Spacetime Meshing for Discontinuous Galerkin Methods

Shripad Thite

Ph.D. defense — August 22, 2005
Collaborators

Members of the Center for Process Simulation & Design (CPSD)

**CS** Jonathan Booth, Sayantan Chakraborty, Shuo-Heng Chung, Prof. Jeff Erickson, Prof. Michael Garland, Harigovind Gopinathapannicker, Damrong Guoy (also CSAR), Mark Hills, Prof. Sanjay Kale, **Shripad Thite**, Terry Wilmarth, Yuan Zhou

**Math** Prof. John Sullivan (also TU-Berlin)

**TAM** Reza Abedi, Yong Fan, Prof. Robert Haber, Morgan Hawker, Jayandran Palaniappan

Supported by NSF ITR grant DMR-0121695
Spacetime Meshing with Adaptive Refinement and Coarsening
Reza Abedi, Shuo-Heng Chung, Jeff Erickson, Yong Fan, Michael Garland, Damrong Guoy, Robert Haber, John M. Sullivan, Shripad Thite, Yuan Zhou
20th Symp. Computational Geometry, June 2004

An $h$-adaptive Spacetime-Discontinuous Galerkin Method for Linearized Elastodynamics
Reza Abedi, Robert Haber, Shripad Thite, Jeff Erickson
Submitted to Revue Européenne des Éléments Finis, 2005

Efficient Spacetime Meshing with Nonlocal Cone Constraints
Shripad Thite
13th International Meshing Roundtable, September 2004

A Unified Algorithm for Adaptive Spacetime Meshing with Nonlocal Cone Constraints
Shripad Thite
21st European Workshop on Computational Geometry, March 2005
Invited to special issue of Computational Geometry: Theory and Applications (CGTA)
Important applications in science and engineering involve simulating transient time-dependent phenomena

e.g., Conservation Laws in Elastodynamics and Fluid Dynamics. Ocean waves, Acoustics, Gas dynamics, Traffic flow

Computer simulation of such phenomena involves solving space-time hyperbolic PDEs

e.g., Wave traveling along a taut string with wavespeed $\omega$:

$$u_{tt} - \omega^2 u_{xx} = 0$$
When a rectangular plate with a crack in the center is loaded, the shock wave scatters off the crack-tip.

Meshing: Shuo-Heng Chung, Shripad Thite; Solution: Reza Abedi;
Visualization: Yuan Zhou
A spacetime mesh is a partition of the spacetime domain into simplices (triangles, tetrahedra, etc.)

Given a spacetime mesh $\Omega$, Spacetime Discontinuous Galerkin (SDG) methods compute the approximate numerical solution over $\Omega$

I give algorithms to generate spacetime meshes in $dD \times \text{time}$

The algorithms in my thesis support an efficient, parallelizable, $O(N)$-time solution strategy by SDG methods

I prove worst-case guarantees on the size and quality of the mesh
Causality
Point A influences point B (A \prec B) iff changing the parameters at A could possibly change the solution at B.

Points influenced by A are approximated by a cone of influence with apex A and slope $1/\omega(A)$.

If neither $A \prec B$ nor $B \prec A$, then A and B are independent.
Spacetime element A influences spacetime element B (A \preceq B) iff some point of A influences some point of B.

If A \preceq B, then A must be solved no later than B.

If A \preceq B and B \preceq A; then, A and B are coupled and must be solved simultaneously.
A patch $\Pi$ is a set of elements that must be solved simultaneously.

A patch plus its inflow information is a solvable unit.

We generate meshes incrementally patch-by-patch so that

$$\text{Total computation time} = \sum_{\text{patch } \Pi \in \Omega} \text{Time to solve } \Pi$$

is bounded.
Advancing Front Meshing
Basic incremental algorithm

**Input:** triangulation of the $d$-dimensional space domain $M$ at time $t = 0$

**Output:** $(d + 1)$-dimensional spacetime mesh $\Omega$ of $M \times [0, \infty)$

Construct a sequence of fronts $\tau_0, \tau_1, \tau_2, \ldots$, embedded in spacetime—create small patches between successive fronts

A front is a terrain
Basic incremental algorithm

Initialize $\tau_0 \leftarrow M_{t=0}$

For $i = 0, 1, 2, \ldots$

Advance a local neighborhood $N$ of $\tau_i$ to $N'$

Triangulate the volume between $\tau_i$ and $\tau_{i+1}$

Solve the resulting patch

Each new front is obtained from the previous front by a local operation
Every front must be causal

A front is causal iff it is an independent set—no two points of a front influence each other.

The slope of a causal front is less than that of the cone of influence of each of its points.
Advances in time a vertex $P$ to $P'$

**Greedily** maximizes the **height** of the **tentpole** $PP'$ subject to causality

Old and new fronts causal $\implies$ patch can be solved immediately
Tent Pitcher plus

In 2D×time and higher dimensions, being too greedy at each step can prevent progress in future [Üngör and Sheffer, IMR'00]

Erickson et al. [Engg. with Computers, '05] devised progress constraints

Causality and progress constraints guarantee progress, proportional to local geometry, when a local minimum vertex is pitched
My improvements to Tent Pitcher

1. Adapt mesh resolution to error estimates in 2D×time

2. Adapt duration of spacetime elements to changing wavespeeds in $dD \times \text{time}$

All this requires new constraints and a new meshing algorithm which I derive and prove correct.
3. Adapt size and position of mesh features to track moving boundaries

I propose a set of more general front advancing operations required and useful for boundary tracking

I give a framework to perform various operations when possible and desirable
Technical Details
Progress constraint

Limit progress at each step to guarantee progress in the next step.
Next step is legal
Causality violated in next step
Forbidden zones

Progress constraint $1/\omega$ limits slope of $QR$ and of $PR$
Adapting to changing wavespeeds

Wavespeeds can be solution-dependent for nonlinear PDEs

Increasing wavespeed breaks old algorithm

At every step, the front must satisfy a progress constraint that anticipates the wavespeed in the next step

I derive a progressive invariant that guarantees progress even when wavespeed increases discontinuously

I give an algorithm to greedily maximize progress subject to this invariant
No focusing: $\omega(P) \leq \max_{P \in \text{cone}(Q)} \{ \omega(Q) \}$

Allow us to conservatively estimate future wavespeed given the cone of influence everywhere on the current front.
Handling faster wavespeeds

When wavespeed is not constant, we cannot be greedy because the amount of progress in the current step depends on the future wavespeed.

I give an algorithm to look ahead $h$ tent pitching steps to conservatively estimate future wavespeed.

The new algorithm maximizes tentpole height in the current step subject to this conservative estimate.
Look ahead one step

\[ \| \nabla PQR \| < \frac{1}{\omega(PQR)} \]

\[ \frac{1}{\omega(PQR)} \]

\[ \frac{1}{\omega(PQR)} \]

\[ \frac{1}{\omega(PQR)} \]
Lookahead one step
Look ahead one step
Invariant: $h$-progressive

Let $\delta > 0$ be a function of the shape of $\triangle pqr$. Triangle $PQR$ is $h$-progressive iff it is causal and

$h = 0$: $PQR$ is 0-progressive if it satisfies progress constraint $1/\omega_{\text{max}}$

$h > 0$: $PQR$ is $h$-progressive if after pitching local minimum $P$ of $PQR$ to $P'$ by $\delta$, $\triangle P'QR$ is $(h-1)$-progressive

Horizon $h$ limits number of lookahead steps
When wavespeed is not constant, most limiting cone constraint can be nonlocal
Refinement and coarsening

Newest vertex bisection [Sewell ‘72, Mitchell ‘88]

Bisecting triangles on the current front decreases size of future spacetime elements

Coarsening ≡ De-refinement
Gradient vector is excluded from eight forbidden zones
Linear adaptive example
Mesh adaptivity with nonlinearity

I give an algorithm to adapt size (to error estimates) and duration (to changing wavespeed) of elements

Unify (i) adaptive progress constraint, and (ii) lookahead algorithm for meshing with nonlocal cone constraints

During lookahead, if the algorithm predicts a bad quality element due to increasing wavespeed, then it can preemptively refine the front

Refinement improves or maintains temporal aspect ratio

More progress can be made in the current step if the algorithm prepares for future refinement
New invariant: \((h,l)\)-progressive

A front \(\tau\) is \((h,l)\)-progressive if and only if it is causal and

1. \(\tau\) is \(h\)-progressive; and

2. \(\text{bisect}(\tau)\) is \((h, \max\{l - 1, 0\})\)-progressive.

**Base case** \(l = 0\): \(\tau\) is \((h, 0)\)-progressive if \(\text{bisect}^k(\tau)\) is \(h\)-progressive for every \(k \geq 0\).

Progress constraint adapts to the level of refinement \(l\)
Example II
Main result

Given a simplicial mesh $M \in \mathbb{E}^d$, our algorithm builds a simplicial mesh $\Omega$ of the spacetime domain $M \times [0, \infty)$ that satisfies all the following criteria:

For every $T \geq 0$ the spacetime volume $M \times [0, T]$ is contained in the union of a finite number of simplices of $\Omega$

The minimum temporal aspect ratio of any spacetime element is bounded from below
Additionally, in 2D×Time,

Our algorithm adapts the size of spacetime elements to a *posteriori* error estimate

Provided the number of refinements is finite, our algorithm terminates with a finite mesh of $M \times [0, T]$ for every target time $T \geq 0$

Other results omitted from this talk
New meshing operations

Inclined tentpoles

Edge bisection

Edge flips

Vertex deletion
Boundary tracking

Geometry and topology of domain changes over time e.g., combustion of solid rocket fuel

The mesh must adapt by changing the size of mesh elements, or by varying the placement of mesh features, or both

We incorporate techniques like smoothing, usually performed as a global remeshing step, into our local advancing framework
I devised a policy to assign priorities to various front advancing operations:

1. Coarsen the front by deleting a vertex $u$

2. Flip an edge of the front if it improves the spatial aspect ratio

3. Pitch an interior vertex to smooth local triangulation

4. Pitch a boundary vertex in prescribed direction
Additional heuristics

If an obtuse angle gets too big, flip or bisect the opposite edge

Bisect an edge if its endpoints have very different velocities; choose an average velocity for new midpoint

Each operation performed only when allowed by the solver
An example
Conclusion
Adapt mesh resolution to a posteriori numerical error estimate

**Spacetime Meshing with Adaptive Refinement and Coarsening**
Reza Abedi, Shuo-Heng Chung, Jeff Erickson, Yong Fan, Michael Garland, Damrong Guoy, Robert Haber, John M. Sullivan, Shripad Thite, Yuan Zhou
20th Symp. Computational Geometry, June 2004

**An \( h \)-adaptive Spacetime-Discontinuous Galerkin Method for Linearized Elastodynamics**
Reza Abedi, Robert Haber, Shripad Thite, Jeff Erickson
Submitted to *Revue Européenne des Éléments Finis*, 2005
Summary of results (cont’d.)

Adapt to changing wavespeeds

**Efficient Spacetime Meshing with Nonlocal Cone Constraints**
Shripad Thite
13th International Meshing Roundtable, September 2004

**A Unified Algorithm for Adaptive Spacetime Meshing with Nonlocal Cone Constraints**
Shripad Thite
21st European Workshop on Computational Geometry, March 2005
Invited to special issue of *Computational Geometry: Theory and Applications (CGTA)*
Summary of results (cont’d.)

Give more general meshing operations in spacetime

Propose a framework for smoothing to improve mesh quality, and for tracking moving boundaries and other singular surfaces
My thesis extends Tent Pitcher to efficiently solve more general problems.

<table>
<thead>
<tr>
<th>Nonadaptive linear</th>
<th>Nonadaptive nonlinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive linear</td>
<td>Adaptive nonlinear</td>
</tr>
</tbody>
</table>

Produce an efficiently solvable non-degenerate mesh in all these cases.

Smoothing and boundary tracking do not create inverted elements.

If the front triangles degrade in spatial quality more than expected, the algorithm can get stuck.
Some future directions

Extend algorithms to arbitrary dimensions, first 3D×time

   e.g. *Devise an adaptive meshing algorithm in 3D×time*

Give a provably correct and complete boundary tracking algorithm for interesting classes of motion

Handle changes in topology of domain over time
Thank you!