SciDAC 2009 IOP Publishing
Journal of Physics: Conference Series **180** (2009) 012010 doi:10.1088/1742-6596/180/1/012010

High-order finite-volume adaptive methods on locally rectangular grids

P Colella[1], M Dorr[2], J Hittinger[2], D F Martin[1], P McCorquodale[1]

[1] Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road Mail Stop 50A1148, Berkeley, CA 94720-8142

[2] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Box 808, L-561, Livermore, CA 94551-0808

E-mail: PWMcCorquodale@lbl.gov

Abstract. We are developing a new class of finite-volume methods on locally-refined and mapped grids, which are at least fourth-order accurate in regions where the solution is smooth. This paper discusses the implementation of such methods for time-dependent problems on both Cartesian and mapped grids with adaptive mesh refinement. We show 2D results with the Berger–Colella shock-ramp problem in Cartesian coordinates, and fourth-order accuracy of the solution of a Gaussian pulse problem in a polytropic gas in mapped coordinates.

1. High-order finite-volume methods on Cartesian grids

In the finite-volume approach, the spatial domain in $\mathbb{R}^D$ is discretized as a union of control volumes that covers the domain. With Cartesian grids, a control volume $V_{\boldsymbol{i}}$ takes the form

$$V_{\boldsymbol{i}} = [\boldsymbol{i}h, (\boldsymbol{i}+\boldsymbol{u})h], \boldsymbol{i} \in \mathbb{Z}^D, \boldsymbol{u} = (1, 1, \dots, 1),$$

where h is the grid spacing.

A finite-volume discretization of a partial differential equation is based on averaging that equation over control volumes, applying the divergence theorem to replace volume integrals by integrals over the boundary of the control volume, and approximating the boundary integrals by quadratures. For example, for time-dependent problems of the form

$$\frac{\partial U}{\partial t} + \nabla \cdot \vec{F} = 0 \tag{1}$$

the discretized solution in space is the average of U over a control volume:

$$\langle U \rangle_{\boldsymbol{i}}(t) = \frac{1}{h^D} \int_{V_{\boldsymbol{i}}} U(\boldsymbol{x}, t) d\boldsymbol{x}. \tag{2}$$

In that case, we can compute the evolution of the spatially discretized system by a method-of-lines approach:

$$\frac{d\langle U \rangle_{\boldsymbol{i}}}{dt} = -\frac{1}{h^D} \int_{V_{\boldsymbol{i}}} \nabla \cdot \vec{F} d\boldsymbol{x} = -\frac{1}{h} \sum_{d=1}^{D} (\langle F^d \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle F^d \rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}) \tag{3}$$

$$\langle F^d \rangle_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{h^{D-1}} \int_{A_d^{\pm}} F^d dA, \tag{4}$$

where $A_d^{\pm}$ are the high and low faces bounding $V_{\boldsymbol{i}}$ with normals pointing in the direction of unit vector $\boldsymbol{e}^d$. In the right-hand side of (4), the integrals over faces are approximated using some quadrature rule. Conserved quantities in the original PDE then satisfy an analogous conservation law in the discretized system.

Most finite-volume methods use the midpoint rule to approximate the flux integrals in (4) leading to a second-order accurate method. We are developing higher-order methods (fourth order or better) derived from replacing the integrand on the right-hand side of (4) with a Taylor expansion about the center of the face. This paper will be concerned with fourth-order methods, in which

$$\langle F^d \rangle = F^d(\boldsymbol{x}_0) + \frac{h^2}{24} \sum_{d' \neq d} \frac{\partial^2 F^d}{\partial x_{d'}^2} + O(h^4). \tag{5}$$

In computing fluxes on faces, we use face-averaged variables, which can be obtained to fourth-order accuracy with the standard deconvolution algorithm [6]:

$$\langle w \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{7}{12}(\langle w \rangle_{\boldsymbol{i}} + \langle w \rangle_{\boldsymbol{i}+\boldsymbol{e}^d}) - \frac{1}{12}(\langle w \rangle_{\boldsymbol{i}-\boldsymbol{e}^d} + \langle w \rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}) \tag{6}$$

For hyperbolic problems, it is also necessary to introduce limiters to suppress oscillations in the presence of discontinuities and underresolved gradients. To do this, we apply an extension of the PPM limiter [6, 5] to values obtained from the deconvolution formula (6). In this approach, we view the PPM limiter as producing limited, and possibly doubly-valued face averages of the variable w. In the latter case, we solve a Riemann problem at the face to compute the flux. The use of the limiter [5] leads to an algorithm that preserves fourth-order accuracy in the presence of smooth extrema. If, in addition, there is a constraint that w remain non-negative, we can use a version of the FCT limiter in [7] to impose that constraint, with the low-order method given by the CTU scheme in [3]. Since we do not use the FCT limiter to impose any bounds other than positivity, the method remains fourth-order accurate for smooth data.

For the case of the compressible Euler equations, we also include an artificial viscosity, replacing the flux $\vec{F}$ by $\vec{F} - \nu \nabla U$, where

$$\nu = K_1 h^2 \min\{h^2 \frac{|\nabla \cdot \vec{u}|^2}{K_2 c^2}, 1\} |\nabla \cdot \vec{u}| \tag{7}$$

in which $\vec{u}$ is the velocity, c is the speed of sound, and K_1 and K_2 are constants. Then ν is cubic in $h|\nabla \cdot \vec{u}|$ when $h|\nabla \cdot \vec{u}|$ is small, and linear in $h|\nabla \cdot \vec{u}|$ when $h|\nabla \cdot \vec{u}|$ is large, relative to the speed of sound. In the latter case, (7) reduces to the artificial viscosity in the second-order method of [6]. For smooth flows, (7) leads to a fourth-order accurate method.

The equation (3) is of the form

$$\frac{dU}{dt} = f(t, U) \tag{8}$$

and we solve this with a standard fourth-order Runge–Kutta method to integrate from $U(t^n)$ to $U(t^n + \Delta t)$, using:

$$f_0 = f(t^n, U(t^n)); \quad f_1 = f(t^n + \frac{\Delta t}{2}, U(t^n) + f_0 \frac{\Delta t}{2}); \tag{9}$$

$$f_2 = f(t^n + \frac{\Delta t}{2}, U(t^n) + f_1 \frac{\Delta t}{2}); \quad f_3 = f(t^n + \Delta t, U(t^n) + f_2 \Delta t); \tag{10}$$

$$U(t^n + \Delta t) = U(t^n) + \frac{\Delta t}{6}(f_0 + 2f_1 + 2f_2 + f_3) + O((\Delta t)^5). \tag{11}$$

2. Multilevel algorithm

In a multilevel algorithm, for each pair of successive levels, we shall need to interpolate from data on coarser-level patches at the beginning and end of each coarse timestep, to ghost cells of finer-level patches at intermediate times. After advancing the coarse solution one timestep, from time t^n to time $t^n + \Delta t_c$, we implement this interpolation in the following stages:

(i) Copy data: Copy $U(t^n)$ and $U(t^n + \Delta t_c)$ from the original coarser-level patches to a new set of patches made from coarsening the finer-level patches by the refinement ratio n_{ref} and then expanding each patch by three layers of ghost cells.

(ii) Interpolate in time: Using the intermediate results f_0, f_1, f_2, f_3 from (9)–(10) in the Runge–Kutta method as applied on the coarser level, interpolate from data at times t^n and $t^n + \Delta t_c$ on the expanded coarsened fine patches to times $t^n + k\Delta t_c/n_{ref}$ for all $k = 1, 2, ..., n_{ref} - 1$, on the same patches. Do this using a Taylor series about the midpoint in time:

$$\begin{aligned} U(t^n + \alpha\Delta t) =& U(t^n) + \frac{\Delta t}{24}(5f_0 + 4f_1 + 4f_2 - f_3) + (\frac{f_1 + f_2}{2})((\alpha - \frac{1}{2})\Delta t) \\ &+ (\frac{f_3 - f_0}{2\Delta t})((\alpha - \frac{1}{2})\Delta t)^2 + (\frac{f_0 - f_1 - f_2 + f_3}{6(\Delta t/2)^2})((\alpha - \frac{1}{2})\Delta t)^3 + O((\Delta t)^4) \end{aligned} \tag{12}$$

for $0 \leq \alpha \leq 1$.

(iii) Interpolate in space: At all of the intermediate times, interpolate from data on the expanded coarsened finer-level patches to ghost cells of the original finer-level patches. For coarse cell $\boldsymbol{i}$, solve a constrained linear least-squares problem for $a_{\boldsymbol{i},\boldsymbol{p}}$ in the overdetermined system

$$\sum_{p_d \geq 0, \|\boldsymbol{p}\|_1 \leq 3} a_{\boldsymbol{i},\boldsymbol{p}} \langle (\boldsymbol{x} - \boldsymbol{x_i})^{\boldsymbol{p}} \rangle^c_{\boldsymbol{j}} = \langle u \rangle^c_{\boldsymbol{j}} \quad \text{for all } \boldsymbol{j} \in N(\boldsymbol{i}) - \{\boldsymbol{i}\} \tag{13}$$

where $N(\boldsymbol{i})$ is a set of neighboring cells to $\boldsymbol{i}$, and the conservation constraint is

$$\sum_{p_d \geq 0, \|\boldsymbol{p}\|_1 \leq 3} a_{\boldsymbol{i},\boldsymbol{p}} \langle (\boldsymbol{x} - \boldsymbol{x_i})^{\boldsymbol{p}} \rangle^c_{\boldsymbol{i}} = \langle u \rangle^c_{\boldsymbol{i}} \tag{14}$$

where $\boldsymbol{x_i}$ is the center of cell $\boldsymbol{i}$. Then for each fine cell $\boldsymbol{k}$ within coarse cell $\boldsymbol{i}$:

$$\langle u \rangle^f_{\boldsymbol{k}} = \sum_{p_d \geq 0, \|\boldsymbol{p}\|_1 \leq 3} a_{\boldsymbol{i},\boldsymbol{p}} \langle (\boldsymbol{x} - \boldsymbol{x_i})^{\boldsymbol{p}} \rangle^f_{\boldsymbol{k}}. \tag{15}$$

Aside from the details outlined in this section, the algorithm is the same as in [2]. In particular, conservation at refinement boundaries is exactly the same.

3. Finite-volume methods on mapped grids

We can extend this approach to the case of mapped grids. We assume that we have a smooth mapping $\boldsymbol{X}$, independent of time, from some abstract coordinate space into physical space:

$$\boldsymbol{X} = \boldsymbol{X}(\boldsymbol{\xi}) \ , \ \boldsymbol{X} : [0, 1]^D \rightarrow \mathbb{R}^D.$$

Given this mapping, the divergence of a vector field in physical space can be written in terms of derivatives in the mapping space, that is

$$\nabla_{\boldsymbol{x}} \cdot \vec{F} = \frac{1}{J} \nabla_{\boldsymbol{\xi}} \cdot (\boldsymbol{N}^T \vec{F}), \tag{16}$$

$$J = det(\nabla_{\boldsymbol{\xi}} \boldsymbol{X}) \ , \ \boldsymbol{N}^T_{p,q} = det((\nabla_{\boldsymbol{\xi}} \boldsymbol{X})(p|\boldsymbol{e}^q)), \tag{17}$$

where $\boldsymbol{A}(p|\boldsymbol{v})$ denotes the matrix obtained by replacing the p^{th} row of the matrix $\boldsymbol{A}$ by the vector $\boldsymbol{v}$. The relationship (16) is a consequence of the chain rule, equality of mixed partials, and Cramer's rule. We can also rewrite (1) in terms of the independent variable $\boldsymbol{\xi}$:

$$\frac{\partial(JU)}{\partial t} + \nabla_{\boldsymbol{\xi}} \cdot (\boldsymbol{N}^T \vec{F}) = 0. \tag{18}$$

The semi-discrete form of (18) corresponding to (3)–(4) is then

$$\frac{d}{dt}\langle JU\rangle_{\boldsymbol{i}} = -\frac{1}{h^D}\int\limits_{V_{\boldsymbol{i}}} \nabla \cdot (\boldsymbol{N}^T\vec{F})d\boldsymbol{\xi} = -\frac{1}{h}\sum_{d=1}^{D}(\langle(\boldsymbol{N}^T\vec{F})_d\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle(\boldsymbol{N}^T\vec{F})_d\rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}) \tag{19}$$

$$\langle(\boldsymbol{N}^T\vec{F})_d\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{h^{D-1}}\int\limits_{A_d}(\boldsymbol{N}^T\vec{F})_d dA. \tag{20}$$

For a single-level problem, we solve using a generalization of the advection method in [4] to nonlinear hyperbolic systems. With mapped grids and multiple levels of refinement, we can use a straightforward extension of the approach described for the Cartesian case in section 2.

4. Results

Figure 1 shows a color plot and contour lines for density with the shock-ramp problem of Berger and Colella [2], implemented in Cartesian coordinates with two levels of refinement, by a factor of 4, with an effective resolution of 1024 × 256. We use artificial viscosity coefficients of $K_1 = K_2 = 0.3$ in (7).

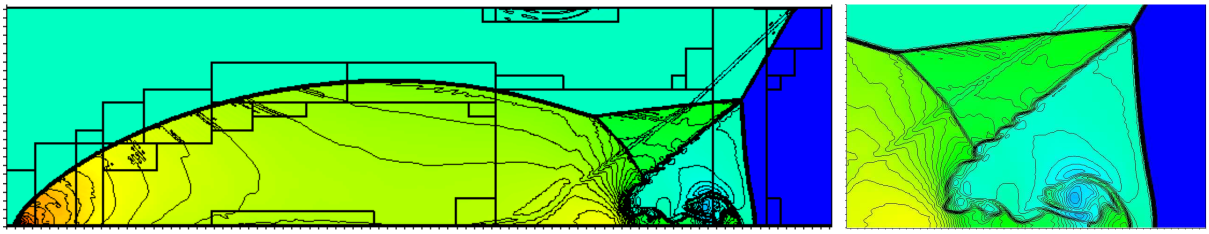

Figure 1. Density and contour plot for ramp problem on two levels in Cartesian coordinates. Outlines of grid boxes are shown with thick black lines. The right-hand plot is a close-up of the left-hand plot showing the double Mach reflection region. This calculation demonstrates the ability of the limiters and the artificial viscosity to produce a robust treatment of problems with strong shocks.

As an example using mapped coordinates, Figure 2 shows a density plot of a Gaussian acoustic pulse in a 2D polytropic gas in a periodic domain, implemented with a mapped coordinate system and two levels of refinement, by a factor of 4. The coordinate mapping is

$$\boldsymbol{X}(\xi,\eta) = (\xi + \sin(2\pi\xi)\sin(2\pi\eta)/10, \eta + \sin(2\pi\xi)\sin(2\pi\eta)/10). \tag{21}$$

There is no artificial viscosity. Table 1 shows fourth-order convergence of the component $\langle J\rho\rangle$, where ρ is density.

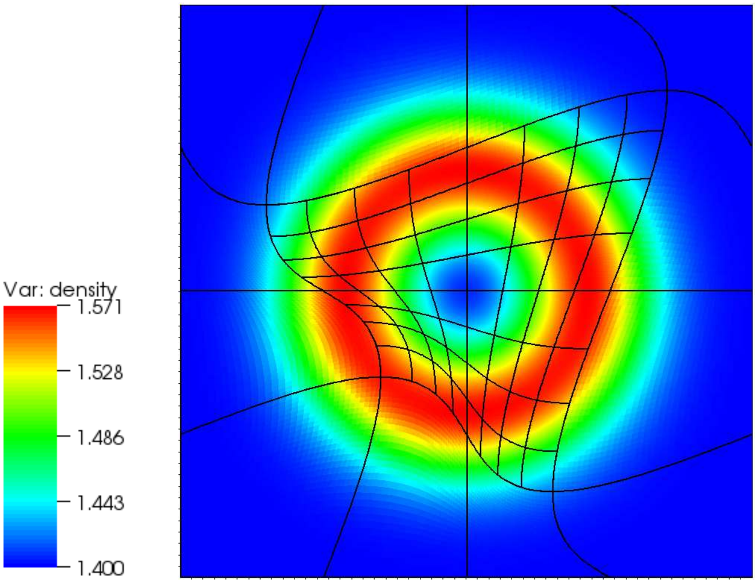

Figure 2. Plot of density with a Gaussian acoustic pulse in a polytropic gas in a periodic domain, using two levels of refinement with the mapped coordinates of (21), after 240 coarse-level time steps each of length 0.0006. At the initial time, the density profile is a circularly symmetric Gaussian with its peak at the center of the domain. Each grid box consists of 32×32 cells; the outlines of these boxes are shown in black.

Table 1. Convergence of $\langle J\rho\rangle$ in two-level Gaussian acoustic pulse example with mapped coordinates of (21). The simulation was run with different mesh spacings, and the difference between simulations with coarse-level mesh spacing (in abstract space) h and $h/2$ was taken at time 0.144. This table shows the norms of those differences in $\langle J\rho\rangle$, and the rates of convergence.

norm	1/32 to 1/64	rate	1/64 to 1/128	rate	1/128 to 1/256	rate	1/256 to 1/512
L_1	1.3072e-04	3.82	9.2338e-06	3.93	6.0686e-07	3.97	3.8791e-08
L_2	2.4617e-04	3.73	1.8504e-05	3.91	1.2332e-06	3.97	7.8870e-08
L_∞	1.5054e-03	3.44	1.3899e-04	3.84	9.6921e-06	3.92	6.3868e-07

References

[1] Barad M and Colella P 2006 *J. Comput. Phys.* **208** 1
[2] Berger M J and Colella P 1989 *J. Comput. Phys.* **82** 64
[3] Colella P 1990 *J. Comput. Phys.* **87** 171
[4] Colella P, Dorr M, Hittinger J and Martin D F 2009 submitted to *J. Comput. Phys.*
[5] Colella P and Sekora M 2008 *J. Comput. Phys.* **227** 7069
[6] Colella P and Woodward P R 1984 *J. Comput. Phys.* **54** 174
[7] Zalesak S T 1979 *J. Comput. Phys.* **35** 335