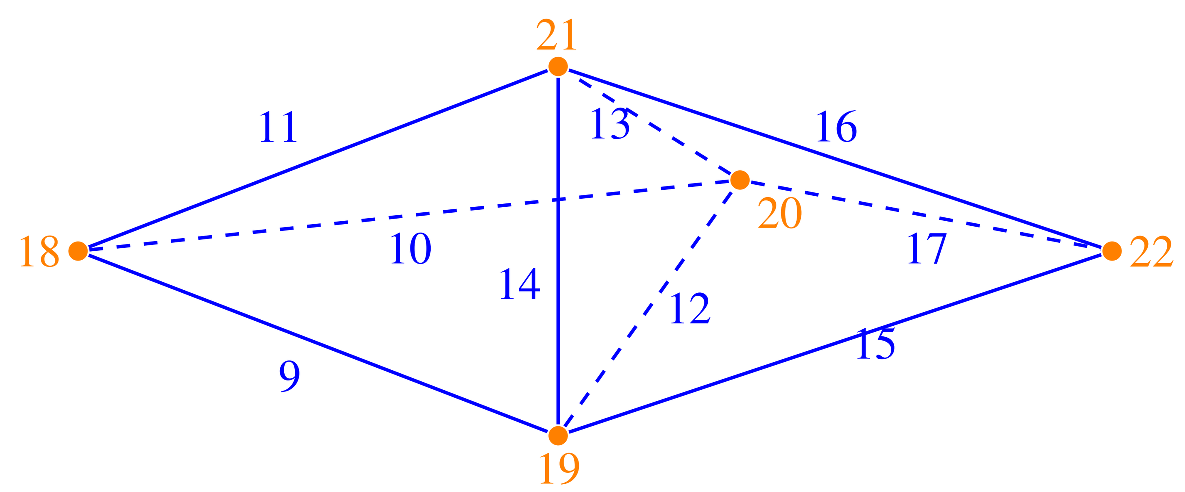
Tobin Isaac
Computation Institute, University of Chicago
tisaac@uchicago.edu

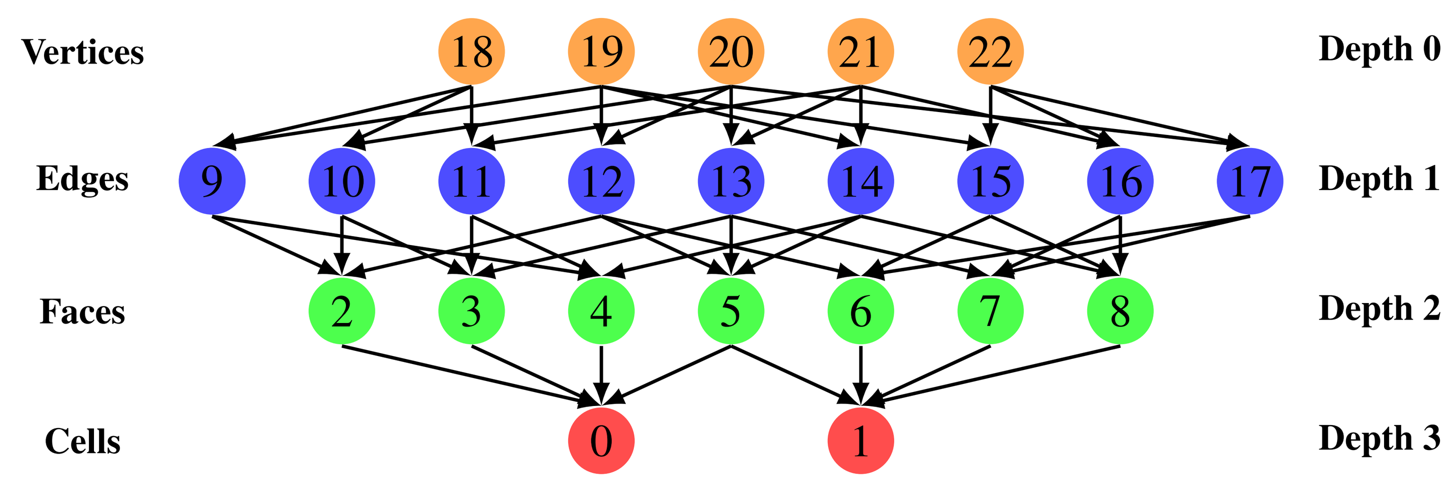
Matthew G. Knepley
Computational and Applied Mathematics, Rice University
knepley@gmail.com

We acknowledge support from the Department of Energy Office of Advanced Scientific Computing, the NSF SI2 Program, and the Rice Intel Parallel Computing Center.

Plex

Plex is an interface for manifold topology that encodes a CW-complex as a Hasse diagram (DAG). It allows true *dimension independent* programming, and arbitrary subsetting of meshes. Below are two tetrahedra sharing a face and the corresponding Hasse diagram.



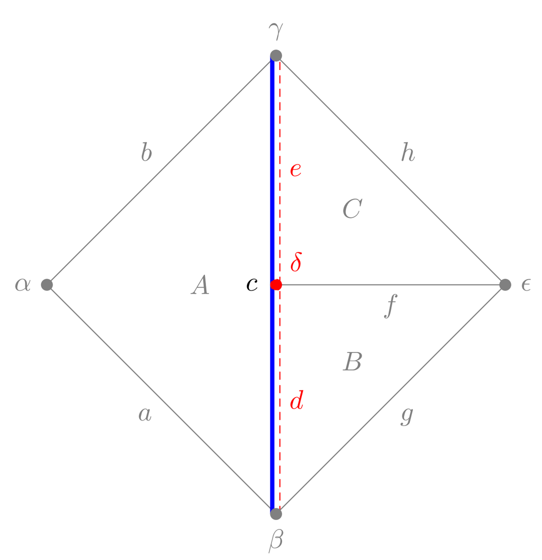


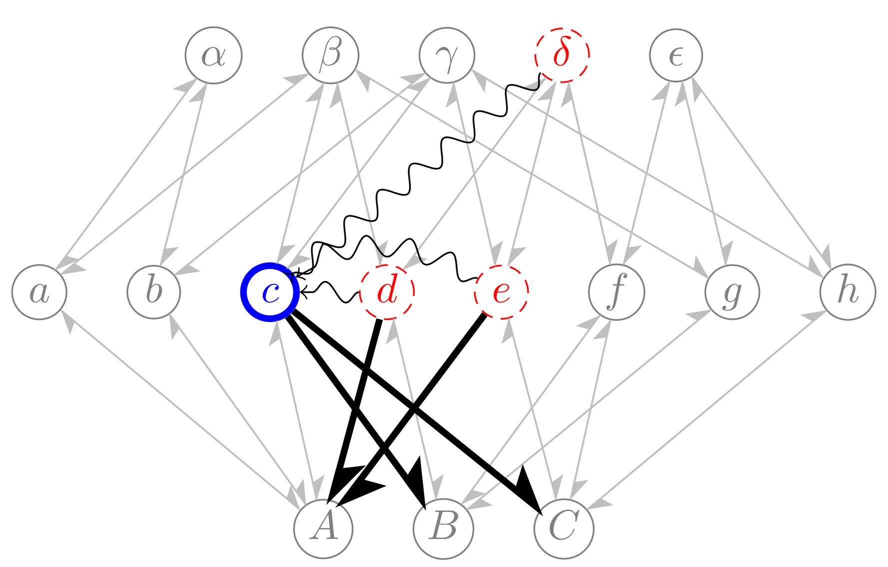
Basic Operations: $cone()$ for in-edges and $support()$ for out-edges, along with their transitive closures in the graph $closure()$ and $star()$.

Nonconforming Extension

We augment the interface with $parent()$ and $child()$ operations to identify subcells and subfaces, and we break the $cone()/support()$ duality such that

- $p \in supp(q) \quad \not\Rightarrow \quad q \in cone(p)$
- $star(p)$ covers the support of p's basis functions (also for FVM and DG)
- $cone(p)$ covers the boundary of p, but not child subfaces





The snaking lines show $parent(d)$, $parent(e)$, and $parent(\delta)$; the bold support arrows do not have matching cone arrows, breaking the duality that is present in conformal meshes.

Nonconforming FEM: H^1 Conformal Spaces on Nonconformal Meshes

- Finite element assembly using *Plex operations (right)* requires a *nodal basis* W for the discrete dual space V_h^*.
- Nodal bases of nonconformal meshes are *overdetermined*: W has linear dependencies.
- We constrain degrees of freedom of *child* points to *anchors()* $:= \overline{closure() \circ parent()}$.

→ enforced transparently during *parallel synchronization* (`DMGlobalToLocalBegin()` / `DMGlobalToLocalEnd()`), constraints automatically computed

- There are *three requirements* for H^1 conformal construction:

→ two on the Ciarlet triple for the reference element T,

$$T := (P(T) \text{ [approximation sp.]}, Q(T) \text{ [dual sp.]}, S(T) \text{ [reference cell CW-complex]}),$$

→ one on the cell embeddings $\{\varphi_i\}$ and their *pullbacks* and *pushforwards*,

$$\varphi_i^* f := f \circ \phi_i \qquad \text{[pullback function } f \text{ from } K_i \text{ to reference cell]}$$
$$\varphi_{i*}\sigma := \sigma \circ \varphi_i^* \qquad \text{[pushforward functional } \sigma \text{ from reference cell to } K_i].$$

I. $\forall \sigma_j \in Q, \exists p \in S \text{ s.t. } \sigma_j(\psi_k) = \delta_{jk} \Rightarrow \mathrm{supp}(\psi_k) = \bigcup star(p)$
("S decomposes the nodal basis of P")

Here $\psi_k \in P(T)$ is a basis function on the reference cell, and the requirement says that the *support()* relation also describes the support for functions "attached" to the points. The dual basis functions are naturally attached to mesh points, because the support of the measure defining the functional is contained within the *star()* of that point.
This assumption allow for the definition of compactly supported basis functions.

II. $\forall \mathcal{F} := \overline{T_i} \cap \overline{T_j} \neq \emptyset, \psi \in P(\varphi_j^{-1}\mathcal{F}) \Rightarrow \varphi_i^* \varphi_j^{-*}\psi \in P(\varphi_i^{-1}\mathcal{F})$
("Bases on neighboring cells line up")

Here $\varphi_j^{-*} := (\varphi_j^*)^{-1}$ and $P(X)$ is the trace space of $P(T)$ on $X \subset \overline{T}$. This makes sure that the embeddings of neighboring cells are compatible. This will allow us to build approximate solutions lying in the correct global function space. The notation is somewhat dense, so we can break this down in the following way for H^1,

$$\varphi_i^* \varphi_j^{-*}\psi \in P(\varphi_i^{-1}\mathcal{F}), \qquad \varphi_j^{-*}\psi \in P(\mathcal{F}), \qquad \psi \in P(\varphi_j^{-1}\mathcal{F})$$

III. $\forall p, q \in S \text{ s.t. } \varphi_i(p) = \varphi_j(q) \Rightarrow \exists M \in S_n \text{ s.t. } \mathcal{Q}_i^p = M\mathcal{Q}_j^q.$
("Neighboring dual bases line up, modulo permutations")

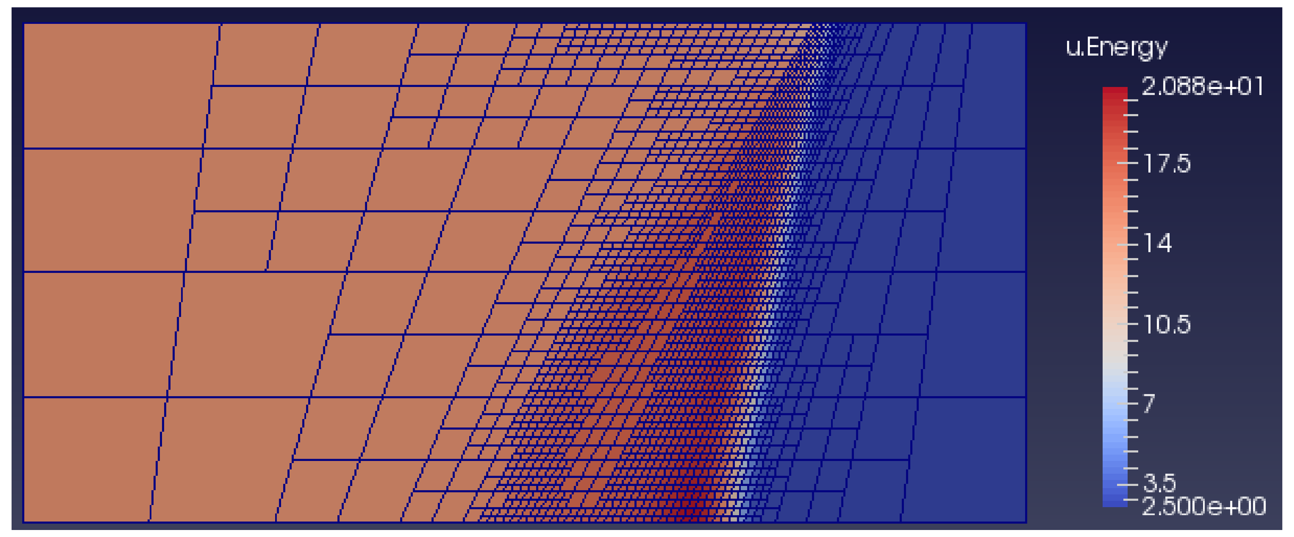
Here we must have $T_i \cap T_j \neq \emptyset$. This assumes that the dual bases of adjacent cells match up, in that the mappings of adjacent cells push functionals forward into each other. The permutations encode the symmetries of the polytopes in S and ensure that we choose the same member of the global dual basis. For example, the outward normal for a shared face is opposite for two cells sharing it, and only one direction must be chosen. Similarly, point evaluations on faces must follow the dihedral symmetries of the polytope.

Capabilities

With this concise, flexible interface, Plex supports

- Parallel mesh loading, partitioning, and redistribution for load balance
- FEM and FVM discretizations
- Unstructured data layout with boundary condition elimination
- Multigrid and block solvers

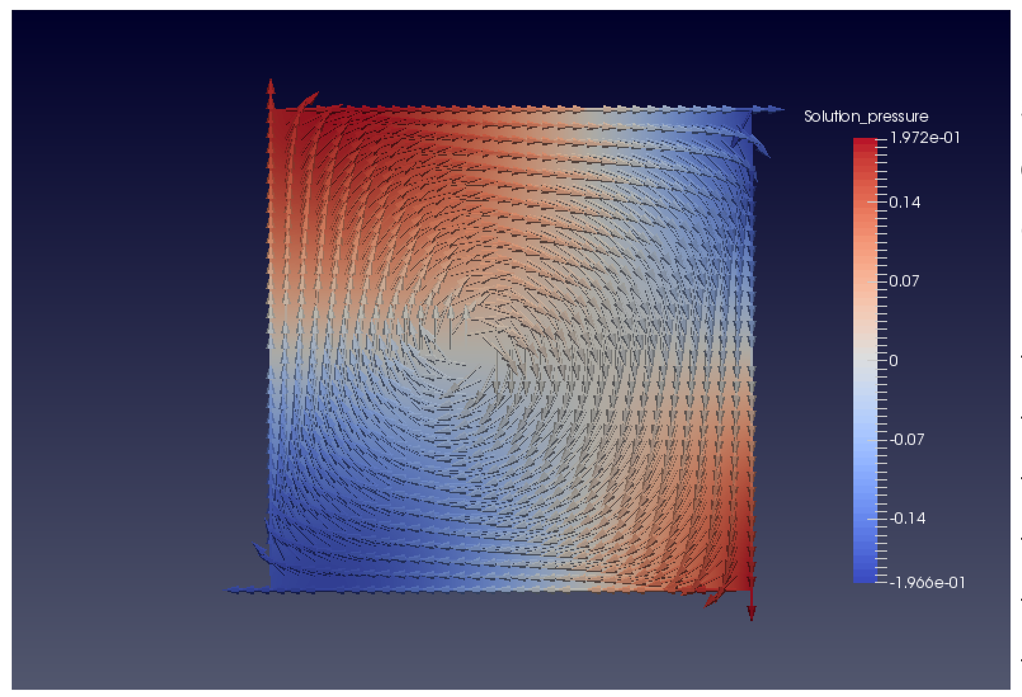
Finite Volume Example



The above figure shows a shock impinging on an oblique density contrast simulating using the Euler equation discretized with a TVD FV method, reproduced using TS ex11:

```
./ex11 -ufv_vtk_interval 1 -monitor density,energy -f -grid_size 2,1 -grid_bounds -1,1.,0.,1 -bc_wall 1,2,3,4
-dm_type p4est -dm_forest_partition_overlap 1 -dm_forest_maximum_refinement 6 -dm_forest_minimum_refinement 2 -dm_forest_initial_refinement 2
-ufv_use_amr -refine_vec_tagger_box 0.5,inf -coarsen_vec_tagger_box 0,1.e-2 -refine_tag_view -coarsen_tag_view
-physics euler -eu_type iv_shock -ufv_cfl 10 -eu_alpha 60. -grid_skew_60 -eu_gamma 1.4 -eu_amach 2.02 -eu_rho2 -3.
-petscfv_type leastsquares -petsclimiter_type minmod -petscfv_compute_gradients 0
-ts_final_time 1 -ts_ssp_type rks2 -ts_ssp_nstages 10
```

Finite Element Example



Instantaneous variable-viscosity Stokes simulation with exponential viscosity constrast e^{2Bx}, reproduced using SNES ex69:

```
./ex69 -dm_plex_separate_marker -dm_refine 4
-vel_petscspace_order 2 -pres_petscspace_order 1
-ksp_rtol 1e-12 -pc_type fieldsplit -pc_fieldsplit_type schur
-pc_fieldsplit_schur_factorization_type full
-pc_fieldsplit_schur_precondition a11 -fieldsplit_velocity_pc_type lu
-fieldsplit_pressure_ksp_rtol 1e-12 -fieldsplit_pressure_pc_type lu
-dm_view hdf5:sol.h5 -sol_vec_view hdf5:sol.h5::append
```