Oil & Gas Science and Technology – Rev. IFP, Vol. 54 (1999), No. 2, pp. 169-173

A Parallel, Unstructured-Mesh Methodology for Device-Scale Combustion Calculations

P. J. O'Rourke[1], S. Zhang[1] and M. S. Sahota[1]

[1] Group T-3, MS B216, Los Alamos National Laboratories, Los Alamos, NM 87545 - United States
e-mail: pjor@lanl.gov

Résumé — Méthodologie parallèle avec maillages destructurés pour des calculs de combustion-effet du nombre de processeurs — *Los Alamos National Laboratories* développent une méthodologie de calcul parallèle, destructurée et en volumes finis pour la mécanique des fluides numérique. Cette méthodologie est implantée dans le code Chad (Computational Hydrodynamics for Advanced Design). Dans cet article, on donne un aperçu de la méthodologie numérique de Chad, et une présentation des résultats de calculs parallèles d'écoulements dans un moteur Diesel à 4 soupapes. L'efficacité en fonction du nombre de processeurs est étudiée.

***Abstract* — *A Parallel, Unstructured-Mesh Methodology for Device-Scale Combustion Calculations* —** Los Alamos National Laboratories *is developing a parallel, unstructured-mesh, finite-volume CFD methodology for the simulation of chemically reactive flows in complex geometries. The methodology is embodied in the Chad (Computational Hydrodynamics for Advanced Design) code. In this report we give an overview of the Chad numerical methodology and present parallel scaling results for calculations of flows in a four-valve Diesel engine.*

INTRODUCTION

Computer simulation of flows in practical combustion devices, such as internal combustion engines, is one of the most difficult problems facing modern computational fluid dynamics because of the number of challenging issues that must be addressed before accurate simulations can be performed. These flows typically involve geometric complexity (complicated internal flow passages; moving boundaries, possibly with flow domain topology changes; small features, such as crevices and gaps, whose inclusion can be vital to correct flow field predictions), physical complexity (turbulence, combustion, multiphase flow, and radiative heat transfer), and numerical challenges (deforming meshes, large density and fluid property variations, thin boundary layers and narrow jets, and coupled Eulerian/Lagrangian algorithms). To help meet these challenges, we at Los Alamos are developing the Chad code for device-scale combustion applications.

Chad improves upon previous combustion codes developed by Los Alamos (e.g., Rice, Apache, Conchas, and Kiva) in three important respects. First, Chad uses hybrid unstructured grids in which cells in the mesh can be hexahedra or any solid figures—such as prisms, pyramids, or tetrahedra—that can be obtained by collapsing faces or edges of hexahedra to lines or points. This unstructured mesh capability will enable faster mesh generation in complex geometries and also improve mesh quality. Second, for computational efficiency Chad uses a variable explicit/implicit method for calculating advection (O'Rourke and Sahota, 1998), in addition to implicit treatment of diffusion and pressure wave propagation terms. When the computational timestep is small, a time-accurate explicit advection scheme is used; when the timestep is large, a variable amount of implicitness is added to maintain stability and monotonicity. This methodology improves computational efficiency in calculations of flows with embedded regions, such as boundary layers and jets, with advective time scales that are much smaller than problem time scales of interest.

Perhaps the largest difference between Chad and its predecessors, however, is that it is written to take full advantage of parallel computers. Calculations of flows in practical combustors are very demanding of computational resources, requiring high resolution and complex physical submodels. Current calculations are often under-resolved and use physical submodels that have known inaccuracies in many applications. Overcoming these deficiencies will require more computer power than is available with scalar and vector computers. Parallel and massively parallel platforms offer the promise of the large increase in computer power necessary for three-dimensional calculations of combustor flows. Because of the variety of today's, and undoubtedly tomorrow's, parallel computer architectures, Chad is written to be highly portable on computers having Fortran 90 compilers. Portability is realized, in part, by placing all data communication in a small number of gather/scatter routines in which all machine-dependent coding is isolated. For distributed memory platforms automatic domain decomposition using spectral graph partitioning is used to balance processor work load and minimize the cost of global data communication. In this case the gather/scatter routines utilize a message-passing library, written in standard MPI, for global data communication.

In this paper we first give an overview of the Chad numerical methodology. We describe the median mesh control volumes used by the finite-volume method and illustrate the differencing of a generic transport equation. Then we present an example calculation of cold flow in a 4-valve, small-bore Diesel engine. Parallel scaling results are given for calculations on a multi-processor distributed-memory computer.

1 CHAD NUMERICAL METHODOLOGY

The Chad computer code uses a node-centered finite volume method to solve the Navier-Stokes equations for a multicomponent mixture of ideal gases. When calculating turbulent flows, the equations are supplemented by those of a standard k/ε turbulence model. For applications with fuel sprays, these equations can be coupled to equations for a vaporizing liquid spray. With minor exceptions, the Chad equations are those solved by the Kiva-II code and will not be given here. We also do not describe here the numerical particle method for solving the spray equations. Our intent in this paper is to give an overview of the solution method for the gas-phase equations.

1.1 The Median Mesh

In Chad, conserved variables are fundamentally located at the nodes, or vertices, of a computational mesh that is composed of logical hexahedra, called elements, whose faces and edges may degenerate to lines or points. The control volumes used to approximate the integral conservation equations are the so-called "median mesh" control volumes. These are the union of a sub-volume of each of the elements touching the node. Each element is subdivided into eight hexahedral sub-volumes, one for each node. The sub-volume associated with a node, as illustrated in Figure. 1, has as its vertices the node itself, the three mid-points of the edges touching the node, three mid-points of the faces touching the node, and the centroid of the element. Figure 2 gives a schematic picture of the median mesh control volume surrounding a node in a two-dimensional mesh. We denote this volume by V_ν, where ν is a nodal index.

Fluxes are computed for each face of a median mesh control volume, and we define one face for each edge in the mesh. A median-mesh face is the union of all sub-volume faces that touch the associated edge. Figure 2 illustrates a median-mesh face in a two-dimensional mesh. Quantities associated with an edge will be denoted by subscript α. The area projection vector for face α will be denoted by $\mathbf{A}_\alpha$ and will be directed outward from the node-ν control volume.

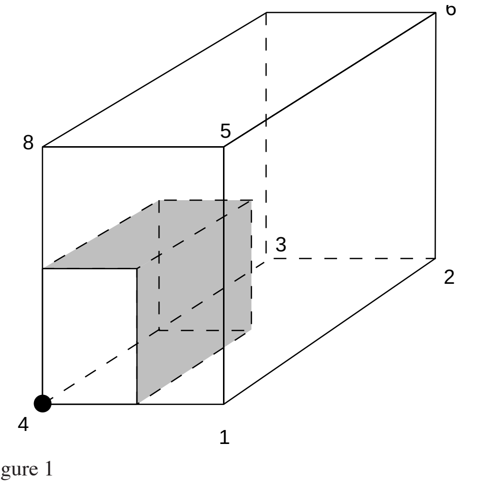

Figure 1

The subvolume of an element that lies in a median mesh control volume of node 4. The shaded surfaces are surfaces of the median mesh control volume.

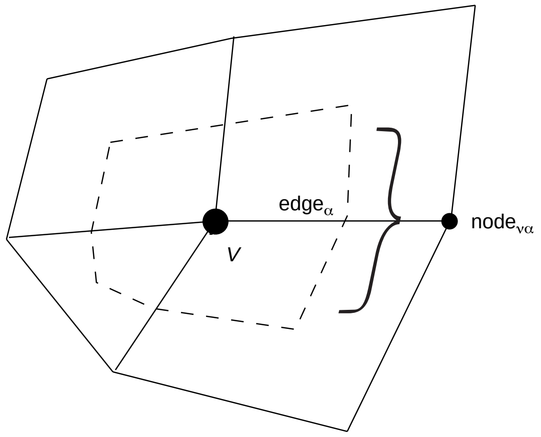

Figure 2

Median mesh control volume (dashed lines) surrounding node ν in a two-dimensional mesh.

Note that although $\mathbf{A}_\alpha$ is associated with both an edge and a node, we drop the node subscript for brevity. Each edge is shared by two control volumes, and we denote by ν_α the node on the opposite side of face α from the node ν under consideration.

An edge-based computational procedure is used in which a list is kept of the edges of all elements and of the two nodes that bound each edge. A typical computational sequence involves performing three DO-loops over the edge list in which we first gather necessary information from nodal arrays to edge arrays, second perform calculations using the gathered information (e.g. calculate fluxes), and third scatter the calculated information back to nodal arrays while performing an operation (e.g. adding fluxes).

Others have used the control volumes and edge-based data structure described above in conjunction with tetrahedral- (Mavriplis, 1995) and hybrid-grids (Selmin, 1993). Our numerical method differs from previous methods in the means by which artificial dissipation is introduced, as described in the next section.

1.2 Transport Equation Differencing

Co-located variable schemes, such as that in Chad, rely on either upwinding, or the introduction of explicit node coupling, to control alternate node uncoupling problems. Chad uses a combination of both streamline upwinding of the advective terms and explicit node coupling to couple pressures at neighboring nodes. The diffusion terms are differenced by means of a simple and novel scheme that also couples solution variables at neighboring nodes.

Transport equation differencing is illustrated by consideration of a generic conservation equation for mass-specific quantity Q:

$$\frac{\mathrm{d}}{\mathrm{d}t}\iiint_{v} \rho Q \,\mathrm{d}v + \iint_{s} \rho Q\,(\mathbf{u}-v)\cdot n\,\mathrm{d}a = \iint_{s} \rho D_Q \nabla Q\cdot n\,\mathrm{d}a + \iiint_{v} S_Q\,\mathrm{d}v \tag{1}$$

where ρ is the gas density, **u** the gas velocity, points on the surface S of control volume V move with velocity v, D_Q is the diffusivity, and S_Q is a source term. The Chad difference approximation to this equation is the following:

$$\frac{\rho_\nu^{n+1}\nabla_\nu^{n+1}Q_\nu^{n+1}-\rho_\nu^{n}V_\nu^{n}Q_\nu^{n}}{\Delta t}+\sum_\alpha \rho_\alpha Q_\alpha(\mathbf{u}_\alpha\cdot\mathbf{A}_\alpha-\phi_\alpha) = \sum_\alpha (\rho D_Q)_\alpha (\nabla Q)_\alpha^{n+1}\cdot\mathbf{A}_\alpha^{n+1}+(S_Q)_\nu^{n+1}V_\nu^{n+1} \tag{2}$$

where Δt is the computational timestep, the superscripts denote time levels of numerical approximation, and the face variables are evaluated by methods we now describe.

The quantities ρ_α and Q_α are differenced using the variable explicit/implicit method detailed in O'Rourke and Sahota (1998). As mentioned in the introduction, this scheme employs a time-accurate explicit scheme when the local material speed Courant number is less than one. In this case streamline upwinding is used, and advected quantities are evaluated at time level $t^n + \Delta t/2$, resulting in an advective scheme that is second-order accurate. When the local Courant number is greater than one, a variable amount of implicitness is used to maintain stability and monotonicity. In this latter case, the difference scheme retains second-order spatial accuracy, but becomes first-order accurate in time.

The quantity ϕ_α is the exact volume swept out by face α in moving from its time-n to its time-$(n + 1)$ position, divided by Δt, and is positive if volume is added to volume V_ν by such motion.

The cell-face velocity $\mathbf{u}_\alpha$ is evaluated by a modified form of the method of Rhie and Chow (1983):

$$\mathbf{u}_\alpha = \frac{\mathbf{u}_\nu^{n+1}+\mathbf{u}_{\nu\alpha}^{n+1}}{2}+\left[\psi_\alpha-\frac{\mathbf{u}_\nu^{n+1}+\mathbf{u}_{\nu\alpha}^{n+1}}{2}\cdot\mathbf{A}_\alpha^{n+1}\right]\frac{\mathbf{A}_\alpha^{n+1}}{\left|\mathbf{A}_\alpha^{n+1}\right|^2} - \frac{2\,\Delta t_\alpha}{\rho_\nu^n+\rho_{\nu\alpha}^n}\left[(\nabla p)_\alpha^{n+1}-\frac{(\nabla p)_\nu^{n+1}+(\nabla p)_{\nu\alpha}^{n+1}}{2}\right] \tag{3}$$

The quantity ψ_α is the rate at which volume would be swept out by face α if the mesh moved with the fluid velocity; thus in a Lagrangian calculation, in which $\mathbf{v}_\nu = \mathbf{u}_\nu$, in the limit of small Δt we would have $\psi_\alpha = \phi_\alpha$. The pressure gradient terms in Equation (3) are defined below. These can be interpreted as introducing a fourth-order pressure node-coupler (O'Rourke and Sahota, in preparation).

The cell-face gradient $(\Delta Q)\alpha$ is defined by averaging nodal gradients, and correcting to ensure that the component of the gradient in the direction of the associated edge, is consistent with the difference between Q_ν and $Q_{\nu\alpha}$:

$$(\nabla Q)_\alpha = \frac{(\nabla Q)_\nu+(\nabla Q)_{\nu\alpha}}{2} + \left(Q_{\nu\alpha}-Q_\nu-\frac{\nabla Q_\nu+\nabla Q_{\nu\alpha}}{2}\cdot\mathbf{d}_\alpha\right)\cdot\frac{\mathbf{d}_\alpha}{\left|\mathbf{d}_\alpha\right|^2} \tag{4}$$

where

$$\mathbf{d}_\alpha = \mathbf{x}_{\nu\alpha} - \mathbf{x}_\nu\,.$$

The nodal gradients are defined by:

$$\nabla Q_\nu = \frac{1}{V_\nu}\sum_\alpha \frac{Q_{\nu\alpha}-Q_\nu}{2}\mathbf{A}_\alpha \tag{5}$$

This cell-face gradient approximation differs from the normal finite-element approximation, but it gives equally accurate results in test calculations we have performed and can be used with an edge-based data structure and hexahedral elements.

The system of coupled, nonlinear finite-difference equations that results from applying Equation (2) to the gas-phase equations, is solved by an adaptation of the Simple method, with individual equations solved by GMRES.

2 EXAMPLE CALCULATION: COLD FLOW IN A 4-VALVE DIESEL ENGINE

Steady flow is being simulated in an engine in which flow measurements are being performed at *Sandia National Laboratories* and at *Ricardo Inc*. The engine and its geometry are illustrated in Figure 3. The engine is of the four-valve-per-cylinder type, with a swirl type port provided by *Ricardo*. Flow through the port and cylinder assembly is driven by a specified pressure drop. The Chad calculations are performed with a mesh of 200 000 hexahedral elements provided by the University of Wisconsin. The calculations performed to date have used the k/ε turbulence model, although we plan to use sub-grid scale turbulence models in future calculations.

The computational results show the formation of three jets resulting from the interaction of flows through the two inflow ports. These are illustrated in the velocity magnitude plot of Figure 4. Rotating jets B and C result from flows that enter from each of the two ports without interacting with flow from the other port. Jet A forms along the center-plane of the engine and results from the interaction of the flows through the two ports. Because the flows are counter-rotating when they meet on the center-plane, there is swirl cancellation and very little rotation associated with jet A.

The calculations are performed on an *SGI Origin 2000* computer, and Figure 5 shows parallel speed-up results for runs on multi-processors of this machine. It is seen that parallel speed-up is super-linear up to four processors, and scales linearly with processor number above four processors.

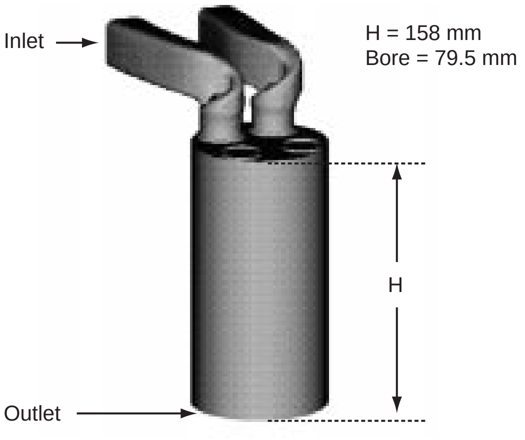

Figure 3

Illustration of the *Sandia* (*Ricardo*) engine.

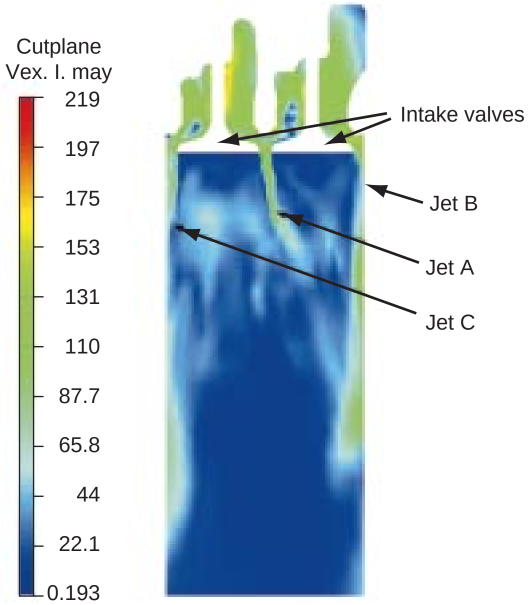

Figure 4

Velocity magnitude plot in a plane through the two intake valves.

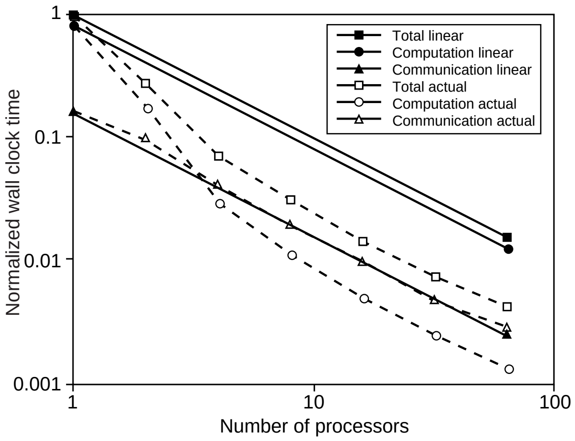

Figure 5

Parallel speed-up for 4-valve engine calculation.

ACKNOWLEDGMENTS

Support for this work was provided by *United States Department of Energy's Office of Advanced Automotive Technologies and Office of Heavy Vehicle Technologies*. We thank Margaret Hubbard for providing Figures 3 and 4. We also wish to thank Dr. Keith Richards of the University of Wisconsin for supplying the mesh for the four-valve engine calculations.

REFERENCES

Mavriplis, D.J. (1995) Three-Dimensional Multigrid Reynolds-Averaged Navier-Stokes Solver for Unstructured Meshes. *AIAA J,* **33**, 445.

O'Rourke, P.J. and Sahota, M.S. (1998) A Variable Explicit/Implicit Numerical Method for Calculating Advection on Unstructured Meshes. *J. Comput. Phys.,* **143**, 312.

O'Rourke, P.J. and Sahota, M.S. *No-Utopia: A Parallel, Implicit Method for Calculating Three-Dimensional Fluid Flows on Unstructured Meshes*, in preparation.

Selmin, V. (1993) The Node-Centered Finite Volume Approach: Bridge between Finite Differences and Finite Elements. *Comput. Methods in Appl. Mech. Engrg.,* **102**, 107.

Rhie, C.M. and Chow, W.L. (1983) *AIAA J.,* **21**, 1525.

Final manuscript received in March 1999