"Nonlinear Elliptic AMG(e) solvers"

by

Miguel Dumett, Panayot S. Vassilevski and Carol S. Woodward

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory

TOPS SciDAC Project Meeting
LLNL, Livermore
January 25-26, 2002

---

* Work performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Content

1. Methods
2. Software
3. Collaboration
4. Some examples

1. Methods

In general, our strategy is, whatever (A)MG(e) method is available adopt it to solve non–linear elliptic PDEs;

- (Modified) Inexact Newton MG: this requires solving linear problems with (A)MG(e);

  problem: resulting matrices are typically non–symmetric and possibly indefinite;

  $$(\mathcal{L}u,\ \varphi) \equiv \int_{\Omega} a(x,\ u)\nabla u \cdot \nabla \varphi \, dx$$

  Its Jacobian is a *convection–diffusion operator* ("nice" convection)

  $$\begin{aligned}(\mathcal{J}(u_0)v,\ \varphi) \ &= \int_{\Omega} [a(x,\ u_0)\nabla v \cdot \nabla \varphi \\ &\qquad + v\underline{b}(u_0) \cdot \nabla \varphi] \ dx\end{aligned}$$

  with

  $$\underline{b}(u_0) = a_u^{'}(x,\ u_0)\nabla u_0.$$

Method I: **True FAS**

For FAS one needs **coarse non-linear problems** constructed (algebraically, for unstructured finite element problems). This includes full MG and "cascadic" MG.

A coarse non–linear operator, for a coarse finite element space V_H (constructed by AMGe, for example) is defined by

$$(\mathcal{L}_H u,\ \varphi) \equiv \sum_{T \in \mathcal{T}_H} a(x_T,\ u_T) \int_T \nabla u \cdot \nabla \varphi \, dx,\ u, \varphi \in V_H,$$

where x_T and u_T are averaged values over every element T.

This strategy can be carried out for unstructured finite element meshes and if access to node coordinates is available, the method of creating coarse non–linear operators is extendable to more general non–linearities, like

$$a(x,\ u,\ \nabla u).$$

- using standard global coarse spaces;

  problem for FAS (and any MG): if direct coarse non–linear problems are built, typically the resulting forms are "non–inherited".

- using special "coarsened away" global spaces (reverse to Mitchell, Bank and Holst)

  problem: parallelizing for $>> 1$ processors; can be overcome by localizing the coarse spaces;

- coarse non–linear problems are solved by Newton-type method; This leads to a non–linear Schwarz / FAS method. Fine–grid smoothing is needed in general.

  problem: choosing appropriate initial iterates if local coarse non–linear problems are used;

- one can solve the non–linearly preconditioned problem in ASPIN framework;

Method II: **Preconditioned non–linear GCG method**

as "parameter–free" Inexact Newton method.

Outline:

$$F(u) = 0.$$

$J(u_0)$ – Jacobian. Let $\{d_k\}$ be a set of search vectors. Given a current iterate u_0 one first solves a non–linear problem for $\{\alpha_k\}$ of small size

$$(F(u_0 + \sum_k \alpha_k d_k),\ J(u_0)d_j) = 0.$$

One can use Newton method here.

The new iterate is

$$u_0 := u_0 + \sum_k \alpha_k d_k.$$

The new search direction d is defined by

$$d = \widetilde{r} - \sum_k \beta_k d_k \ : \ (d,\ d_j) = 0, \text{ all } j,$$

where $\widetilde{r}$ is a preconditioned version of the residual $r = F(u_0)$. If $\widetilde{r} \approx J(u_0)^{-1} r$, i.e., an inexact Newton direction, the thus preconditioned nonlinear GCG converges at least as fast as a corresponding stationary inexact Newton method.

For the preconditioner a natural choice is some approximation to $J(u_0)^{-1} r$ using linear (A)MG(e), or any of the previous Schwarz methods exploiting various coarse spaces.

Software

We are pretty much in a "research phase" exploring algorithms utilizing all the potential of linear solvers available from *HYPRE*.

Our **goal**, for this project, is to have a general purpose unstructured finite element non–linear elliptic PDE solver, exploiting ϱ(A)MG(e) algorithms.

Any parallel linear solver available in *HYPRE* used as a preconditioner, to solve the Jacobian systems. If type of problem is appropriate, one can use BoomerAMG (in parallel).

Generally, our intent is, for general systems of elliptic PDEs, that a parallel (ϱ)AMGe will be used for parallel tests (hopefully soon).

Current status:

Preliminary work has been done using sequential codes for solving unstructured non–linear elliptic finite element problems, implementing:

- FAS with standard agglomeration based AMGe coarsening;

- FAS-Schwarz, based on global coarsened away spaces;

- preconditioned non–linear GCG with inexact Jacobians used as a preconditioner; the Jacobians were preconditioned with additive Schwarz -type (linear) utilizing global coarsened away spaces, and non–inherited forms;

- non–linear preconditioning in ASPIN framework utilizing local coarsened away spaces. The latter are in process of testing in collaboration with Xiao-Chuan Cai and Leszek Marcinkowski.

The problems we have seen were mentioned above.

Collaboration

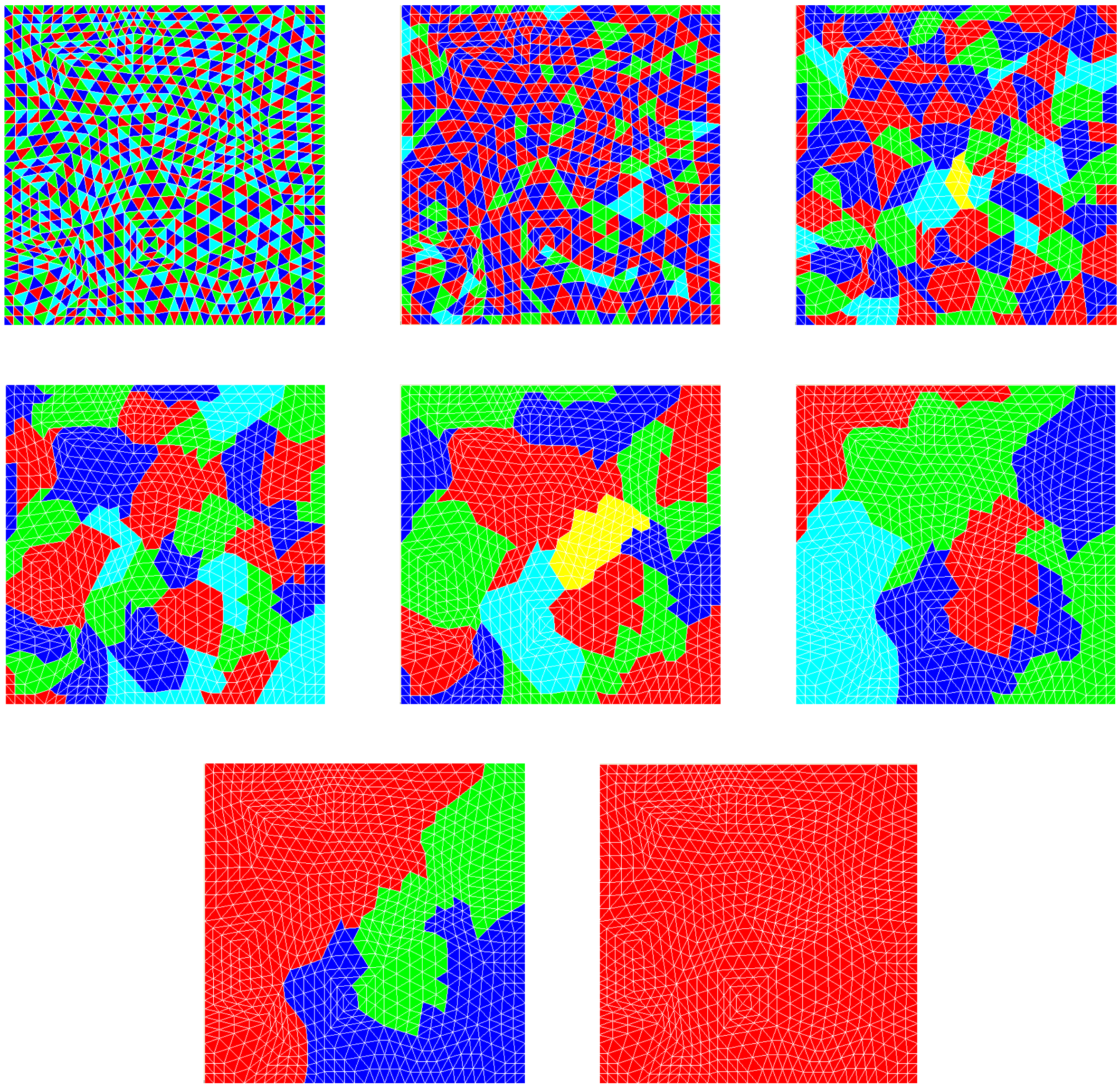
Our main collaborators are:

- scalable linear solvers team, CASC–LLNL: linear solvers part;

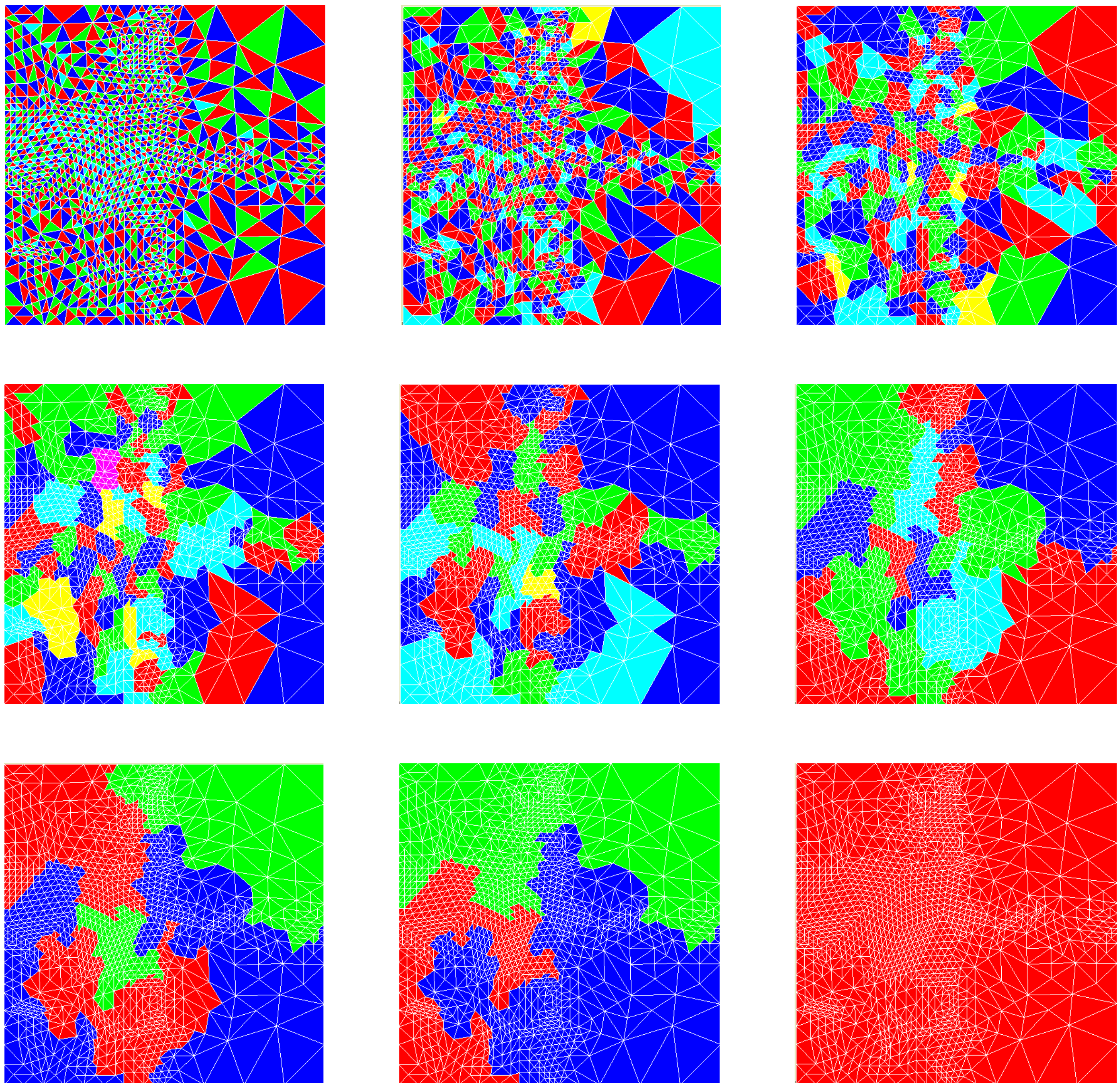
- UC Boulder: Steve McCormick, Tom Manteuffel, John Ruge on the MG part;

- UC Boulder: Xiao-Chuan Cai and Leszek Marcinkowski on the Schwarz part.

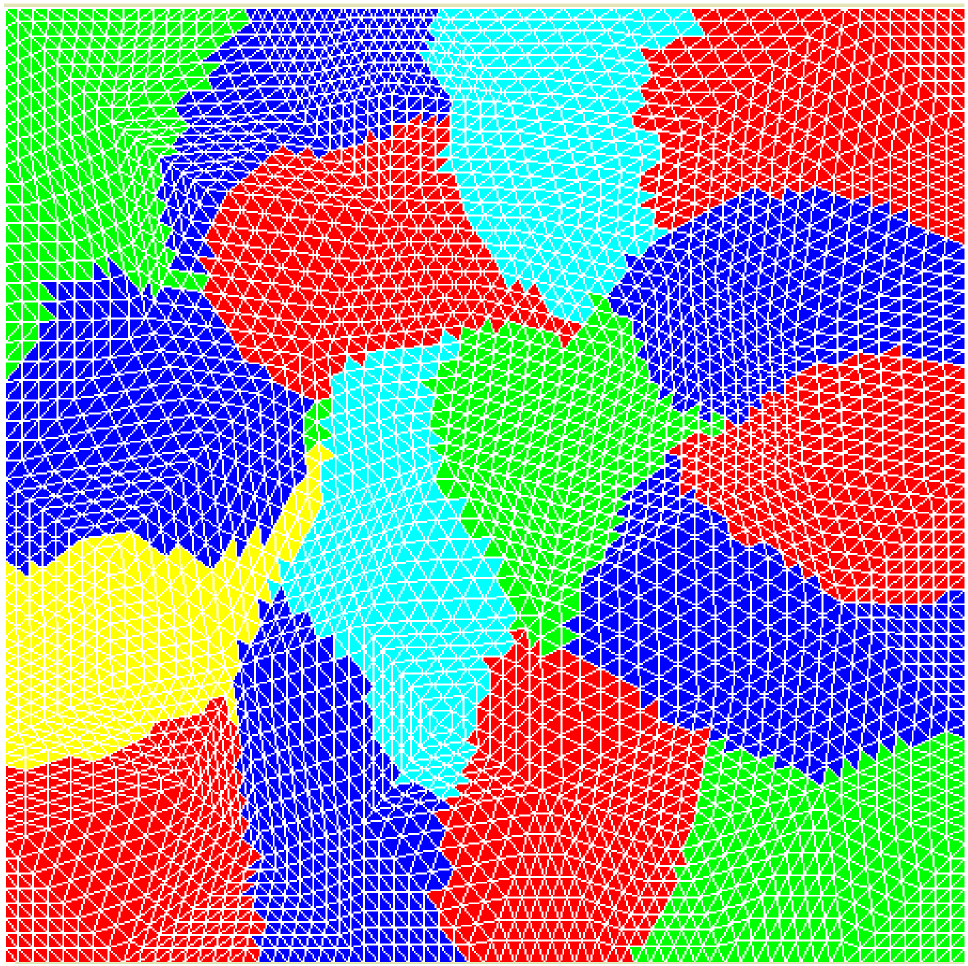
4. Examples of coarse elements



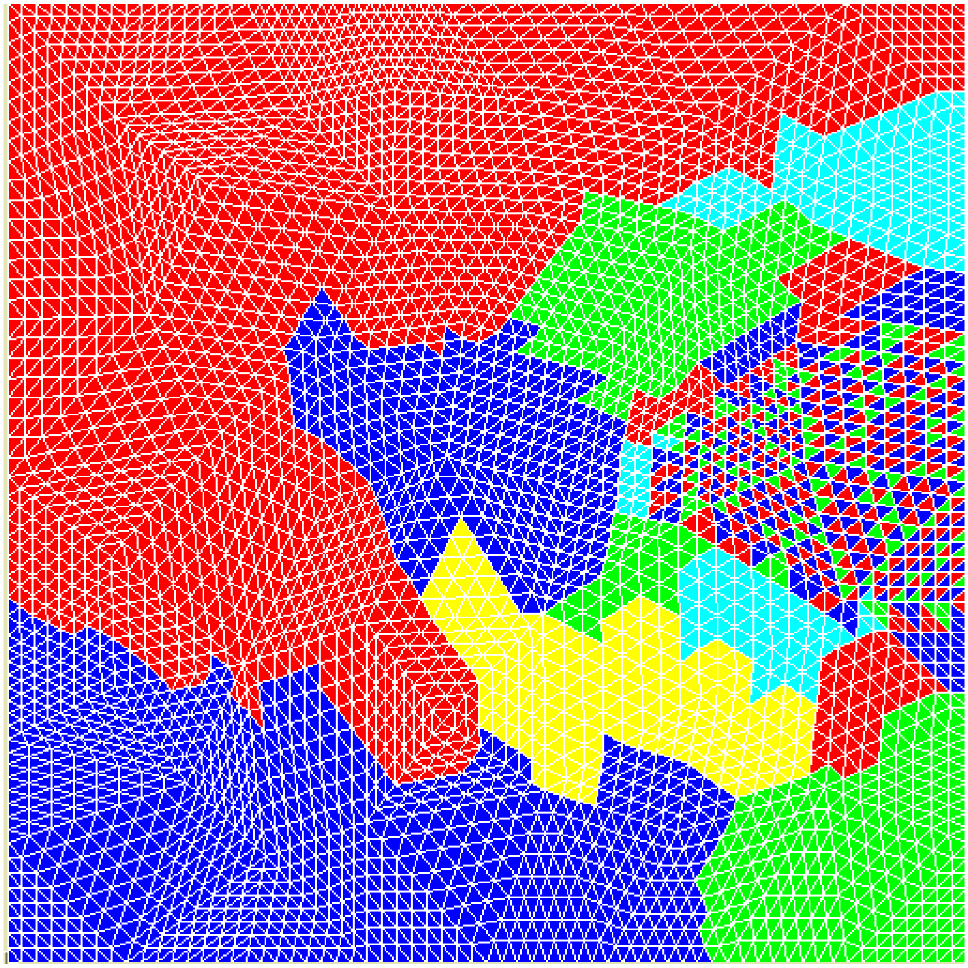
Sequence of increasingly coarse elements, formed by element agglomeration.



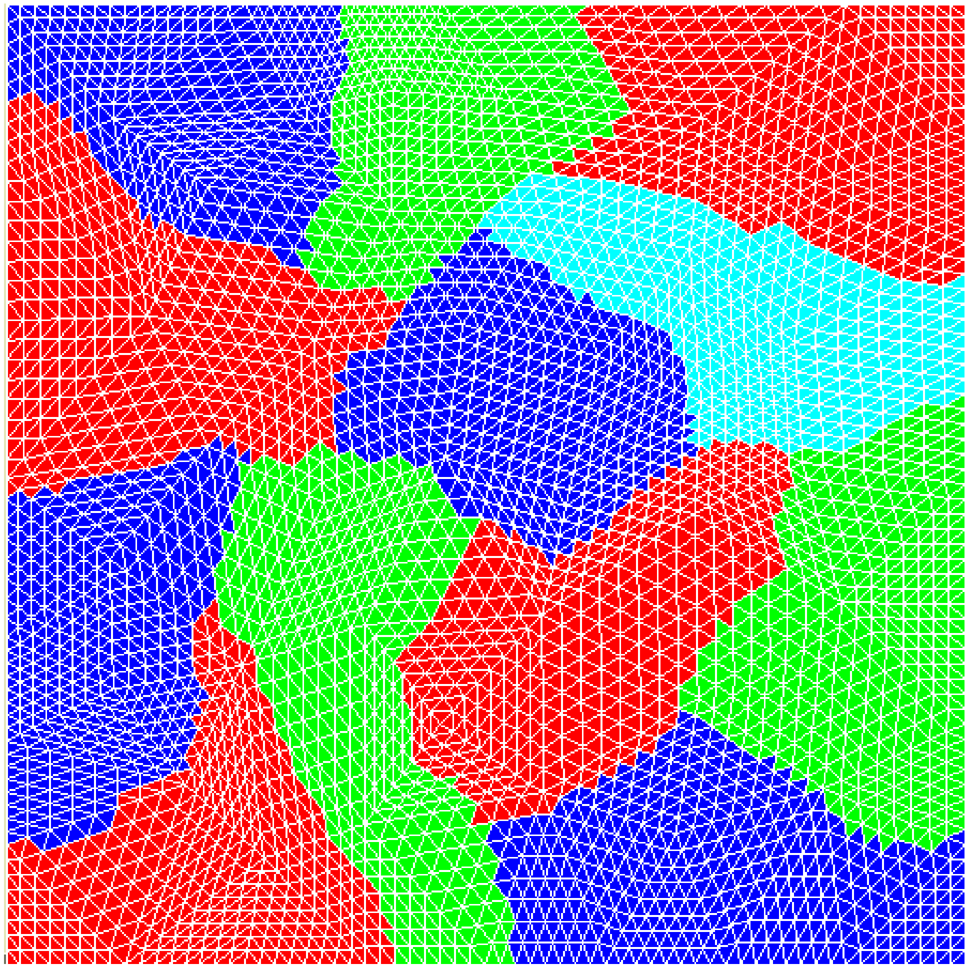
Sequence of increasingly coarse elements formed by element agglomeration; unstructured fine mesh with local refinement



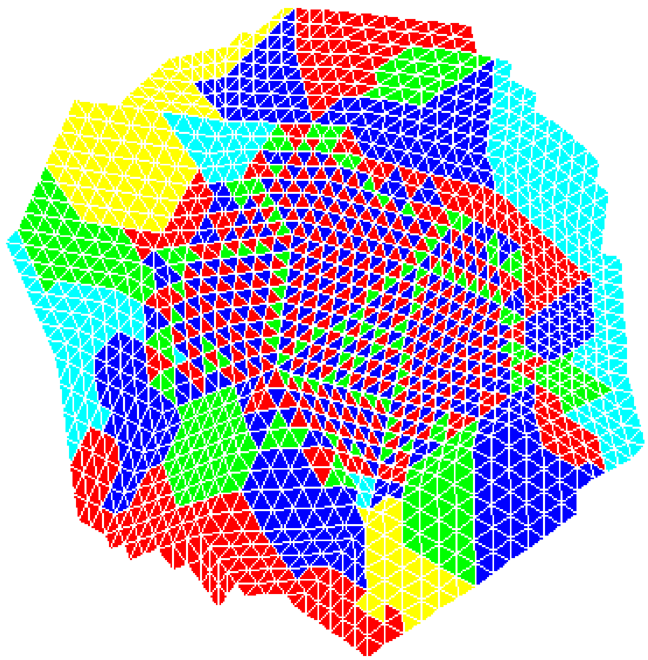
Partitioned mesh of 6,400 fine elements into 16 subdomains: each color represents subdomain of fine–grid elements.



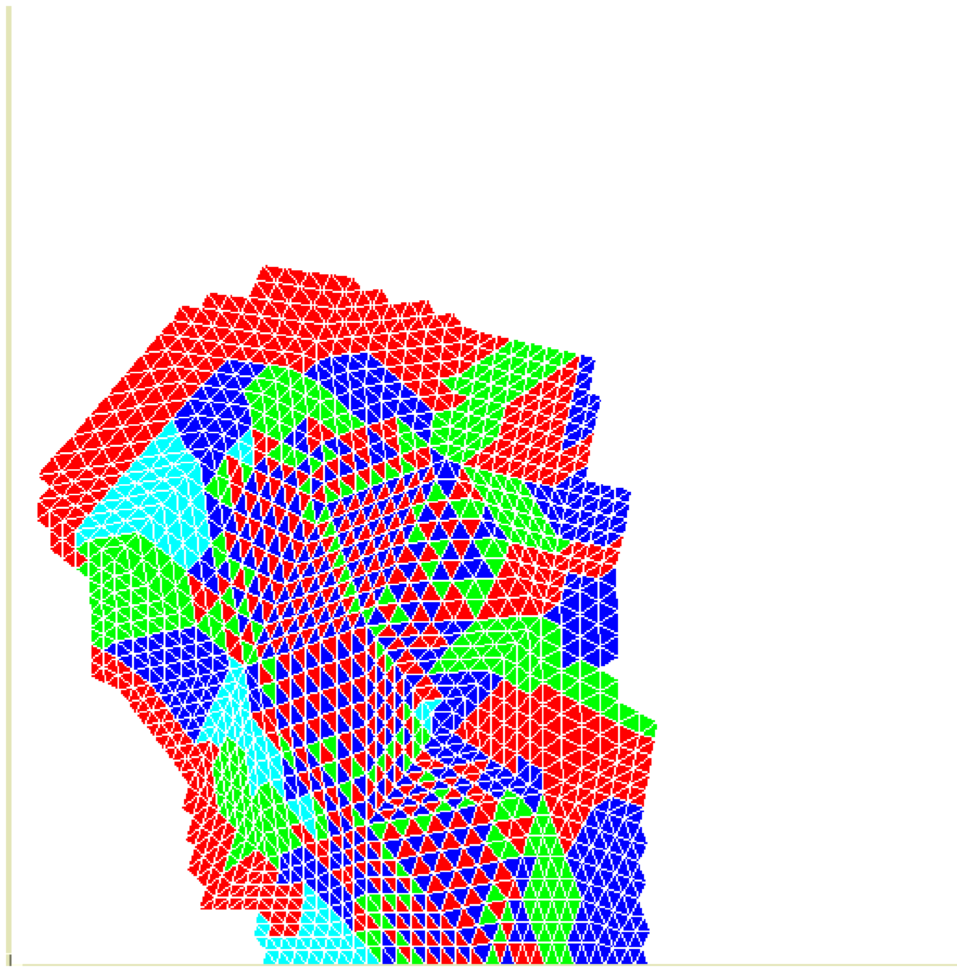
An agglomeration based coarsened away mesh: 6,400 fine elements, 456 agglomerated elements, 400 subdomain elements. Each color represents an agglomerate.



Partitioned mesh of 6,400 fine elements into 12 subdomains: each color represents subdomain of fine–grid elements.



Local coarsened away mesh.



Local coarsened away mesh.