Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 208 (2005) 616–625
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Adaptive unstructured volume remeshing – I: The method

Anthony Anderson [a,1], Xiaoming Zheng [b,2], Vittorio Cristini [c,*]

[a] Department of Chemical Engineering and Materials Science, University of Minnesota, United States
[b] Department of Mathematics, University of California, Irvine, CA 92697-2715, United States
[c] Department of Biomedical Engineering, REC 204, University of California, Irvine, CA 92697-2715, United States

Received 21 October 2004; received in revised form 28 February 2005; accepted 28 February 2005
Available online 11 May 2005

Abstract

We present an adaptive remeshing algorithm for meshes of unstructured triangles in two dimensions and unstructured tetrahedra in three dimensions. The algorithm automatically adjusts the size of the elements with time and position in the computational domain in order to resolve the relevant scales in multiscale physical systems to a user-prescribed accuracy while minimizing computational cost. The optimal mesh that provides the desired resolution is achieved by minimizing a spring-like mesh energy function that depends on the local physical scales using local mesh restructuring operations that include edge-swapping, element insertion/removal, and dynamic mesh-node displacement (equilibration). The algorithm is a generalization to volume domains of the adaptive surface remeshing algorithm developed by Cristini et al. [V. Cristini, J. Blawzdziewicz, M. Loewenberg, An adaptive mesh algorithm for evolving surfaces: simulations of drop breakup and coalescence. J. Comp. Phys., 168 (2001) 445] in the context of deforming interfaces in two and three dimensions.

The remeshing algorithm is versatile and can be applied to a number of physical and biological problems, where the local length scales are dictated by the specific problem. In Part II [X. Zheng, J. Lowengrub, A. Anderson, V. Cristini, Adaptive unstructured volume remeshing – II: application to two- and three-dimensional level-set simulations of multiphase flow, J. Comp. Phys., in press], we illustrate the performance of an implementation of the algorithm in finite-element/level-set simulations of deformable droplet and fluid–fluid interface interactions, breakup and coalescence in multiphase flows.
© 2005 Elsevier Inc. All rights reserved.

* Corresponding author. Also at the Department of Mathematics. Formerly at the Department of Chemical Engineering and Materials Science and School of Mathematics, University of Minnesota, United States.
E-mail address: cristini@math.uci.edu (V. Cristini).
[1] Present address: Department of Applied Mathematics, Northwestern University, United States.
[2] Formerly at the School of Mathematics, University of Minnesota, United States.

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.02.023

1. Introduction

In many physical and biomedical problems, the evolution critically depends on the nonlinear interaction of phenomena in a complex geometry and over a wide range of length and time scales. For example, in turbulent flows energy is exchanged among the large and small scales. In multiphase flows, the nonlinear coupling between macro scale driving forces (e.g., gravity, applied shear flow), micro scale lubrication pressures (between interfaces) and nano scale van der Waals attractive forces strongly affects drop and interface coalescence and breakup rates and the overall topology and rheological properties of the flow. In materials science, analogous coupling arises between macro scale driving forces (e.g., supersaturation, diffusion, elastic stress) and micro and nano scale physics (e.g., defects, interface anisotropy and kinetics) during crystal growth, thin film deposition and alloy formation. In biology, the distribution of chemical species at the micro scale affects and is affected by cellular, tissue and organ structure and function at the meso and macro scales.

Accurately capturing physical phenomena over wide ranges of scales, such as those described above, requires the use of robust and efficient adaptive mesh refinement algorithms. Thus, there has been much recent research in this direction. Adaptive mesh refinement has typically used two basic strategies: (i) moving mesh and mesh mapping methods and (ii) local refinement algorithms. In the former, the physical space mesh is evolved and/or deformed so as to cluster grid points in specific regions where high resolution is deemed necessary. See for example the texts by Baines [3] and Zegeling [48]. Recent applications of this approach include hyperbolic conservation laws [24,42], Hamilton–Jacobi equations [43], Boussinesq flow [10], drop formation [46] and deformation [51], and the nonlinear Schroedinger equation [8,33]. In several recent implementations, the computational mesh is determined by the solution of a global Euler–Lagrange equation for a mesh energy functional (e.g., see [10,24,26,33]). Adaptive moving meshes have even been constructed using the level-set method [27,37]. In locally adaptive mesh refinement methods (AMR), a new mesh is constructed directly in physical space by adding or removing computational elements (or fine grids) to achieve a desired level of accuracy. See for example the text [9] and the review [30]. The level of accuracy can be determined according to a number of criteria including a posteriori error estimates (e.g., see the review [5]) as well as the maximum variation of a quantity across a mesh element. Recent applications of AMR include hyperbolic conservation laws [20,21], the Stokes and Navier–Stokes equations [1,4,5,12,22,31] and thermal and reactive flows [5,23]. In the context of fluid interfaces, AMR has been used in immersed boundary methods [11,36], front tracking [13,16,19,44], volume of fluid methods [18,45], phase-field methods [17,25,32] and level-set methods [39–41]. We note that the literature on the mesh mapping and AMR is very extensive and we have cited above only the most recent articles and review articles when possible.

Both adaptive mesh approaches discussed above have advantages and drawbacks. For instance, in mapping methods, data need not be interpolated at new physical space grid points, unless grid points are added, as the solutions are parametrized with respect to the fixed reference mesh. However, the physical equations are supplemented with additional equations (e.g., Euler–Lagrange equations) and boundary conditions for the grid mapping, which must then be solved simultaneously. These equations can be expensive to solve. In local refinement methods, on the other hand, no additional equations are imposed. However, interpolation/extrapolation is required to define the solutions on the fine/coarse mesh. Further, care must be taken to stably discretize the system, particularly in the finite difference context (e.g., see [6,7]).

In this paper, we present a hybrid adaptive mesh algorithm using unstructured triangles in 2-D and unstructured tetrahedra in 3-D that combines the favourable features of both the mapping and refinement approaches. The algorithm is a generalization to volume domains of the adaptive surface remeshing algorithm developed by Cristini et al. [12] in the context of deforming interfaces in two and three dimensions.

The optimal mesh that provides the desired resolution is achieved by minimizing a spring-like mesh energy function by performing local mesh restructuring operations (edge-swapping and mesh-element

insertion/removal) and dynamic mesh-node displacement (mapping step). The mesh is viewed as a system of massless, overdamped linear springs each with a space- and time-dependent resting length $L(\mathbf{x}, t)$. The mesh energy is the spring energy for this system and the dynamic node displacement (relaxation) arises via mesh-evolution using an overdamped Hooke's law that results in a decreased mesh energy. This mapping step is performed only locally in contrast to other mesh mapping methods. Since relaxation typically yields only local minima, the local restructuring operations are introduced to achieve the global minima. The resulting algorithm is efficient and has a cost that scales linearly with the total number of computational elements. By combining local mesh refinement and a physically based, energy-minimizing mesh-moving step, we achieve, for the first time to our knowledge, an optimal mesh characterized by *both* the desired local element size and high element quality (nearly equilateral elements) that guarantees minimization of the numerical error in finite-element calculations [49]. In particular, our unstructured adaptive mesh is optimal for discretization of arbitrary domain geometry and deforming and evolving interfaces.

The link between our remeshing algorithm and the specific physical problem to be simulated is the choice of the resting length L. This is the minimum length scale to be resolved in the simulation. Our algorithm is versatile in that L can be chosen using a variety of different criteria. For example, using a posteriori error estimates the local length scale L can be determined. Or, the specific scales that arise in an application, e.g., fluid dynamics, materials science and biology, can be directly resolved. Very often, these scales can be associated to gradients of physical quantities, e.g., flow vorticity, component concentrations during crystal growth and alloy formation, cell densities and concentrations of chemical species during growth in tissue. In Part II [49], we present an implementation of the unstructured adaptive remeshing algorithm in finite-element/level-set simulations of droplet and fluid–fluid interface interactions and coalescence in multiphase flows.

The adaptive mesh refinement algorithm is described in Section 2 and conclusions and future work are in Section 3.

2. Adaptive mesh algorithm

2.1. Resolution of the physical scales

For a given problem, one can define $L(\mathbf{x}, t)$ to be the space- and time-dependent smallest local length scale to be resolved in simulations. Accurate computation then requires that the local mesh size be equal to L. A disadvantage of using nonadaptive meshes (uniform and constant mesh size) is that accuracy of computation is associated with great expense since the mesh size needs to be equal to the minimum physical scale over the entire computational domain and time interval of computation.

In our adaptive remeshing strategy, we consider a computational mesh consisting of unstructured triangles in 2-D and unstructured tetrahedra in 3-D. The algorithm described in the following of Section 2 is a generalization to two- and three-dimensional volume domains of the algorithm developed in [12] for two- and three-dimensional surfaces discretized with boundary elements. In particular, our two-dimensional remeshing algorithm reduces to that in [12] applied to a flat surface. Following [12], we model every mesh element edge as a damped massless spring with tension

$$\gamma = h - \langle L \rangle, \tag{1}$$

where h is the local edge length and $\langle L \rangle$ is an appropriate weighted average of the continuous field L setting the desired edge length as described below. We seek an optimal mesh that minimizes the spring energy

$$E = \sum_{\text{edges}} \gamma^2. \tag{2}$$

To calculate local weighted averages $\langle L \rangle$, following [12], we define a local node density in terms of the local length scale:

$$\beta = \alpha L^{-d}, \tag{3}$$

where $d = 2, 3$ is the dimension and $\alpha = 2/\sqrt{3}$ in 2-D and $\alpha = \sqrt{2}$ in 3-D. In 2-D triangulated meshes, we straightforwardly use the averaging techniques developed in [12]. In 3-D tetrahedral meshes, after some experimentation following a similar approach, we define an optimal tetrahedral volume from geometrical considerations: $\langle V \rangle = (2/3)(\sum_{i=1}^{4} \beta_i)^{-1}$, where the local desired node density β_i is calculated from Eq. (3) for the four vertices of the tetrahedron. The optimal length of an edge is then: $\langle L \rangle = (6\sqrt{2})^{1/3} (\sum_{j=1}^{N_T} \langle V_j \rangle / N_T)^{1/3}$, where $\langle V_j \rangle$ are the optimal volumes for the N_T tetrahedra sharing that edge.

2.2. Mesh motion

For a given mesh topology, following [12], we achieve a local minimum of the energy (2) by dynamically displacing the computational nodes with a velocity

$$\dot{\mathbf{x}} = \sum_{j=1}^{N_c} \hat{\mathbf{e}}_j \gamma_j \tag{4}$$

that brings spring tensions to equilibrium ($|\dot{\mathbf{x}}| \to 0$). The summation is over the N_c edges that emanate from the node, $\hat{\mathbf{e}}_j$ is the unit vector parallel to edge j, and γ_j is the edge tension (1). The equilibration process is executed by pseudo-time-stepping Eq. (4) and is terminated when

$$|\dot{\mathbf{x}}| < \eta L. \tag{5}$$

Equilibration is performed over the entire mesh or locally on the nodes directly affected by mesh restructuring operations [12]. In our implementation, we update these node positions at each time step using a Runge–Kutta scheme with $\eta = 0.05$. In most cases the equilibration criterion (5) is achieved after less than 10 iterations. Note that the parameter η is a dimensionless tolerance for the tension values in the optimal equilibrated mesh and thus is universal, independent of the absolute length scales L or h.

In order to prevent inversion of triangles or tetrahedra adjacent to a node we require that $|\dot{\mathbf{x}}|\Delta t \leqslant 0.25L$ (Δt is local equilibration time step), and that a local element shape measure has a lower bound. In 2-D, we use a Delaunay criterion [12] for the local element shape measure. In 3-D, we follow [28] and define the shape measure of a tetrahedron by

$$S = \operatorname{sign}(V) \frac{12(9V^2)^{1/3}}{\sum_{i=1}^{6} h_i^2}, \tag{6}$$

where V is the signed volume and h_i are the edge lengths of the tetrahedron. According to Eq. (6), valid tetrahedra (not inverted) have $0 < S \leqslant 1$, with $S = 1$ indicating an equilateral tetrahedron. After equilibration, we find that usually $S > 0.5$ for all tetrahedra adjacent to the node being displaced. During equilibration, this value is checked and an equilibration step is not performed if it leads to violate this condition.

Equilibration using this algorithm typically leads to equilibrium of tensions at all nodes. This corresponds to a local minimum of the energy (2). The tensions themselves are in general nonzero due to topological constraints and thus a global equilibrium is typically not achieved by dynamic node displacements alone. This implies that the local mesh size is not equal to the length scale L. To further reduce the mesh energy towards a global minimum, it is necessary to perform energy-reducing local mesh-restructuring operations.

2.3. Mesh restructuring operations

2.3.1. Node addition and subtraction

In regions of the mesh that are too sparse (dense) compared to the desired mesh density β defined in Eq. 3, node and element addition (subtraction) provides access to configurations with lower energy that are not accessible by equilibration alone.

In 2-D, we use techniques [12] that allow insertion of three new nodes at once at edge midpoints and subtraction of one node. In 3-D, one new node is inserted using an edge bisection technique (see also [2,29,34,35]) illustrated in Fig. 1(a). The reverse operation, collapsing two nodes into one, is used for mesh coarsening (e.g. [14]).

Edge tensions γ defined in Eq. (1) identify edges that are overly extended or compressed thus providing node addition/subtraction criteria: addition/subtraction are performed until $-\eta_1 L < \gamma < \eta_2 L$ for all edges. These tolerances are universal as η described above, and are necessary to prevent mesh oscillations between node addition and subtraction. Following [12], we select $\eta_1 = 0.5$ and $\eta_2 = 0.5$ after some experimentation. During node collapsing the position of the new node that replaces the two parent nodes along the shared edge is determined by maximizing the shape measure S defined in Eq. (6) for all adjacent tetrahedra (if $S \leqslant 0$ the nodes are not collapsed).

Node addition/subtraction always reduces the mesh energy. To optimize the number of additions/subtractions, local equilibration and reconnection (see below) are performed on the region directly affected by the addition/subtraction.

2.3.2. Reconnection

Node reconnection performed by edge swapping decreases the mesh energy by eliminating topological constraints produced by poorly shaped elements. We have implemented edge swapping for $4 \leqslant N_T \leqslant 6$ (N_T is the number of tetrahedra that share that edge, and $N_T > 6$ rarely occurs). The possible reconfigurations for these cases are shown in Fig. 1(b). Edge swapping with $N_T = 4$ is illustrated in Fig. 1(c). In Fig. 1(b), the number of tetrahedra sharing the edge before swapping is indicated by N_T for three cases. Below the corresponding schematics, the number of tetrahedra and of possible reconfigurations after swapping are reported. The schematics are projected perpendicular to the swapped edge and represent, for each case, all possible rotation-invariant reconfigurations.

In 2-D, a Delaunay criterion [12], and in 3-D, maximization of the shape measure S from Eq. (6) is used to decide whether to perform node reconnection [15].

The mesh achieved by performing iteratively equilibration, node addition/subtraction and reconnection has an optimal number of nearly equilateral elements that ensures the local resolution of the length scale L.

2.4. Example: 3-D mesh adaption about a sphere

In this section, we illustrate the adaptive mesh refinement algorithm by resolving a narrow region about a sphere in 3-D. A sphere of radius $a = 5$ is placed in an initially unstructured, tetrahedral mesh with nearly uniform edge-lengths discretizing a spherical volume domain with radius $R = 20$. The local length scale is prescribed as the distance L_Σ from the spherical interface Σ, i.e.

$$L(\mathbf{x}, t) = L_\Sigma = \min(h_0, h_1 + s\,\mathrm{dist}(\mathbf{x}, \Sigma)), \tag{7}$$

where h_1 is the imposed linear size of the computational elements in the sub-domain of elements that contain the interface Σ, and the slope s is set to smoothly increase the element size from h_1 to the uniform size $h_0 > h_1$ away from Σ.

A cross-section of the domain is shown in Fig. 2 before (a) and after (b) refinement. Those tetrahedra that contain the cross-section are visualized. Since tetrahedra are projected onto the plane of the figure,

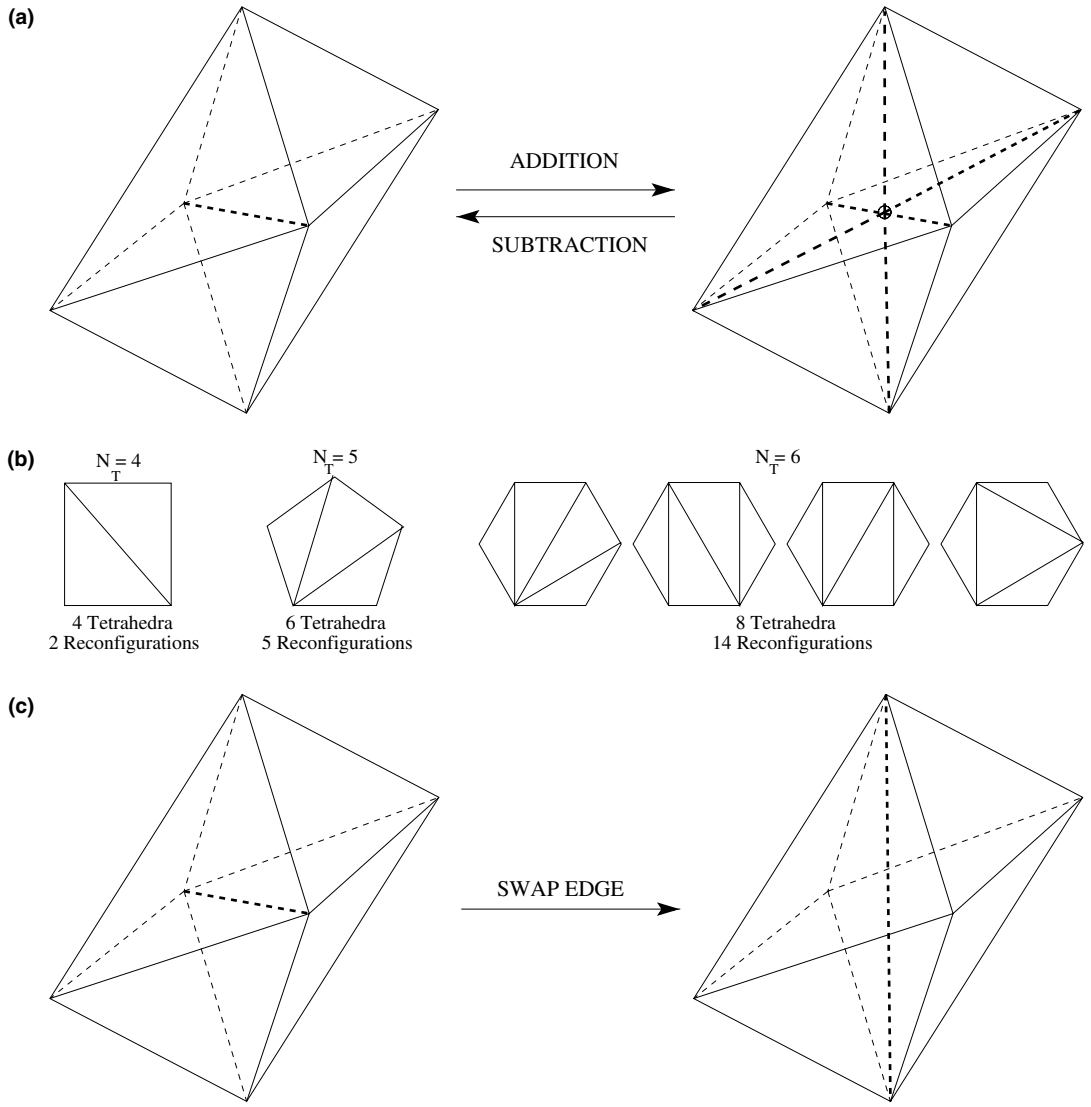

Fig. 1. Local mesh restructuring operations. Node addition via edge bisection and subtraction via collapsing two nodes (a). All possible rotation-invariant configurations (b) projected onto a plane perpendicular to the swapped edge, for 4, 5, and 6 tetrahedra sharing the edge before swapping. Swapping (c) of an edge (bold) surrounded by 4 tetrahedra. The result is 4 new tetrahedra.

some may appear skewed. The mesh parameters in Eq. (7) were chosen to be $h_0/a = 0.71$, $h_1/a = 0.0426$ and $s = 0.8$. In (c), a blow-up by a factor $h_0/h_1 \approx 16$ of the mesh near the interface is shown. Note that the blown-up tetrahedra size and shape are roughly the same as those in (a), thus confirming adaptivity according to Eq. (7) and mesh quality invariance. The tetrahedra size quickly grows away from the interface.

To perform the refinement, 14 remeshing steps were used each combining equilibration and all mesh restructuring operations. We reduced h_1/a successively from 0.71 to the final value 0.0426 in equal decrements. In Fig. 2(d), the mesh energy E divided by the total number of edges calculated from Eq. (2) is shown as a function of remeshing step. The energy monotonically decreases although at a non constant

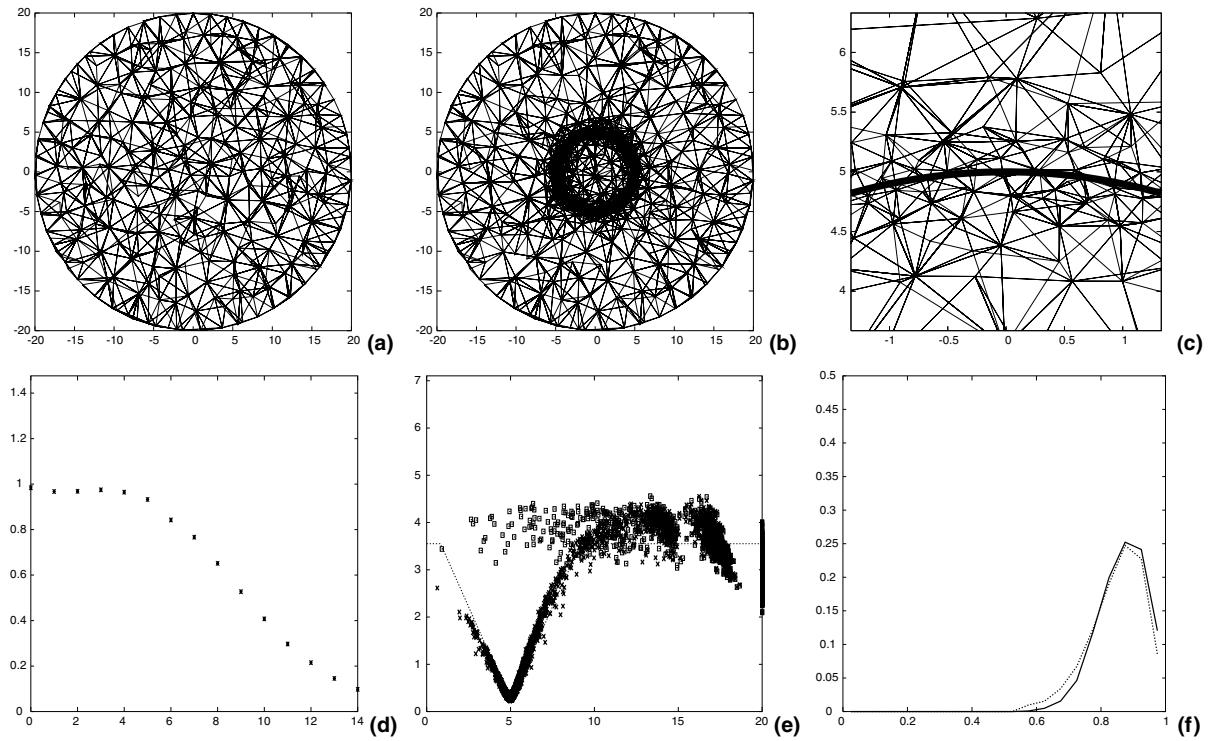

Fig. 2. Mesh adaption about a sphere. Cross-section (a) of an unstructured, nearly uniform mesh discretizing a spherical domain. Tetrahedra containing the cross-section are visualized. After 14 remeshing steps, final mesh (b) refining a spherical interface (radius $a = 5$) according to Eq. (7) with mesh parameters $h_0/a = 0.71$, $h_1/a = 0.0426$, $s = 0.8$. Blow-up (c) of the mesh around the interface by a factor of $h_0/h_1 \approx 16$. Mesh-energy (d) averaged over all the edges during adaption. Initial (□) and final (×) edge lengths (e) vs radial position; prescribed length scale L_Σ (——) from Eq. (7). Percentage (f) of tetrahedra (y-axis) with a given tetrahedral shape (x-axis) calculated from Eq. (6), for initial (solid) and final (dashed) meshes.

rate. In Fig. 2(e), the initial (squares) and final (crosses) edge lengths h are shown as a function of radial position in the spherical domain together with the prescribed local length scale (solid curve) from Eq. (7), which is minimum at the spherical surface Σ. It is clear that the prescribed length scale distribution is achieved after remeshing.

Mesh equilibration and restructuring operations are performed according to criteria that prevent mesh degradation. The quality of the elements and important topological quantities, like the tetrahedral shape distribution and the nodal and tetrahedral coordination numbers, are preserved in the execution of the mesh refinement algorithm. Fig. 2(f) shows the distribution of tetrahedral shape measure S defined in Eq. (6) for the initial (solid) and final (dashed) meshes. The graphs nearly overlap showing that the shape quality is fully preserved.

3. Conclusions

We have developed an adaptive remeshing algorithm for volume domains discretized using meshes of unstructured triangles (2-D) and unstructured tetrahedra (3-D). This work is a generalization of the adaptive surface remeshing algorithm developed by Cristini et al. [12], which was applied by these authors to

perform accurate boundary-integral calculations of multiphase flows. We have developed a new remeshing tool that is applicable to a wider range of problems involving more complex physics that those approachable by boundary-integral techniques, such as for example multiphase flows in the presence of significant inertial effects solved using the finite-element method. The algorithm automatically resolves the relevant scales in multiscale physical systems to a user-prescribed accuracy while minimizing computational cost, by continuously adjusting the size of the elements with time and position in the computational domain. This optimal mesh is maintained throughout simulations by minimizing a spring-like mesh energy function that depends on the local physical scales using local mesh restructuring operations that include edge-swapping, element insertion/removal, and dynamic mesh-node displacement (equilibration). Here for simplicity, we have prescribed a length scale proportional to the distance from the interfaces to set the local size of the mesh elements.

In Part II [49], we illustrate the performance of an implementation of the algorithm in finite-element/level-set simulations of deformable droplet and fluid–fluid interface interactions, breakup and coalescence in multiphase flows with and without inertia. We compare our simulation results to experiments and to theoretical and sharp-interface (boundary-integral) numerical results, demonstrating that the wide range of length scales characterizing the dynamics are accurately resolved, while the computational cost is found to be competitive even with respect to boundary-integral methods.

The remeshing algorithm is versatile and can be applied to a number of physical and biological problems, where the local length scales are dictated by the specific problem. We have also applied our adaptive remeshing algorithm to multiphase flow simulations using a combined level-set/volume-of-fluid method [47]. Beyond the context of multiphase flow, the remeshing algorithm is currently being applied to material science and biomedical problems, including multiscale simulations of tumor progression [50,38] and of replacement tissue growth on scaffolds (X. Zheng and J. Lowengrub, In preparation).

Acknowledgements

V.C. acknowledges funding from the National Science Foundation – Division of Mathematical Sciences. Part of this work is contained in AA's undergraduate honors thesis (2004) at the University of Minnesota. AA acknowledges the Undergraduate Research Opportunity Program at U Minnesota for support. The authors also acknowledge the Minnesota Supercomputer Institute, and the Departments of Biomedical Engineering and Mathematics and the Network and Academic Computing Services, UC Irvine, for computing resources.

References

[1] A. Almgrenm, P. Colella, J. Bell, L.H. Howell, M. Welcome, A conservative adaptive projection method for the incompressible Navier–Stokes equations in three dimensions, J. Comp. Phys. 142 (1999) 1.
[2] D. Arnold, A. Mukherjee, L. Pouly, Locally adapted tetrahedral meshes using bisection, SIAM J. Sci. Comput. 22 (1999) 431–448.
[3] M.J. Baines, Moving Finite Elements, Clarendon Press, Oxford, 1994.
[4] E. Bansch, P. Morin, R.H. Nochetto, An adaptive Uzawa fem for the stokes problem: convergence without the inf-sup condition, SIAM J. Num. Anal. 40 (2002) 1207.
[5] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, Acta Numerica 10 (2001) 1.
[6] M. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comp. Phys. 82 (1989) 62.
[7] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comp. Phys. 53 (1984) 484.
[8] C.J. Budd, M.D. Piggott, The geometric integration of scale-invariant ordinary and partial differential equations, J. Comp. Appl. Math. 128 (2001) 399.

[9] G.F. Carey, Computational Grids: Generation, Refinement and Solution Strategies, Taylor and Francis, Bristol, 1997.
[10] H.D. Ceniceros, T.Y. Hou, An efficient dynamically adaptive mesh for potentially singular solutions, J. Comp. Phys. 172 (2001) 609.
[11] H.D. Ceniceros, A.M. Roma, Study of the long-time dynamics of a viscous vortex sheet with a fully adaptive non-stiff method, Phys. Fluids, in review.
[12] V. Cristini, J. Blawzdziewicz, M. Loewenberg, An adaptive mesh algorithm for evolving surfaces: simulations of drop breakup and coalescence, J. Comp. Phys. 168 (2001) 445.
[13] M.Z. Dai, H.S. Wang, J.B. Perot, D.P. Schmidt, Direct interface tracking of droplet deformation, Atom. Sprays 12 (5–6) (2002) 721.
[14] H.L. de Cougny, M.S. Shephard, Parallel refinement and coarsening of tetrahedral meshes, Int. J. Num. Meth. Eng. 46 (1999) 1101–1125.
[15] L.A. Freitag, C. Ollivier-Gooch, Tetrahedral mesh improvement using swapping and smoothing, Int. J. Num. Meth. Engng. 40 (1997) 3979–4002.
[16] O.S. Galaktionov, P.D. Anderson, G.W.M. Peters, F.N. van de Vosse, An adaptive front tracking technique for three dimensional transient flows, Int. J. Num. Meth. Fluids 32 (2000) 201.
[17] H. Garcke, M. Rumpf, U. Weikard, The Cahn–Hilliard equation with elasticity: finite element approximation and quantitative studies, Int. Free Bound, in press.
[18] I. Ginzberg, G. Wittum, Two phase flows on interface refined grids modeled with VOF, staggered finite volumes and spline interpolants, J. Comp. Phys. 166 (2001) 302.
[19] J. Glimm, J.W. Grove, X.L. Li, R. Young, Q. Zhang, Y. Zeng, Three dimensional front tracking, J. Sci. Comp. 19 (1998) 703.
[20] R. Hartmann, P. Houston, Adaptive discontinuous galerkin finite element methods for nonlinear hyperbolic conservation laws, SIAM J. Sci. Comp. 24 (2002) 979.
[21] P. Houston, B. Senior, E. Suli, HP-discontinuous Galerkin finite element methods for hyperbolic problems: error analysis and adaptivity, Int. J. Num. Meth. Fluids 40 (2002) 153.
[22] L.H. Howell, J. Bell, An adaptive mesh projection method for viscous incompressible flow, SIAM J. Sci. Comput. 18 (1997) 996.
[23] L.H. Howell, J.A. Greenough, Radiation diffusion for multi-fluid Eulerian hydrodynamics with adaptive mesh refinement, J. Comp. Phys. 184 (2003) 53.
[24] W. Huang, R.D. Russel, Moving mesh strategy based on a gradient flow equation for two-dimensional problems, SIAM J. Sci. Comput. 20 (1999) 998.
[25] X. Jiang, R. Nochetto, C. Verdi, A p^1–p^1 finite element method for a phase relaxation model. Part II: adaptively refined meshes, SIAM J. Num. Anal. 36 (1999) 974.
[26] R. Li, T. Tang, P. Zheng, A moving mesh finite element algorithm for singular problems in two and three dimensions, J. Comp. Phys. 177 (2002) 365.
[27] G. Liao, F. Liu, C. de la Pena, D. Peng, S. Osher, Level-set based deformation methods for adaptive grids, J. Comp. Phys. 159 (2000) 103.
[28] A. Liu, B. Joe, On the shape of tetrahedra from bisection, Math. Comp. 63 (1994) 141.
[29] A. Liu, B. Joe, Quality local refinement of tetrahedral meshes based on bisection, SIAM J. Sci. Comput. 16 (1995) 1269–1291.
[30] D.J. Mavriplis, Unstructured grid techniques, Ann. Rev. Fluid Mech. 29 (1997) 473.
[31] M. Minion, A projection method for locally refined grids, J. Comp. Phys. 127 (1996) 158.
[32] N. Provatas, N. Goldenfeld, J. Dantzig, Adaptive mesh refinement computation of solidification microstructures using dynamic data structures, J. Comp. Phys. 148 (1999) 265.
[33] W. Ren, X.-P. Wang, An iterative grid redistribution method for singular problems in multiple dimensions, J. Comp. Phys. 159 (2000) 246.
[34] M.C. Rivara, Selective refinement/derefinement algorithms for sequences of nested triangulations, Int. J. Num. Meth. Eng. 28 (1989) 2889–2906.
[35] M.C. Rivara, A 3-d refinement algorithm suitable for adaptive and multigrid techniques, Commun. Appl. Numer. Meth. 8 (1992) 281–290.
[36] A.M. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, J. Comp. Phys. 153 (1999) 509.
[37] J. Sethian, Curvature flow and entropy conditions applied to grid generation, J. Comp. Phys. 115 (1994) 440.
[38] J. Sinek, H. Frieboes, X. Zheng, V. Cristini, Two-dimensional chemotherapy simulations demonstrate fundamental transport and tumor response limitations involving nanoparticles, Biomed. Microdev. 6 (2004) 297–309.
[39] V. Sochnikov, S. Efrima, Level set calculations of the evolution of boundaries on a dynamically adaptive grid, Int. J. Num. Meth. Eng. 56 (2003) 1913.
[40] J. Strain, Tree methods for moving interfaces, J. Comp. Phys. 151 (1999) 616.
[41] M. Sussman, A. Almgren, J. Bell, P. Collela, L. Howell, M. Welcome, An adaptive level set approach for incompressible two-phase flows, J. Comp. Phys. 148 (1999) 81.

[42] H.Z. Tang, T. Tang, Moving mesh methods for one- and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2003) 487.
[43] H.Z. Tang, T. Tang, P. Zhang, An adaptive mesh redistribution method for nonlinear Hamilton–Jacobi equations in two- and three-dimensions, J. Comp. Phys. 188 (2003) 543.
[44] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front tracking method for the computations of multiphase flow, J. Comp. Phys. 169 (2001) 708.
[45] O. Ubbink, R.I. Issa, A method for capturing sharp fluid interfaces on arbitrary meshes, J. Comp. Phys. 153 (1999) 26.
[46] E.D. Wilkes, S. Phillips, O. Basaran, Computational and experimental analysis of dynamics of drop formation, Phys. Fluids 11 (1999) 3577.
[47] X. Yang, A.J. James, J. Lowengrub, X. Zheng, V. Cristini, An adaptive coupled level-set/volume-of-fluid interface tracking method for unstructured triangular grids, in press.
[48] P.A. Zegeling, Moving Grid Methods, Utrecht Univ. Press, Utrecht, 1992.
[49] X. Zheng, J. Lowengrub, A. Anderson, V. Cristini, Adaptive unstructured volume remeshing – II: application to two- and three-dimensional level-set simulations of multiphase flow, J. Comp. Phys., in press.
[50] X. Zheng, S. Wise, V. Cristini, Nonlinear simulation of tumor necrosis, neo-vascularization and tissue invasion via and adaptive finite-element/level-set method, Bull. Math. Biol. 67 (2005) 211–259.
[51] A.Z. Zinchenko, M.A. Rother, R.H. Davis, A novel boundary integral algorithm for viscous interaction of deformable drops, Phys. Fluids 9 (1997) 1493.