A Parallel Variational Mesh Quality Improvement Method for Distributed Memory Machines

Suzanne M. Shontz[1], Maurin A. Lopez Varilla[2], Weizhang Huang[3]

[1]Department of Electrical Engineering and Computer Science
University of Kansas

[2]Husky Injection Molding, Luxembourg

[3]Department of Mathematics, University of Kansas

Workshop on Adaptive Numerical Methods
for PDEs with Applications
Banff, Canada
May 28 - June 1, 2018

Motivation

There are numerous large-scale applications requiring **mesh adaptivity**, e.g., computational fluid dynamics and weather prediction.

Parallel processing (e.g., CPUs, GPUs, Phi co-processors, ...) is needed in order to perform simulations involving parallel adaptive meshes.

In this talk, we propose a **parallel variational mesh quality improvement method**.

Our method is a parallel implementation of a serial variational mesh quality improvement method by Huang and Kamenski.

Outline

Overview of Variational Mesh Adaptation

In the variational approach, an **adaptive mesh** is **generated as the image of a reference mesh under a coordinate transformation**.

The coordinate transformation is determined as the **minimizer of a meshing functional**.

The **mesh concentration** is typically **controlled through a scalar or a matrix-valued function**. This is referred to as the **metric tensor** or **monitor function**.

Monitor functions are defined based on **error estimates** and/or **physical considerations**.

Variational Mesh Adaptation Methods

Some example **variational methods** and **meshing functionals**:

- **Equipotential method** based on **variable diffusion** (Winslow)
- Method combining **mesh concentration, smoothness, and orthogonality** (Brackbill and Saltzman)
- Method based on the **energy of harmonic mappings** (Dvinsky)
- Methods based on **conditioning the Jacobian matrix of the coordinate transformation** (Knupp, Knupp and Robidoux)
- Methods based on the **equidistribution and alignment conditions** (Huang, Huang and Russell)

Serial Variational Mesh Adaptation Method

Moving Mesh PDE (MMPDE) Method: (Huang, Ren, Russell, 1994)

New implementation: (Huang and Kamenski, 2015)

- Consider a domain $\Omega \subset \mathbb{R}^d$ $(d \geq 1)$ and a simplicial mesh $\mathcal{T}_h = \{K\}$ of N elements and N_v vertices thereon.
- Denote the affine mapping $F_K : \hat{K} \to K$ and its Jacobian matrix by F_K', where $\hat{K}$ is the master element.
- Let the edge matrices for K and $\hat{K}$ be E_K and $\hat{E}$.
- Assume that a metric tensor (or a monitor function) $\mathbb{M} = \mathbb{M}(\mathbf{x})$ is given on Ω which provides directional and magnitude information for elements.

Moving Mesh PDE (MMPDE) Method

A key idea of the MMPDE method is to view an adaptive mesh as a uniform one in the metric $\mathbb{M}$:

- The size of all elements K in the metric $\mathbb{M}_K$ is the same
- All elements K in the metric $\mathbb{M}_K$ are similar to $\hat{K}$

These give rise to the equidistribution and alignment conditions:

$$|K|\sqrt{\det(\mathbb{M}_K)} = \frac{\sigma_h}{N}, \quad \forall K \in \mathcal{T}_h$$

$$\frac{1}{d}\operatorname{tr}\left((F_K')^T \mathbb{M}_K F_K'\right) = \det\left((F_K')^T \mathbb{M}_K F_K'\right)^{\frac{1}{d}}, \quad \forall K \in \mathcal{T}_h$$

where $|K|$ is the volume of K and $\sigma_h = \sum_K |K|\sqrt{\det(\mathbb{M}_K)}$.

Moving Mesh PDE (MMPDE) Method

An energy function for the equidistribution and alignment conditions is

$$I[\mathcal{T}_h] = \sum_K \sqrt{\det(\mathbb{M}_K)}\; G_K$$

$$G_K = \frac{1}{3}\left(\mathrm{tr}(\mathbb{J}\mathbb{M}_K^{-1}\mathbb{J}^T)\right)^d + \frac{1}{3}d^d\left(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M}_K)}}\right)^2$$

$$\mathbb{J} = (F_K')^{-1} = \hat{E}E_K^{-1}$$

The MMPDE moving mesh equation is defined as the (modified) gradient system of $I[\mathcal{T}_h]$, i.e.,

$$\frac{d\mathbf{x}_i}{dt} = -\frac{\det(\mathbb{M}_i)^{\frac{1}{d}}}{\tau}\frac{\partial I[\mathcal{T}_h]}{\partial \mathbf{x}_i}, \quad i = 1, \ldots, N_v$$

where $\tau > 0$ is a parameter for adjusting the response time scale of mesh movement to the change in $\mathbb{M}$.

Moving Mesh PDE (MMPDE) Method

For quality improvement, we choose $\mathbb{M} = \mathbb{I}$, which means we want **the mesh to be as uniform as possible in the Euclidean space**. In this case, the moving mesh equation reads as

$$\frac{d\mathbf{x}_i}{dt} = \sum_{K \in \omega_i} |K| \mathbf{v}_{i_K}^K, \quad i = 1, \ldots, N_v \tag{1}$$

where ω_i is the element patch associated with $\mathbf{x}_i$, i_K is the local index for $\mathbf{x}_i$ on K, and $\mathbf{v}_{i_K}$ is the nodal velocity associated with node $\mathbf{x}_i$,

$$\begin{bmatrix} (\mathbf{v}_1^K)^T \\ \vdots \\ (\mathbf{v}_d^K)^T \end{bmatrix} = G_K E_K^{-1} + E_K^{-1} \frac{\partial G_K}{\partial \mathbb{J}} \hat{E} E_K^{-1} + \frac{\partial G_K}{\partial \det(\mathbb{J})} \frac{\det(\hat{E})}{\det(E_K)} E_K^{-1}$$

$$(\mathbf{v}_0^K)^T = -\sum_{j=1}^{d} (\mathbf{v}_j^K)^T$$

Moving Mesh PDE (MMPDE Method)

The nodal velocities of the **boundary nodes** are set to 0. (They can also be modified so that the boundary nodes slide on the boundary.)

Solve (1) using the **adaptive fourth-order Runge-Kutta-Fehlberg ODE solver (RKF45)**.

It is shown analytically and numerically in (Huang and Kamenski, 2018) that the mesh governed by the MMPDE moving mesh equation will stay **nonsingular** (no crossing over nor tangling) if it is nonsingular initially.

Parallel Variational Mesh Quality Improvement Method

Strategy: Develop a parallel algorithm for use on distributed-memory HPC machines with p processors.

1. **Partition** the mesh into p **connected components** using the multilevel k-way partitioning scheme in METIS (minimizing the number of edge cuts).
2. **Solve the discretized system of ODEs in a parallel, distributed manner** using the RKF45 method.
   - 2.1 **Key ingredient: Computation of the nodal velocities**.
   - 2.2 Each processor loops over the elements it owns and computes the nodal velocities for the interior nodes.
   - 2.3 **Asynchronous communication** occurs to compute the nodal velocities for the shared nodes.
   - 2.4 The **step size**, dt, is **adapted** based on the **global error.**

Parallel Variational Mesh Quality Improvement Method

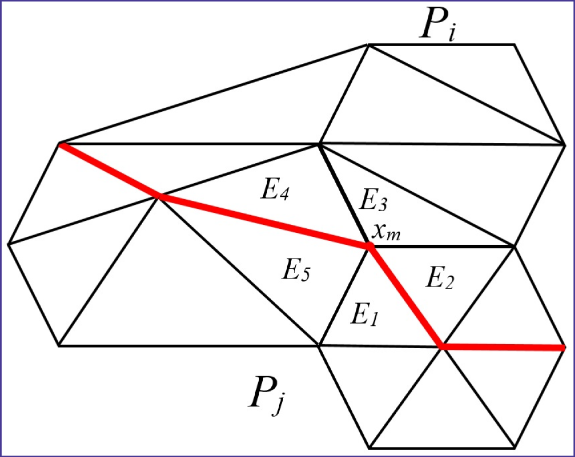


Figure: Patch of elements with x_m as one of its vertices.

Parallel Variational Mesh Quality Improvement Method

3. The computation **terminates** when the mesh **quality is acceptable.**

We **overlap the computation and the communication** for efficient performance.

We implement the parallel algorithm using **MPI and C/C++**.

Performance Gain: Parallel Algorithm Analysis

Total parallel time per iteration is bounded by:

$$\begin{aligned} T_P = & (d+1) * \lceil N/p \rceil_{sh} t_{vn} + Tc_{total} \\ & + N_{sh} * t_c + \lceil N/p \rceil * (d+1) * d * t_e + \log_2(p) \\ & + N_{sh} * t_c + \lceil N/p \rceil * t_q + \log_2(p), \end{aligned}$$

where the four terms denote the time to compute nodal velocities, overlapped computation and communication time for solving the ODE, time to compute the max error, and time to compute the average mesh quality.

Rearranging:

$$\begin{aligned} T_P = 2(N_{sh} t_c + log_2(p)) + (d+1)(\lceil N/p \rceil_{sh} t_{vn} + \lceil N/p \rceil d * t_e) \\ + \lceil N/p \rceil t_q + Tc_{total}. \end{aligned}$$

Performance Gain: Parallel Algorithm Analysis

Observation: The **communication time increases by a factor of** $\log_2(\mathbf{p})$ if the number of processors increases with the number of elements (and the number of shared nodes).

Conclusion: Good strong scaling results are expected for our implementation.

Numerical Experiments: Geometric Domains

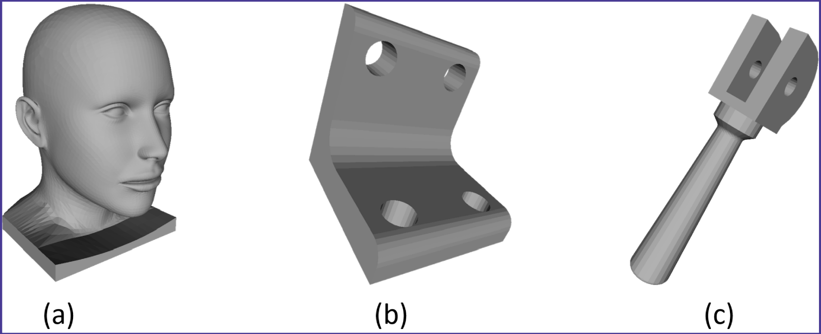


Figure: Geometric domains: (a) bust, (b) bracket and (c) double cam tool

Numerical Experiments: Tetrahedral Mesh Sizes

Table: Size of the tetrahedral meshes

Mesh	# Nodes	# Elements
Bust	12,895,493	80,000,012
Double cam tool	7,089,753	41,405,684

Numerical Experiments: Mesh Quality Results

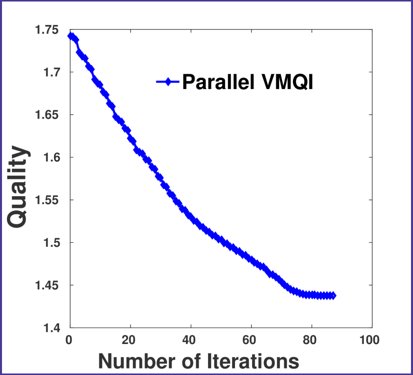


Figure: Average quality versus number of iterations for the 80M element mesh on the bust domain

Numerical Experiments: Performance Results

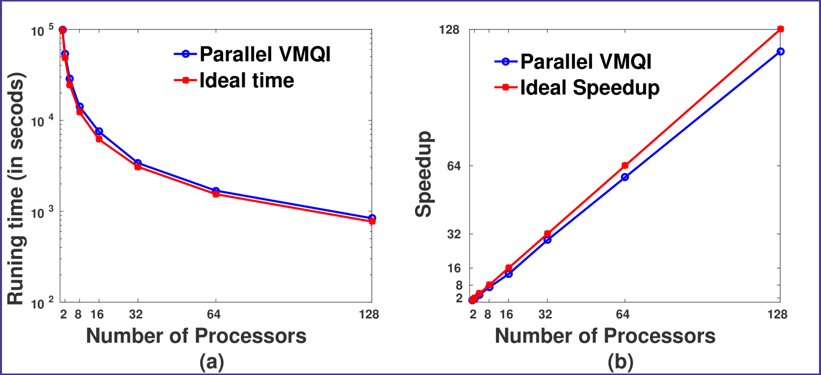


Figure: (a) Total runtime and (b) speedup for the Parallel VMQI algorithm for the 80M element mesh on the bust domain

Numerical Experiments: Mesh Quality Results

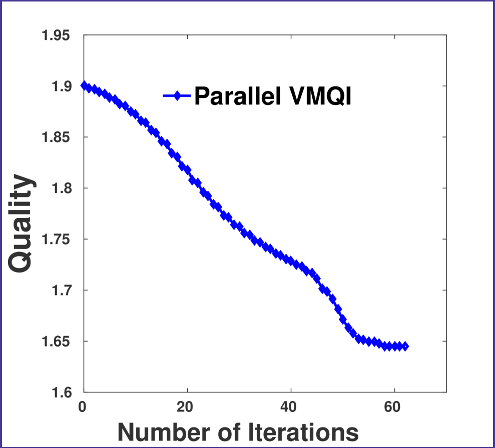


Figure: Average quality versus number of iterations for the 40M element mesh on the double cam tool domain

Numerical Experiments: Performance Results

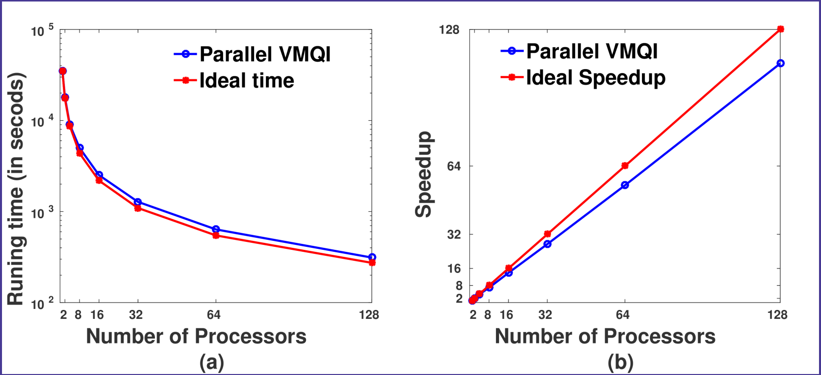


Figure: (a) Total runtime and (b) speedup for the Parallel VMQI algorithm for the 40M elements tetrahedral mesh of the double cam tool domain

Numerical Experiments: Overlapping Communication with Computation

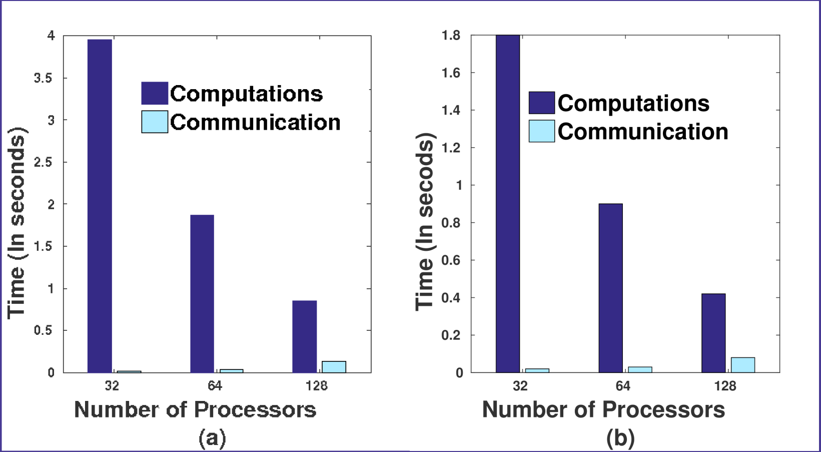


Figure: Communication and computation time to calculate the nodal velocities in one region for one iteration: (a) 80M elements (b) 40M elements

Numerical Experiments: Mesh Sizes for Weak Scaling

Table: Different mesh sizes for the bracket domain

Mesh	# Nodes	# Elements
bracket	450,960	2,500,032
	864,028	5,000,025
	1,716,222	9,999,990
	3,269,784	19,999,978
	6,497,224	40,000,000
	12,957,609	80,000,037
	24,177,335	159,745,245

Numerical Experiments: Performance Results for Weak Scaling

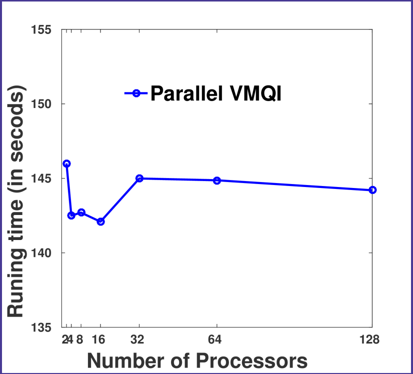
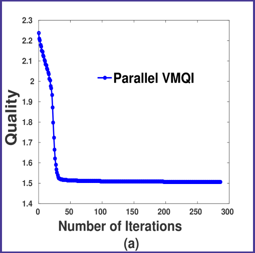
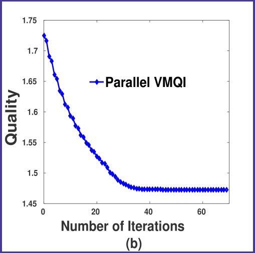


Figure: Runtime vs. number of processors

Numerical Experiments: Mesh Quality Results for Weak Scaling





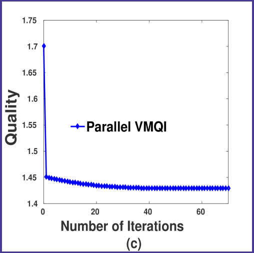


Figure: Quality vs. number of iterations for the bracket domain with (a) 2.5M, (b) 10M, and (c) 40M elements

Conclusions and Future Work

We have proposed a **parallel and distributed formulation** of the **mesh quality improvement algorithm** by Huang and Kamenski.

The **RKF45 method** is used to **solve the system of ODEs** associated with the **MMPDE method** to obtain the **nodal velocities** on the distributed mesh; this is done in parallel.

The **communication and computation are overlapped** in order to obtain an efficient method.

Excellent strong scaling results and typical weak scaling results were obtained.

For future work, various **partitioning and communication srategies** will be employed.

Acknowledgments

- NSF CAREER Award ACI-1500487 (formerly ACI-1330054 and OCI-1054459)
- NSF PECASE Award
- Army Research Office Grant W911-NF-1510377 (cyberinfrastructure)

27th International Meshing Roundtable

- The 27th International Meshing Roundtable will be held in Albuquerque, NM on October 1-5, 2018.
- Participants can give two kinds of talks: Talks on technical papers and research notes.
- To give a talk on a technical paper, participants must be a co-author of an accepted paper.
- **NEW:** For research notes, authors submit an (optional) 5-page paper. Alternatively, an abstract may be submitted.
- **Paper submission deadline: May 30, 2018**
- **Research note deadline: August 8, 2018**

KU Engineering Dean Search

- KU is searching for a dynamic leader to be Dean of the School of Engineering.
- The School of Engineering has seen tremendous expansion and growth in the numbers of students and faculty and the facilities in the past five years.
- It's an exciting time to work for the School of Engineering.
- Handouts are available with more information.
- Please consider applying for the position!