Compressible inviscid fluid flow solver using the MUSCLE-Hancock method and a HLLC Riemann solver.

Jan I.C. Vermaak[1,2], Jim E. Morel[1,2]

[1]Center for Large Scale Scientific Simulations, Texas A&M Engineering Experiment Station, College Station, Texas, USA.
[2]Nuclear Engineering Department, Texas A&M University, College Station, Texas, USA.

Abstract:
Work is work for some, but for some it is play.

Keywords: hydrodynamics

1 Introduction

For this research we develop a fluid flow solver for the solution of flow problems involving compressible inviscid ideal gases. The governing equations are the Euler equations defined as

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{u}) = 0 \tag{1.1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \boldsymbol{\nabla} \cdot \{\rho \mathbf{u} \otimes \mathbf{u}\} + \boldsymbol{\nabla} P = \mathbf{f} \tag{1.2}$$

$$\frac{\partial E}{\partial t} + \boldsymbol{\nabla} \cdot [(E+p)\mathbf{u}] = q, \tag{1.3}$$

where ρ is the fluid density, $\mathbf{u} = [u_x, u_y, u_z] = [u, v, w]$ is the fluid velocity in cartesian coordinates, P is the fluid pressure, $\mathbf{f} = [f_x, f_y, f_z]$ is an arbitrary momentum-density source or sink, E is the material energy-density comprising kinetic energy-density, $\frac{1}{2}\rho||\mathbf{u}||^2$, and internal energy-density, ρe, such that $E = \frac{1}{2}\rho||\mathbf{u}||^2 + \rho e$, where e is the specific internal energy. The value q is an arbitrary energy-density source or sink.

The ideal gas law provides the closure relation

$$p = (\gamma - 1)\rho e \tag{1.4}$$

where γ is the ratio of the constant-pressure specific heat, c_p, to the constant-volume specific heat, c_v, i.e., $\gamma = \frac{c_p}{c_v}$, and is a material property.

1.1 Notation in preparation for numerical schemes

The conservation of mass-, momentum-, and energy equations can be written in the following form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) + \frac{\partial}{\partial y}\mathbf{G}(\mathbf{U}) + \frac{\partial}{\partial z}\mathbf{H}(\mathbf{U}) = \mathbf{Q} \tag{1.5}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u u + p \\ \rho u v \\ \rho u w \\ u(E+p) \end{bmatrix}, \mathbf{G}(\mathbf{U}) = \begin{bmatrix} \rho v \\ \rho v u \\ \rho v v + p \\ \rho v w \\ v(E+p) \end{bmatrix}, \mathbf{H}(\mathbf{U}) = \begin{bmatrix} \rho w \\ \rho w u \\ \rho w v \\ \rho w w + p \\ w(E+p) \end{bmatrix}, \text{ and } \mathbf{Q} = \begin{bmatrix} 0 \\ f_x \\ f_y \\ f_z \\ q \end{bmatrix}. \tag{1.6}$$

The $\mathbf{U}$ vector is now a collection of the conserved variables, the $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ vectors is representative of generic flux terms, and the $\mathbf{Q}$ vector is a generic source term. We will be using this notation in the sections that follow. 4

1.2 General Finite Volume discretization

Consider the cell volume, in 3D, shown in Figure 1 below.

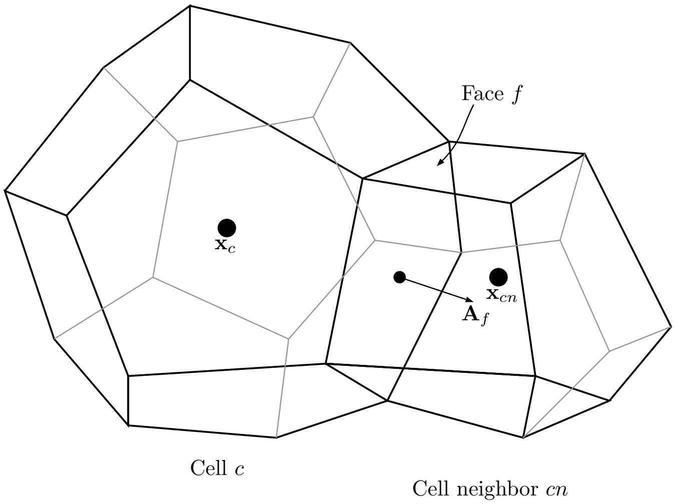

Figure 1: Schematic of a multidimensional cell.

Integrating Eq. (1.5) over a finite volume,

$$\int_V \left(\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{\nabla} \cdot \mathcal{F}(\mathbf{U}) \right) dV = \int_V \mathbf{Q} dV \tag{1.7}$$

where $\mathcal{F}(\mathbf{U}) = \big[\mathbf{F}(\mathbf{U}), \mathbf{G}(\mathbf{U}), \mathbf{H}(\mathbf{U})\big]$. Next, using Gauss's divergence theorem, allows us to write

$$\int_V \frac{\partial \mathbf{U}}{\partial t} dV + \int_S \mathbf{n} \cdot \mathcal{F} dA = \int_V \mathbf{Q} dV. \tag{1.8}$$

Now, using cell c as the volume of integration, and assuming $\mathbf{U}$ and $\mathbf{Q}$ constant over the cell, with values $\mathbf{U}_c$ and $\mathbf{Q}_c$, the above equation becomes

$$V_c \frac{\partial \mathbf{U}_c}{\partial t} + \sum_{f=0}^{N_{f,c}-1} \mathbf{A}_f \cdot \mathcal{F}_f = V_c \mathbf{Q}_c. \tag{1.9}$$

where V_c is the volume of the cell, $N_{f,c}$ is the number of faces for cell c, $\mathbf{A}_f$ is the area-vector of face f, which is the product of the face area, A_f, and the face normal, $\mathbf{n}_f$, and $\mathcal{F}_f = \mathcal{F}(\mathbf{U}_f)$ is the face flux vector. The treatment of the $\mathcal{F}_f$ term is the topic of the MUSCL-Hancock method which we detail in section 2.

1.3 Multidimensional transformation of interface fluxes

Most of the Riemann solver schemes presented in [1] are for one dimensional geometries. A transformation technique is prescribed in [1] that allows one to seamlessly use the one dimensional formulations, and is as follows.

Given the arbitrary face area vector, $\mathbf{A}_f = \mathbf{n}_f A_f$, we first seek a rotation matrix, $R_{\hat{\mathbf{i}}}$, to rotate any vector about an axis $\mathbf{a}_{\hat{\mathbf{i}}}$ such that it is aligned with $\hat{\mathbf{i}}$. To determine $R_{\hat{\mathbf{i}}}$ we first set the rotation axis as

$$\mathbf{a}_{\hat{\mathbf{i}}} = \frac{\mathbf{n}_f \times \hat{\mathbf{i}}}{||\mathbf{n}_f \times \hat{\mathbf{i}}||} \tag{1.10}$$

and the angle of rotation, $\theta_{\hat{\mathbf{i}}}$, as

$$\theta_{\hat{\mathbf{i}}} = \arccos(\mathbf{n}_f \cdot \hat{\mathbf{i}}). \tag{1.11}$$

We then apply Rodrigues's formula as detailed in appendix A to obtain $R_{\hat{\mathbf{i}}}$. With this matrix in hand we can form the general transformation matrix, $T_{\hat{\mathbf{i}}}$, defined as

$$T_{\hat{\mathbf{i}}} = \mathrm{diag}(1, R, 1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & R_{00} & R_{01} & R_{02} & 0 \\ 0 & R_{10} & R_{11} & R_{12} & 0 \\ 0 & R_{20} & R_{21} & R_{22} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.12}$$

which can be used to define

$$\hat{\mathbf{F}}_f = \mathbf{F}(\hat{\mathbf{U}}_f) \tag{1.13}$$

where

$$\hat{\mathbf{U}}_f = T_{\hat{\mathbf{i}}}\mathbf{U}_f \tag{1.14}$$

such that

$$\mathbf{n}_f \cdot \mathcal{F}_f = \mathbf{n} \cdot [\mathbf{F}(\mathbf{U}_f), \mathbf{G}(\mathbf{U}_f), \mathbf{H}(\mathbf{U}_f)] = T_{\hat{\mathbf{i}}}^{-1}\hat{\mathbf{F}}_f \tag{1.15}$$

allowing us to write Eq. (1.9) as

$$\frac{\partial \mathbf{U}_c}{\partial t} + \frac{1}{V_c}\sum_{f=0}^{N_{f,c}-1} A_f T_{\hat{\mathbf{i}}}^{-1}\hat{\mathbf{F}}_f = \mathbf{Q}_c. \tag{1.16}$$

In this form any Riemann solver can simply be supplied with $\hat{\mathbf{U}}$ for the cells on either side of an interface in order to use the classical one dimensional formulations contained in [1]. The given Riemann solver will then produce the appropriate value for $\hat{\mathbf{F}}_f$.

2 MUSCL-Hancock Method (MHM) and the HLLC Riemann-solver

The MUSCL-Hancock method is conceptually simple. It prescribes how the interface fluxes are to be computed at the beginning of a time step and then how to supply the necessary inputs to a Riemann-solver for the computation of the preserved variables at the end of the timestep.

2.1 The MUSCL-Hancock Method

MUSCL stands for Monotone Upstream-centered Scheme for Conservation Laws. The scheme as modified by S. Hancock gives the method its name. We will refer to this method by simply using the abbreviation *MHM*.

The MHM has three basic steps.

2.1.1 Step 1 - Extrapolate $\mathbf{U}$ using an estimate of the gradient $(\boldsymbol{\nabla}\mathbf{U})$ and compute the interface fluxes at time step n

In general notation the extrapolation of the conserved variable for cell c, $\mathbf{U}_c$, to a cell face f, $\mathbf{U}_{f,c}$, can be computed from

$$\mathbf{U}_{f,c}^n \approx \mathbf{U}_c^n + (\mathbf{x}_f - \mathbf{x}_c) \cdot \left\{\boldsymbol{\nabla}\mathbf{U}\right\}_c^n, \tag{2.1}$$

where $\mathbf{U}_f$ is the extrapolated value of $\mathbf{U}$ at the face center, $\mathbf{U}_c$ is the finite volume cell-constant value of $\mathbf{U}$ for cell c, $\mathbf{x}_f$ is the face centroid, $\mathbf{x}_c$ is the cell centroid, and $\left\{\boldsymbol{\nabla}\mathbf{U}\right\}_c$ is the gradient of $\mathbf{U}$ at the center of cell c.

Provided that an orthogonal mush is used, the gradient can be estimated in each cell c from

$$\left\{\boldsymbol{\nabla}\mathbf{U}\right\}_c \approx \frac{1}{V_c}\sum_{f=0}^{N_{f,c}-1}\left\{\mathbf{A}_f\otimes\left(\frac{||\mathbf{x}_f-\mathbf{x}_c||}{||\mathbf{x}_{cn}-\mathbf{x}_c||}\mathbf{U}_c+\frac{||\mathbf{x}_c-\mathbf{x}_f||}{||\mathbf{x}_{cn}-\mathbf{x}_c||}\mathbf{U}_{cn}\right)\right\}, \tag{2.2}$$

where V_c is the volume of cell c, $N_{f,c}$ is the number of faces for cell c, A_f the face area-vector of face f (i.e., $\mathbf{A}_f = A_f\mathbf{n}_f$), $\mathbf{x}_{cn}$ is the centroid of the neighboring cell cn at face f, and finally $\mathbf{U}_{cn}$ is the finite volume cell-constant value of $\mathbf{U}$ for cell cn.

Check slope limiters.

2.1.2 Step 2 - Advance the conserved variables by half a time step

Using the extrapolated values $\mathbf{U}_{f,c}^n$, compute associated transformed interface fluxes, $\hat{\mathbf{F}}_f^n = \mathbf{F}(T_{\hat{\mathbf{i}}}\mathbf{U}_{f,c}^n)$, after which

$$\mathbf{U}_c^{n+\frac{1}{2}} = \mathbf{U}_c^n - \frac{\frac{1}{2}\Delta t^n}{V_c}\sum_{f=0}^{N_{f,c}-1}\left(A_f T_{\hat{\mathbf{i}}}^{-1}\hat{\mathbf{F}}_f^n\right) + \frac{1}{2}\Delta t^n\mathbf{Q}. \tag{2.3}$$

These half-time step values are then passed to a suitable Riemann solver.

2.1.3 Step 3 - Execute a Riemann solver

The Riemann solver computes the interface fluxes, $\hat{\mathbf{F}}_f^{\mathcal{R},n+\frac{1}{2}}$, using $\mathbf{U}_c^{n+\frac{1}{2}}$, where $\mathcal{R}$ denotes the specific Riemann solver. The discontinuity across a face is treated as a one dimensional problem with a left and right side having different values for $\mathbf{U}$, i.e., $\mathbf{U}_L = \hat{\mathbf{U}}_c^{n+\frac{1}{2}}$ and $\hat{\mathbf{U}}_R = \mathbf{U}_{cn}^{n+\frac{1}{2}}$ for the left and right side respectively.

When using the HLLC solver, as described in [1], the Riemann solver will compute $\hat{\mathbf{F}}_f^{hllc,n+\frac{1}{2}}$ after which we compute the conserved variables at $n+1$ from

$$\mathbf{U}_c^{n+1} = \mathbf{U}_c^n - \frac{\Delta t^n}{V_c}\sum_{f=0}^{N_{f,c}-1}\left(A_f T_{\hat{\mathbf{i}}}^{-1}\hat{\mathbf{F}}_f^{hllc,n+\frac{1}{2}}\right) + \Delta t^n\mathbf{Q}. \tag{2.4}$$

The HLLC Riemann solver is detailed in section 2.2.

2.2 The HLLC Approximate Riemann Solver in one dimension

The Harten Lax and van Leer (HLL) solver scheme was developed in 1983 [1] and requires estimates for the fastest wave/signal/shock velocities emerging from a discontinuity. Later Toro, Spruce and Speares proposed the Harten Lax van Leer *Contact* (HLLC) scheme [1] which adds another wave to the problem.

The primary inputs for this solver are

$$\begin{aligned}\mathbf{U}_L &= \hat{\mathbf{U}}_c^{n+\frac{1}{2}}\\ \mathbf{U}_R &= \hat{\mathbf{U}}_{cn}^{n+\frac{1}{2}}\\ \mathbf{F}_L &= \mathbf{F}(\hat{\mathbf{U}}_c^{n+\frac{1}{2}})\\ \mathbf{F}_R &= \mathbf{F}(\hat{\mathbf{U}}_{cn}^{n+\frac{1}{2}})\end{aligned} \tag{2.5}$$

2.2.1 Fastest wave speed estimation

The HLLC Riemann solver is predicated on knowing an estimate for the maximum wave speed, S_{max}, which can be computed from

$$S_{max} = \max(|u_L| + a_L, |u_R| + a_R), \tag{2.6}$$

where a_L and a_R are the sound speeds associated with the left- and right conserved variables as

$$a = \sqrt{\frac{\gamma p}{\rho}}. \tag{2.7}$$

Once a value for S_{max} has been determined, we can set

$$S_L = -S_{max} \tag{2.8}$$

and

$$S_R = S_{max}. \tag{2.9}$$

2.2.2 Contact wave speed

The contact wave speed, S_*, is given by

$$S_* = \frac{p_R - p_L + \rho_L u_L(S_L - u_L) - \rho_R u_R(S_R - u_R)}{\rho_L(S_L - u_L) - \rho_R(S_R - u_R)}. \tag{2.10}$$

2.2.3 Intermediate fluxes

As per [1] the intermediate fluxes, $\mathbf{F}_{*L}$ and $\mathbf{F}_{*R}$ are given by

$$F_{*K} = \frac{S_*(S_K\mathbf{U}_K - \mathbf{F}_K) + S_K(p_K + \rho_L(S_K - u_K)(S_* - u_K))D_*}{S_K - S_*} \tag{2.11}$$

for $K = L$ and $K = R$. The vector $\mathbf{D}_*$ is a vector such that

$$\mathbf{F}(\mathbf{U}) = u\mathbf{U} + p\mathbf{D}, \tag{2.12}$$

therefore

$$\mathbf{D}_* = [0, 1, 0, 0, S_*]^T \tag{2.13}$$

2.3 Interface flux

The interface flux $\hat{\mathbf{F}}_f^{hllc}$ is now given by

$$\hat{\mathbf{F}}_f^{hllc} = \begin{cases}\mathbf{F}_L & \text{, if } S_L \geq 0,\\ \mathbf{F}_{*L} & \text{, if } S_L \leq 0 \leq S_*,\\ \mathbf{F}_{*R} & \text{, if } S_* \leq 0 \leq S_R,\\ \mathbf{F}_R & \text{, if } S_R \leq 0\end{cases} \tag{2.14}$$

References

[1] Toro E.F., *Riemann Solvers and Numerical Methods for Fluid Dynamics - A Practical Introduction*, third edition, Springer, 2009.

A Roderigues's formula

Roderigues' formula for the rotation of a vector $\mathbf{v}$ about a unit vector $\mathbf{a}$ with right-hand rule

$$\mathbf{v}_{rotated} = \cos\theta\mathbf{v} + (\mathbf{a}\cdot\mathbf{v})(1-\cos\theta)\mathbf{a} + \sin\theta(\mathbf{a}\times\mathbf{v}) \tag{A.1}$$

In matrix form

$$\mathbf{v}_{rotated} = A\mathbf{v} \tag{A.2}$$

where

$$A = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{A.3}$$

and

$$R = I + \sin\theta A + (1-\cos\theta)A^2 \tag{A.4}$$