Modern Numerical Methods for Fluid Flow

Phillip Colella
Department of Mechanical Engineering
University of California, Berkeley

and

Elbridge Gerry Puckett
Department of Mathematics
University of California, Davis

Please feel free to make copies of this draft for your students and colleagues. Your are also welcome to use it as course material. We would appreciate any comments, suggestions and errata that you may have. These should be sent to the second author at

egpuckett@ucdavis.edu

I.Compressible Flow

II.Incompressible Flow

I. Compressible Flow

1. Finite difference methods for linear scalar problems

We would like to numerically model the behavior of a compressible fluid. We begin with a study of the one-dimensional scalar advection equation. We seek a solution $u(x, t)$ of the following partial differential equation:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{1}$$

where a is a positive constant. The initial conditions are given as:

$$u(x, 0) = \psi(x) \tag{2}$$

The solution to this problem is known to be:

$$u(x, t) = \psi(x - at) \tag{3}$$

It is useful to start with a problem for which the solution is known in order to test the accuracy of our numerical methods. In the following section we discuss finite-difference methods for solving this PDE.

We will assume that the mesh spacing, Δx, and the time step, Δt, are constant. This means that the continuous value of the solution at some location and time, $u(x, t)$, can be expressed as $u(j\Delta x, n\Delta t)$. We will represent the analogous discretized solution at the mesh cell j and time level n as u_j^n. We represent this graphically in x-t space as

:

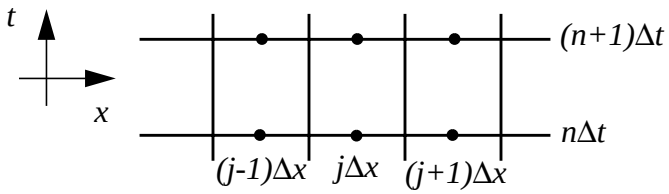


Figure 1. Discretization in space and time

The solution u_j^n is defined at the nodes which are placed at the cell centers in physical space. We discretize the partial differential equation by replacing the derivatives in equation (1) with one-sided finite difference approximations:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + O(\Delta t) + a\frac{u_j^n - u_{j-1}^n}{\Delta x} + O(\Delta x) = 0$$

This numerical scheme uses one-sided forward differencing in time and upwind differencing in space. Dropping the error terms and rewriting this equation, we obtain the discrete evolution equation for u_j^n:

$$u_j^{n+1} = u_j^n + \frac{a\Delta t}{\Delta x}(u_{j-1}^n - u_j^n) \tag{4}$$

The term $\frac{a\Delta t}{\Delta x}$ is the CFL (Courant, Friedrichs, and Lewy) number, sometimes denoted by σ. It plays a pivotal role in determining the stability of a scheme such as this one. More generally, we will consider finite difference schemes of the form

$$u_j^{n+1} = Lu_j^n = \sum_{|s| \le S} c_s u_{j+s}^n$$

where c_s is independent of u and j and s is an integer set by the specific spatial discretization used. We often write this as a discrete evolution operator acting on sequences, i.e.

$$u^n = \{u^n{}_j\} : u^{n+1} = Lu^n$$

Note that in this case, the operator L is linear, i.e., $L(\alpha v + w) = \alpha L(v) + L(w)$, where α is a real or complex number, and v, w are grid vectors. Also note that the evolution operator is explicit, i.e., u_j^{n+1} is a only function of the values at the current time step, u_{j+s}^n, where $|s| \le S$.

1.1 Consistency, stability, and convergence

So far, we have adequately expressed the discretized form of the governing equation to define the linear operator L. We would like to evaluate how well this operator stacks up against the exact solution. In addition, we would also like to know whether or not a specific numerical scheme will converge. To address these issues, we will carefully define a benchmark value against which to compare the computed solution. Let us denote $u_e = u_e(x, t)$ as the exact solution to the original differential equation (1),

$$\frac{\partial u_e}{\partial t} + a\frac{\partial u_e}{\partial x} = 0 \tag{5}$$

We can evaluate this exact solution at the discrete points in space and time where the numerical solution is defined. We define $u_{e,j}(t) = u_e(j\Delta x, t)$ and $u_{e,j}^n = u_e(j\Delta x, n\Delta t)$.

We also denote the sequences $u_e(t) = \{u_{e,j}(t)\}_j$ and $u^n{}_e = \{u^n{}_{e,j}\}_j$.

The local truncation error (*LTE)* of a scheme measures how much the discrete evolution differs from the exact solution after one time step and is given by:

$$LTE = u_e^{n+1} - Lu_e^n \tag{6}$$

Graphically, in u_e space:

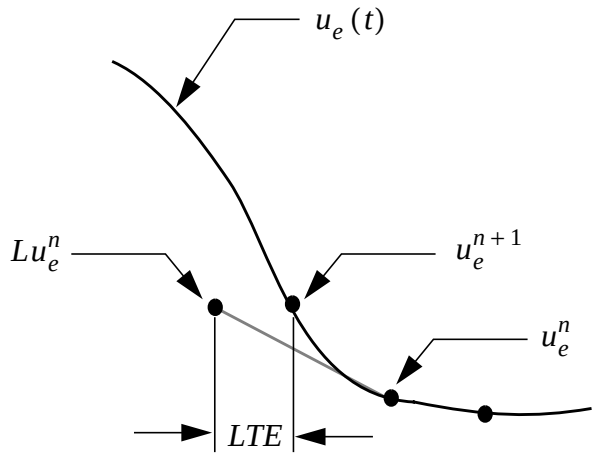


Figure 2. Graphical depiction of the local truncation error

We say that a numerical scheme is consistent if the $LTE \to 0$ as $\Delta x, \Delta t \to 0$:

$$LTE = u_e^{n+1} - Lu_e^n = O\left(\Delta t \sum_{p,\, q \geq 0,\, p+q=l} \Delta t^p \Delta x^q\right) \tag{7}$$

where $l \geq 1$. The order of the *LTE* is given by *l+1*, while the global error of the numerical scheme is of order *l*.

What is the local truncation error for upwind differencing? We showed earlier in equation (4) that the evolution equation is:

$$(Lu_e^n)_j = u_{e,j}^n + \sigma\left(u_{e,j-1}^n - u_{e,j}^n\right) \tag{8}$$

where σ is the CFL number defined by:

$$\sigma = \frac{a\Delta t}{\Delta x} \tag{9}$$

The *LTE* can therefore be written:

$$\begin{aligned} LTE &= u_{e,j}^{n+1} - \left[u_{e,j}^n + \sigma\left(u_{e,j-1}^n - u_{e,j}^n\right)\right] \\ &= \frac{\partial u_e}{\partial t}\Delta t + O\left(\Delta t^2\right) + \sigma\frac{\partial u_e}{\partial x}\Delta x + \sigma O\left(\Delta x^2\right) \\ &= \left(\frac{\partial u_e}{\partial t} + a\frac{\partial u_e}{\partial x}\right)\Delta t + O\left(\Delta x \Delta t + \Delta t^2\right), \end{aligned}$$

where all derivatives are evaluated at $(j\Delta x, n\Delta t)$. From the PDE, we know that the first term on the right-hand side above is zero. We can thus express the *LTE* as:

$$LTE = \Delta t O(\Delta x + \Delta t) \tag{10}$$

The coefficients multiplying the bounds on the local truncation error are all proportional to second derivatives of the solution (and therefore, of the initial data). They are uniformly bounded as long as those derivatives are also bounded. Looking back at the requirement for consistency above in equation (7), we can say that upwind differencing is consistent to first order globally. Notice that we used in an essential way that u_e^n is the exact solution to the PDE evaluated at discrete points in space and time.

Suppose that we had two discrete solutions at a given time, u_e^n and u^n. Let's apply the operator L to both solutions and look at how far they move apart in one time step. Consider:

$$\begin{aligned} \left\| u_e^{n+1} - u^{n+1} \right\| &= \left\| u_e^{n+1} - Lu_e^n + Lu_e^n - Lu^n \right\| \\ &\leq \left\| u_e^{n+1} - Lu_e^n \right\| + \left\| Lu_e^n - Lu^n \right\|. \end{aligned} \tag{11}$$

The first term on the right-hand side of equation (11) above is by now familiar to us as the local truncation error, a measure of the consistency error. The second term measures how far apart two discrete solutions are after one time step. A method is stable if $\left\| Lu_e^n - Lu^n \right\|$ can be bounded appropriately by $\left\| u_e^n - u^n \right\|$, multiplied by a constant which is independent of both u^n and u_e^n.

This leads to the Lax equivalence theorem, which states that stability and consistency for a given numerical scheme imply its convergence. Analysis of the stability for a given linear difference scheme is simplified by the fact that stability implies boundedness (and vice versa) for any L. So we would like to show that:

$$\begin{aligned} \| Lu - Lv \| &= \| L(u - v) \| \\ &= \| L\varepsilon \| \\ \| Lu - Lv \| &\leq (1 + K\Delta t) \| \varepsilon \| \end{aligned} \tag{12}$$

The following questions arise:

- What is the proper choice for the norm $\| \ \|$?
- What is the constant K in the stability bound?

For the first question, we can take a hint from the analogous bounds in the exact solution, i.e., that the solution translates with shape unchanged for all time. The L_∞ norm takes as its value the maximum magnitude over the domain, i.e.:

$$\|\delta\|_\infty = max_x|\delta(x)| \tag{13}$$

Looking at the form of the exact solution to the advection equation, we deduce that the L_∞ norm does not change in time. However, the analogous restriction on the discrete formulation is inconvenient for a numerical scheme. Ultimately, the goal is to create a numerical scheme which is at least second-order accurate. We shall show later (Godunov's Theorem) that a linear scheme which is consistent and max-norm stable is only first order accurate. Therefore, we will instead use the L_2 norm, defined by:

$$\|\delta\|_2 = [\int|\delta|^2 dx]^{\frac{1}{2}} \tag{14}$$

Again, it is clear that this norm does not change in time for the exact solution. We take as the discrete norm

$$\|\delta\|_2 = \left[\sum_j \delta_j^2 \Delta x\right]^{\frac{1}{2}} \tag{15}$$

This also answers the question of what *K* ought to be in equation (12). Since the exact solution doesn't grow with time, we would like to ensure that no growth occurs in the numerical solution, which can be assured if we choose $K = 0$.

Now let's prove the Lax equivalence theorem, which is so important that we will give its statement a special paragraph all of its own:

Lax Equivalence Theorem: $Stability + Consistency \rightarrow Convergence$

Proof:

$$\begin{aligned}\left\|u^n - u_e^n\right\| &\le \left\|Lu^{n-1} - Lu_e^{n-1}\right\| + \left\|Lu_e^{n-1} - u_e^n\right\| \\ &\le \left\|u^{n-1} - u_e^{n-1}\right\| + \Delta t O\left(\sum_{p+q=l} (\Delta x)^p (\Delta t)^q\right) \\ &\le \left\|u^0 - u_e^0\right\| + n\Delta t O\left(\sum_{p+q=l} (\Delta x)^p (\Delta t)^q\right)\end{aligned}$$

As $\Delta t \rightarrow 0$, $n\Delta t \rightarrow T$, and the desired result is proven. Recall that *l* is the global order of the method, while *l+1* is the order of the *LTE*.

1.2 The Von Neumann stability analysis

We will now restrict the problem further: instead of considering solutions on the entire real line, we will consider them on the interval *[0, D]* with periodic boundary conditions and given initial conditions:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{16}$$

$$u(0, t) = u(D, t) \tag{17}$$

$$u(x, 0) = \psi(x) \tag{18}$$

Equivalently, we can consider periodic initial data on the real line; from that point of view, we still have an exact solution $u_e(x, t) = u(x - at, 0) = \psi(x - at)$.

The discrete solution is given by $u^n = \{u_j^n\}$ where the range of j is $j = 0, \ldots, M-1$ and the mesh spacing $\Delta x = 1/M$. For the purposes of computing boundary conditions, u^n is extended periodically, so that $u_{j+Mp}^n = u_j^n$ for all integers p. We can define the linear finite-difference schemes as the sum of shifts:

$$Lu^n = \sum c_s u_{j+s} \tag{19}$$

or

$$L = \sum c_s S^s \tag{20}$$

This shift notation represents location relative to cell j and is defined by

$$(Su)_j = u_{j+1}, j = 0, \ldots, M-2 \tag{21}$$

$$(S^{-1}u)_j = u_{j-1}, j = 1, \ldots, M-1 \tag{22}$$

At the endpoints of the periodic domain:

$$(Su)_{M-1} = u_0 = u_M \tag{23}$$

$$(S^{-1}u)_0 = u_{M-1} = u_{-1} \tag{24}$$

The following complex exponentials form the orthonormal basis for the set of all square-integrable functions on the interval *[0, D]*:

$$W^{(k)}(x) = e^{\frac{2\pi i k x}{D}}, k = 0, \pm 1, \pm 2, \ldots \tag{25}$$

In addition, these exponentials formally diagonalize the $\frac{\partial}{\partial x}$ operator:

$$\frac{\partial}{\partial x}\left(e^{\frac{2\pi i k x}{D}}\right) = \left(\frac{2\pi i k}{D}\right) e^{\frac{2\pi i k x}{D}} \tag{26}$$

Can we find an analogous basis for the discrete solutions $\{u_j^n\}_{j=0}^{M-1}$ that diagonalizes L?

We define the following inner product on complex M-dimensional vectors:

$$\langle u, v \rangle = \frac{1}{M} \sum_{j=0}^{M-1} u_j \bar{v}_j \tag{27}$$

where the overbar denotes the complex conjugate. Assuming that M is even, we define the vectors $w^{(k)}$, $k = -\frac{M}{2} + 1, \ldots, \frac{M}{2}$ as the values of the continuous exponentials evaluated at discrete points:

$$w_j^{(k)} = W^{(k)}(j\Delta x) = e^{\frac{2\pi i k j \Delta x}{D}} \tag{28}$$

There are two useful results which follow from this definition:

(i) The vectors $w^{(k)}$ form a complete orthonormal basis with respect to the inner product:

$$\langle w^{(k)}, w^{(k')} \rangle = \begin{cases} 1 \ldots if \ldots k = k' \\ 0 \ldots otherwise \end{cases} \tag{29}$$

Proof:

Compute:

$$\langle w^{(k)}, w^{(k')} \rangle = \frac{1}{M} \sum_{j=0}^{M-1} e^{\frac{2\pi i (k-k') j \Delta x}{D}}$$

$$= \frac{1}{M} \sum_{j=0}^{M-1} z^j$$

where $z = e^{\frac{2\pi i (k-k') \Delta x}{D}}$. Continuing:

$$\langle w^{(k)}, w^{(k')} \rangle = \frac{1}{M} \frac{z^M - 1}{z - 1}$$

$$= \begin{cases} 1 \ldots if \ldots z = 1 \\ 0 \ldots if \ldots z \neq 1 \end{cases}$$

Saying that $z = 1$ is equivalent to saying that $\frac{(k-k')\Delta x}{D} = p$, where p is an integer. For k and k' in the range of $-\frac{M}{2} + 1, \ldots, \frac{M}{2}$, this is possible only if $k = k'$.

(ii) The second feature is that the application of the shift operator becomes ridiculously easy:

$$S w^{(k)} = e^{i\beta} w^{(k)} \tag{30}$$

and

$$S^{-1} w^{(k)} = e^{-i\beta} w^{(k)} \tag{31}$$

where

$$\beta = \frac{2\pi k \Delta x}{D} \tag{32}$$

Proof: Compute:

$$(Sw^{(k)})_j = w_{j+1}^{(k)} \tag{33}$$

$$= e^{\frac{2\pi i k (j+1) \Delta x}{D}} \tag{34}$$

$$= e^{\frac{2\pi i k \Delta x}{D}} e^{\frac{2\pi i k j \Delta x}{D}} \tag{35}$$

$$= e^{i\beta} w_j^{(k)}, j = 0, \ldots, M-2 \tag{36}$$

This still holds at the edge of the domain

$$(Sw^{(k)})_{M-1} = w_0^{(k)} = 1 \tag{37}$$

$$= e^{\frac{2\pi i k \Delta x}{D}} e^{\frac{2\pi i k (M-1) \Delta x}{D}} \tag{38}$$

$$= e^{i\beta} w_{M-1}^{(k)}, \tag{39}$$

The procedure for S^{-1} is similar.

Given the basis vectors $w^{(k)}$, we can arrive at a particularly simple criterion for stability. Define the norm to be that given by the inner product: $\|u\|^2 = \langle u, u \rangle$. We also note that the basis $w^{(k)}$ diagonalizes L. In addition, L acts on the $w^{(k)}$'s by simple complex multiplication:

$$S^2 w^{(k)} = S(Sw^{(k)}) = S(e^{i\beta} w^{(k)}) = e^{2i\beta} w^{(k)} \tag{40}$$

This procedure can be repeated so that we can write:

$$Lw^{(k)} = (\sum c_s S^s) w^{(k)} = \sum c_s e^{i\beta s} w^{(k)} \tag{41}$$

We define the coefficient $\lambda(\beta) = \sum c_s e^{i\beta s}$ to be the *symbol* of L.

Let u be some vector; u can then be expanded in terms of the $w^{(k)}$'s:

$$u = \sum_{k=-\frac{M}{2}+1}^{\frac{M}{2}} b_k w^{(k)} \tag{42}$$

where $b_k = \langle u, w^{(k)} \rangle$ and $\|u\|^2 = \sum |b_k|^2$. Then

$$Lu = L \sum b_k w^{(k)} \tag{43}$$

$$= \sum b_k \lambda(\beta)\, w^{(k)} \tag{44}$$

and

$$\|Lu\|^2 = \sum |b_k|^2 |\lambda(\beta)|^2 \tag{45}$$

If $|\lambda(\beta)| \le 1$, then $\|Lu\| \le \|u\|$. Note that the range of β is contained in $-\pi < \beta \le \pi$. To check stability for all $M \to \infty$, we need only look at a single function $\lambda(\beta)$.

The various options for determining the stability of a particular numerical discretization are:

(i) Analytically verify a bound of the form $|\lambda(\beta)| \le 1$ for $-\pi < \beta \le \pi$. This is difficult to do in all but the simplest cases.

(ii) Verify the bound computationally, i.e., sample $\lambda(\beta)$ and see whether it is less than or equal to 1; or possibly graph $|\lambda(\beta)| - 1$.

(iii) Compute a Taylor series around $\beta = 0$ of:

$$|\lambda(\beta)|^2 = 1 + C_q \beta^q + O(\beta^{q+1}) \tag{46}$$

(a) If $C_q > 0$, then the scheme is *unstable* and the solution will grow without limit.

(b) If $(C_q < 0)$, then the scheme *may be stable* (*is* stable for long wavelengths).

In practice, a combination of (ii) and (iii) above give the most insight on “complicated” schemes. When we see instability for $\beta \sim 0$, this means that $k\Delta x \ll 1$ and long-wavelength signals are unstable. This means big trouble. When $\beta \sim \pi$ and we see wavelengths on the order of $2\Delta x$, we may be able to damp these scales.

1.3 Some simple finite difference schemes

Let’s get into the specifics of stability calculations for a number of common finite-difference schemes.

1.3.1 Upwind Differencing

For this case, the discretized PDE is given as follows:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + O(\Delta t) + a \frac{u_j^n - u_{j-1}^n}{\Delta x} + O(\Delta x) = 0 \tag{47}$$

or rewriting as an explicit evolution equation:

$$u_j^{n+1} = u_j^n + \sigma (u_{j-1}^n - u_j^n) \tag{48}$$

where σ is the CFL number, defined as:

$$\sigma = \frac{a\Delta t}{\Delta x} \tag{49}$$

We can write the discretized evolution equation in shift notation:

$$Lu^n = u^n + \sigma (S^{-1} - I) u^n \tag{50}$$

where I is the identity. The symbol for this scheme is deduced from the form above in equation (50),

$$\lambda (\beta) = 1 + \sigma (e^{-i\beta} - 1)$$

or

$$\lambda (\beta) = (1 - \sigma) + \sigma e^{-i\beta} \tag{51}$$

The constraint on the CFL number arises from the requirement that $|\lambda (\beta)| \leq 1$ for all β. For upwinding:

$$\begin{aligned}
\lambda (\beta) \bar{\lambda} (\beta) &= ((1 - \sigma) + \sigma e^{-i\beta}) ((1 - \sigma) + \sigma e^{i\beta}) \\
&= (1 - \sigma)^2 + 2\sigma (1 - \sigma) (e^{i\beta} + e^{-i\beta}) + \sigma^2 \\
&= 1 - 2\sigma + 2\sigma^2 + 2\sigma (1 - \sigma) \cos\beta \\
&= 1 - 2\sigma + 2\sigma^2 - 2\sigma (1 - \sigma) ((\cos\frac{\beta}{2})^2 + 1) \\
&= 1 - 4\sigma (1 - \sigma) (\sin\frac{\beta}{2})^2
\end{aligned}$$

From this result, we can see that if $0 < \sigma < 1$, then $|\lambda (\beta)| \leq 1$ and the scheme is stable. (For this range of σ, the symbol $|\lambda (\beta)|$ can also be seen to be a linear interpolation between 1 and $e^{-i\beta}$ by a geometric argument.)

1.3.2 Downwind Differencing

For this scheme, we will keep the same forward differencing in time, as in upwind differencing above, but replace the spatial derivative with downwind differencing:

$$\left.\frac{\partial u}{\partial x}\right|_{j\Delta x,\, n\Delta t} = \frac{u_{j+1}^n - u_j^n}{\Delta x} + O(\Delta x) \tag{52}$$

The evolution equation is:

$$u_j^{n+1} = u_j^n + \sigma (u_j^n - u_{j+1}^n) \tag{53}$$

We can see from equation (53) above that the linear operator L should be, in terms of shift notation:

$$L = I + \sigma (I - S) \tag{54}$$

The symbol is:

$$\lambda(\beta) = 1 + \sigma (1 - e^{i\beta}) \tag{55}$$

or

$$\lambda(\beta) = (1 + \sigma) - \sigma e^{i\beta} \tag{56}$$

Although the truncation error indicates that this scheme has the same order accuracy as forward differencing, this scheme has the undesirable property that is *unstable* for all $\sigma > 0$!

1.3.3 Centered Differencing

This time, we'll replace the partial derivative with respect to x by the following centered difference approximation:

$$\frac{\partial u}{\partial x} = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + O(\Delta x^2) \tag{57}$$

resulting in the following evolution equation:

$$u_j^{n+1} = u_j^n + \frac{\sigma}{2} (u_{j-1}^n - u_{j+1}^n) \tag{58}$$

so that the symbol for this scheme is:

$$\lambda(\beta) = 1 + \frac{\sigma}{2} (e^{-i\beta} + e^{i\beta}) = 1 - i\sigma \sin\beta \tag{59}$$

and

$$|\lambda(\beta)| = [1 + \sigma^2 (\sin\beta)^2]^{1/2} \geq 1 \tag{60}$$

Therefore, this scheme is also unstable.

1.3.4 The Two-Step Lax-Wendroff Scheme

This predictor / corrector method adds an extra step in solving for half-step values in time (the "predictor" step) and a subsequent evolution to the next time level (the "corrector" step) based on those half-step values. The scheme has the dual advantages of a global spatial error of $O(\Delta x^2)$ and is stable for reasonable values of σ. We can think of this graphically as solving for the discrete values of u at the points in space/time shown by the circles in below.

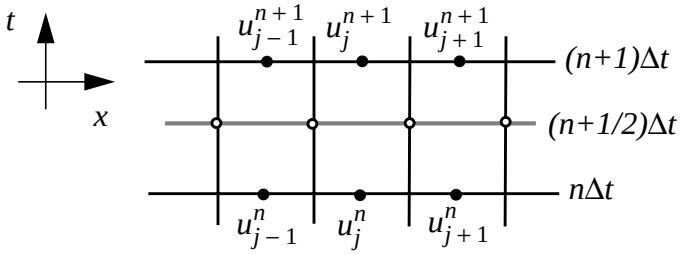


Figure 3. Discretization for the Lax-Wendroff predictor / corrector method

The solid circles represent quantities at the full-time step. We know the values at $n\Delta t$ before the start of the time step; we would like to solve for the values at $(n+1)\Delta t$. In order to calculate this, we will have to find values for the points in open circles. How are we going to do this? In the predictor step, we will solve for the half-step values:

$$\tilde{u}_{j+1/2}^{n+1/2} = \frac{1}{2}(u_j^n + u_{j+1}^n) + \frac{\sigma}{2}(u_j^n - u_{j+1}^n) + O(\Delta x^2) \tag{61}$$

In the subsequent corrector step:

$$u_j^{n+1} = u_j^n + \sigma(\tilde{u}_{j+1/2}^{n+1/2} - \tilde{u}_{j-1/2}^{n+1/2}) \tag{62}$$

We would like to show that this scheme is globally second-order accurate, i.e., if u_e is the exact solution, then we must show that the local truncation error is:

$$u_e^{n+1} - L(u_e) = O(\Delta x^3) \tag{63}$$

We will do this in two parts. First we need to show that:

$$u_{e,j+1/2}^{n+1/2} - \tilde{u}_{e,j+1/2}^{n+1/2} = C\left(\left(j + \frac{1}{2}\right)\Delta x\right)\Delta x^2 + O(\Delta x^3) \tag{64}$$

where $u^{n+1/2}_{e,j+1/2} = u_e((j+\frac{1}{2})\Delta x, (n+\frac{1}{2})\Delta t)$ and C denotes a smooth function of the quantities in parentheses. Using the expression for the half-step values from equation (61) above:

$$u^{n+1/2}_{e,j+1/2} - \tilde{u}^{n+1/2}_{e,j+1/2} = u^{n+1/2}_{e,j+1/2} - (\frac{1}{2}(u^n_{e,j} + u^n_{e,j+1}) + \frac{\sigma}{2}(u^n_{e,j} - u^n_{e,j+1}))$$

$$= u^{n+1/2}_{e,j+1/2} - \left[u^n_{e,j+1/2} + \frac{1}{2}(\frac{\Delta x}{2})^2 \frac{\partial^2 u_e}{\partial x^2}\bigg|_{n\Delta t,\ (j+\frac{1}{2})\Delta x} + O(\Delta x^3) \right]$$

$$+ \frac{\sigma}{2}\left[\Delta x \frac{\partial u_e}{\partial x}\bigg|_{n\Delta t,\ (j+\frac{1}{2})\Delta x} + O(\Delta x^3) \right]$$

From this point, we will assume implicitly that all derivatives are evaluated at the point $(n\Delta t,\ (j+1/2)\Delta x)$. Continuing:

$$u^{n+1/2}_{e,j+1/2} - \tilde{u}^{n+1/2}_{e,j+1/2} = \frac{\Delta t}{2}\frac{\partial u_e}{\partial t} + \frac{\Delta t^2}{8}\frac{\partial^2 u_e}{\partial t^2} + O(\Delta t^3) - \frac{\Delta x^2}{8}\frac{\partial^2 u_e}{\partial x^2} + O(\Delta x^3)$$

$$+ \frac{a\Delta t}{2}\frac{\partial u_e}{\partial x} + O(\Delta x^3)$$

$$= \frac{\Delta t}{2}\left[\frac{\partial u_e}{\partial t} + a\frac{\partial u_e}{\partial x} \right] + \frac{\Delta t^2}{8}\frac{\partial^2 u_e}{\partial t^2} + O(\Delta t^3) - \frac{\Delta x^2}{8}\frac{\partial^2 u_e}{\partial x^2} + O(\Delta x^3)$$

$$= \left[\frac{1}{8}(\frac{\sigma}{a})^2 \frac{\partial^2 u_e}{\partial t^2} - \frac{1}{8}\frac{\partial^2 u_e}{\partial x^2} \right]\Delta x^2 + O(\Delta x^3, \Delta t^3)$$

which means that if u_e is smooth, we have arrived at the desired result:

$$u_{e,j+1/2}{}^{n+1/2} - \tilde{u}^{n+1/2}_{e,j+1/2} = C((j+\frac{1}{2})\Delta x^2 + O(\Delta x^3))$$

With this result in hand, we can now show that the local truncation error is of $O(\Delta x^3)$:

$$u^{n+1}_{e,j} - L(u^n_e)_j = u^{n+1}_{e,j} - u^n_{e,j} + \sigma(\tilde{u}^{n+1/2}_{e,j+1/2} - \tilde{u}^{n+1/2}_{e,j-1/2})$$

$$= u^{n+1}_{e,j} - u^n_{e,j} + \sigma(u^{n+1/2}_{e,j+1/2} - u^{n+1/2}_{e,j-1/2})$$

$$+ \sigma\Delta x^2 (C((j+\frac{1}{2})\Delta x) - C((j-\frac{1}{2})\Delta x)) + O(\Delta x^3)$$

$$= \Delta t \frac{\partial u_e}{\partial t}\bigg|_{(n+\frac{1}{2})\Delta t, j\Delta x} + O(\Delta t^3) + \sigma \Delta x \frac{\partial u_e}{\partial x}\bigg|_{(n+\frac{1}{2})\Delta t, j\Delta x} + O(\Delta x^3)$$

$$= \Delta t \left[\frac{\partial u_e}{\partial t} + a \frac{\partial u_e}{\partial x} \right] + O(\Delta x^3, \Delta t^3)$$

Note that the term in brackets is zero. The delicate cancellations of the above procedure depended in a critical way on the following assumptions: (a) that the function u is smooth; and (b) that the grid spacing is uniform. The first assumption begs the question of what happens at shocks and other discontinuities. The second brings up the question of irregular grids.

Now that we have examined the accuracy of this method, let's take a look at the stability. We can write the equation for the predictor step in shift notation as:

$$\tilde{u}^{n+1/2} = \{ \frac{1}{2} [I + S] + \frac{\sigma}{2} [I - S] \} u^n \tag{65}$$

Putting this into the expression for the corrector step:

$$u^{n+1} = \{ I + \sigma (S^{-1} - I) \left[\frac{1}{2} (I + S) + \frac{\sigma}{2} (I - S) \right] \} u^n \tag{66}$$

This means that we can write the symbol as:

$$\lambda(\beta) = 1 + \sigma (e^{-i\beta} - 1) \left[\frac{1}{2} (1 + e^{i\beta}) + \frac{\sigma}{2} (1 - e^{i\beta}) \right] \tag{67}$$

or in simpler form

$$\lambda(\beta) = \sigma^2 \cos\beta + (1 - \sigma^2) - i\sigma \sin\beta \tag{68}$$

Recall that $\beta = 2\pi k \Delta x / D$. Is this stable? We can check this by performing a Taylor expansion,

$$|\lambda(\beta)|^2 = 1 + (\sigma^4 - \sigma^2)(\cos\beta - 1)^2 \tag{69}$$

$$= 1 + \frac{\sigma^4 - \sigma^2}{4} \beta^4 + O(\beta^6) \tag{70}$$

Thus we get lucky, because:

$$1 - |\lambda(\beta)|^2 = (\sigma^4 - \sigma^2)(\cos\beta - 1)^2 \tag{71}$$

which will be true for $\sigma \leq 1$. Although the Lax-Wendroff scheme is unstable for $\sigma > 1$, it is stable in the long-wavelength limit for $\sigma \leq 1$.

1.4 Upwind schemes and the geometric interpretation of finite differences.

Suppose we wanted to find the function *u(x,t)* which satisfies the scalar hyperbolic equation:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{72}$$

and appropriate boundary conditions for some constant $a \geq 0$. In an upwinding scheme, we think of the direction of information transfer as flowing from the upstream to the downstream locations. Thus, to evaluate something at point $j\Delta x$, we only need data from upstream locations. Using single-sided differencing (forward in time, backward in space), the finite-difference representation of equation (72) above is:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_{j-1}^n - u_j^n}{\Delta x} = 0 \tag{73}$$

Or, rewriting, we might express the discrete evolution of *u* as:

$$u_j^{n+1} = u_j^n + \sigma\,(u_{j-1}^n - u_j^n) \tag{74}$$

where the CFL number, $\sigma = a\Delta t/\Delta x$, is a positive number. We can interpret u_j^n as the average value of the function *u(x,t)* over the *j*th cell at time $n\Delta t$:

$$u_j^n \approx \frac{1}{\Delta x}\int_{(j-\frac{1}{2})\Delta x}^{(j+\frac{1}{2})\Delta x} u\,(x, n\Delta t)\,dx \tag{75}$$

where $u_j^n \Delta x$ is the total amount of the conserved quantity in cell *j*, e.g., mass, momentum, energy, etc. Graphically:

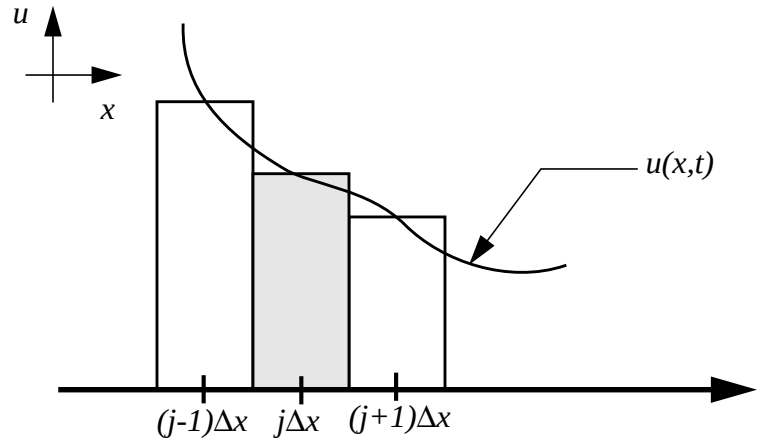


Figure 4. The geometric interpretation of the numerical solution.

The area of the shaded rectangle represents an approximation to the area under the function *u(x,t)*, i.e., $u_j^n \Delta x$. We can treat each cell as a control volume and, assuming no generation within the cell, derive a conservation equation in the form:

$$\Delta x u_j^{n+1} = \Delta x u_j^n + F_{in} - F_{out} \tag{76}$$

where *F* represents a flux through the cell edge. In order to compute $u_j^n \Delta x$, we would like to find a suitable interpolating function $u_I(x)$ at every time step which is easy to calculate but does not violate the conservation constraint. We therefore require that $u_I(x)$ satisfy:

$$u_j^n \Delta x = \int_{(j-\frac{1}{2})\Delta x}^{(j+\frac{1}{2})\Delta x} u_I(x)\, dx \tag{77}$$

The quantity $u_j^n \Delta x$ thus represents a *constraint* on the choice of $u_I(x)$. A trivial example of such an interpolating function might be simply a piecewise constant interpolation, i.e., $u_I(x) = u_j^n$ for $(j-\frac{1}{2})\Delta x < x < (j+\frac{1}{2})\Delta x$.

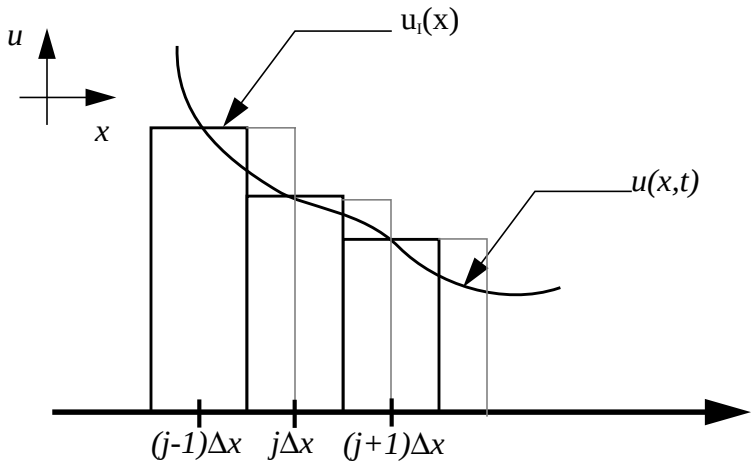


Figure 5. Piecewise constant interpolation function $u_I(x)$.

The broad outline for the classical finite-difference scheme can be broken up into the following steps:

(i) Given u^n, construct $u_I(x)$

(ii) Compute the exact solution to the differential equation, using the interpolated function at time $n\Delta t$: $u_I^{n,\,n+1}(x,\ (n+1)\,\Delta t) = u_I(x - a\Delta t)$

(iii) Calculate u_j^{n+1} to be the average of $u_I^{n,\,n+1}(x,\ (n+1)\,\Delta t)$ over the finite difference cell:

$$u_j^{n+1} = \frac{1}{\Delta x}\int u_I^{n,\,n+1}(x,\ (n+1)\,\Delta t)\,dx$$

$$= \frac{1}{\Delta x}\int_{(j-\frac{1}{2})\,\Delta x}^{(j+\frac{1}{2})\,\Delta x} u_I(x - a\Delta t)\,dx$$

or simply

$$u_j^{n+1} = \frac{1}{\Delta x}\int_{(j-\frac{1}{2})\,\Delta x - a\Delta t}^{(j+\frac{1}{2})\,\Delta x - a\Delta t} u_I(x)\,dx \tag{78}$$

A geometric representation of this process is shown in Figure 6

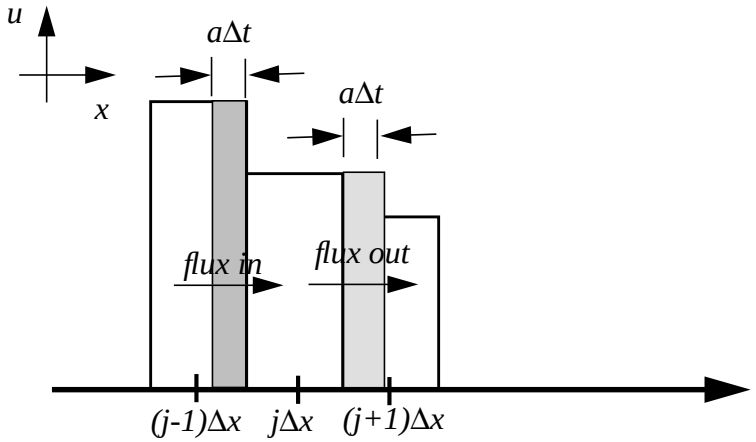


Figure 6. Geometric representation of fluxes moving through cell *j*

The advantage of this formulation is that the second and third steps outlined above are *exact*. The error is therefore governed entirely by the interpolation procedure. This observation would suggest that the next logical step would be to improve this step of the numerical method. Fromm's scheme (i.e., second-order upwinding) utilizes a piecewise linear interpolation function for $u_I(x)$, i.e.

$$u_I(x) = u_j^n + \frac{(x - j\Delta x)}{\Delta x}\Delta u_j \tag{79}$$

where Δu_j is the undivided difference, or the finite-difference approximation to:

$$\Delta u_j \cong \left.\frac{\partial u}{\partial x}\right|_{j\Delta x} \Delta x$$

For example, this numerical scheme uses central differencing:

$$\Delta u_j = \frac{u_{j+1}^n - u_{j-1}^n}{2} \tag{80}$$

Note that this method is not necessarily continuous at the cell edges:

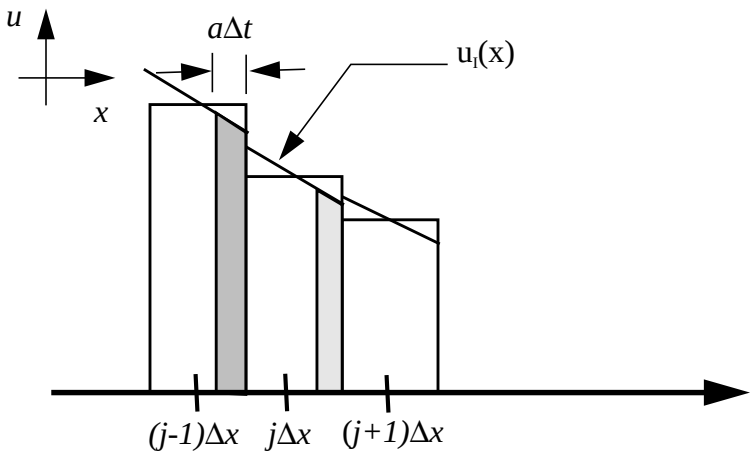


Figure 7. Conservation considerations for piecewise linear interpolation

We know that the integral of $u_l(x)$ over each cell is constrained by $u_j^n \Delta x$ so that we can express the conservation form of the finite-difference representation as

$$\Delta x u_j^{n+1} = \Delta x u_j^n + F_{in} - F_{out} \tag{81}$$

where the flux in, F_{in}, is shown in the figure above by the cross-hatched region and is given by:

$$F_{in} = a\Delta t u_I\left[(j - \frac{1}{2})\Delta x - \frac{a\Delta t}{2}\right]$$

The interpolation function is evaluated at the location specified in the brackets. An equivalent form is:

$$F_{in} = a\Delta t\left[u^n_{j-1} + \frac{1}{2}(1-\sigma)\,\Delta u_{j-1} \right] \tag{82}$$

Let us denote the term in the brackets above by $u_{j-1/2}$, since it is actually the first-order accurate approximation to the time-centered value of the solution at the cell edge, in the sense of the predictor-corrector formulation of Lax-Wendroff. The flux out is given by substituting the subscript j for $j-1$ in (82).

Another way to arrive at the evolution equation is by interpolating u_j^n from j to $j+1/2$ at time level $n\Delta t$, and then interpolating from that spatial location to the next time increment $(n+1)\Delta t$:

$$u^{n+\frac{1}{2}}_{j+\frac{1}{2}} \cong u\left((j+\frac{1}{2})\,\Delta x,\ (n+\frac{1}{2})\,\Delta t \right)$$

$$u^{n+\frac{1}{2}}_{j+\frac{1}{2}} = u^n_j + \frac{\Delta x}{2}\frac{\partial u}{\partial x}\bigg|_{j\Delta x} + \frac{\Delta t}{2}\frac{\partial u}{\partial t}\bigg|_{n\Delta t} + \ldots$$

Next, using the PDE to substitute for the temporal derivative

$$u^{n+\frac{1}{2}}_{j+\frac{1}{2}} = u^n_j + \left[\frac{\Delta x}{2} - \frac{a\Delta t}{2} \right] \frac{\partial u}{\partial x}\bigg|_{j\Delta x} + \ldots$$

or rewriting

$$u^{n+\frac{1}{2}}_{j+\frac{1}{2}} = u^n_j + \frac{1}{2}(1-\sigma)\,\Delta u_j \tag{83}$$

which is the same as before. The local truncation error is $O(\Delta x^3)$, which can be readily proved if enough terms of the Taylor expansion are retained.

Fromm's scheme keeps track of whether the wave speed is positive or negative, and alters the direction of information transfer accordingly.

$$u^{n+\frac{1}{2}}_{j+\frac{1}{2}} = u^n_j - \frac{1}{2}(1+\sigma)\,\Delta u_j \tag{84}$$

This scheme is stable for $\sigma \le 1$, since:

$$|\lambda(\beta)|^2 = 1 + (\sigma-1)\,\sigma\,(\sigma^2-\sigma+1)\,\frac{\beta^4}{4} + \ldots \tag{85}$$

where $\beta = (2\pi k\Delta x)/D$. In addition, the local truncation error is $O(\Delta x^3)$.

1.5 Fourier analysis and accuracy

We have derived and analyzed three stable schemes: upwind differencing, the Lax-Wendroff method, and Fromm's method. From our truncation error analysis for these schemes, we know that the global error for the upwind scheme is $O(\Delta x)$, and that for Lax-Wendroff and Fromm it is $O(\Delta x)^2$. We can check to see whether these estimates are reliable indicators of the actual performance of these methods. In the following table, we give the values of the error, defined by:

$$\left\| u_e^n - u^n \right\| = \sum_{(-M/2)+1}^{M/2} \left| u_{e,j}^n - u_j^n \right|^2 \Delta x = \varepsilon(\Delta x, n\Delta t) \tag{86}$$

For these three schemes, for different values of M. We also show plots of the solution cor-

Table 1L2 error in various schemes for smooth Gaussian pulse

Method	M= 32	M=64	M=128
Upwind	4.11e-02	2.42e-02	1.32e-02
Lax-Wendroff	3.66e-02	1.62e-02	3.92e-03
Fromm	1.62e-02	5.95e-03	1.11e-03

responding to the times that those errors were computed. For these schemes, we find that $\frac{\varepsilon(\Delta x, T)}{\varepsilon(2\Delta x, T)} = 2^{-l}$, where l is the global order of accuracy. Thus the errors go to zero at a rate consistent with the assumption that the higher order terms in the truncation error estimate (74) are negligible, relative to the leading order terms. Comparing the actual values of ε for each of the schemes, we also find that for a fixed mesh spacing, the first order upwind method has much larger errors than either of the second order schemes. However, we also notice that the Lax Wendroff and Fromm schemes have considerably different errors from each other for a given mesh spacing; in particular, the Fromm scheme appears to be more accurate. In this section, we will develop more refined analytical tools that will enable us to distinguish between the two second-order schemes and will provide some more general design principles for finite difference methods.

Let $f_j = \sum_{-M/2+1}^{M/2} b_k w_j^k$ be the discrete Fourier transform of the M-vector f. The function

$$f_S(x) = \sum_{-M/2+1}^{M/2} b_k W^k(x) \tag{87}$$

defined on the interval $[0, D]$ is an interpolation function for the values f_j:

$$f_S(j\Delta x) = \sum_{-M/2+1}^{M/2} b_k W^k(j\Delta x) = f_j \tag{88}$$

Furthermore, if $f_j = f(j\Delta x)$ for $j = 0 \ldots M-1$, for some q-times continuously differentiable function $f(x)$, then

$$\| f(x) - f_S(x) \| = O\left(\frac{1}{M^{q+1}}\right) \tag{89}$$

uniformly in x.

We can apply this to the discretization of (1). Let $\varphi_j = \varphi(j\Delta x)$, where $\varphi(x)$ is infinitely differentiable data for (1), and let

$$\varphi_j = \sum_{-M/2+1}^{M/2} b_k w_j^k$$

be the discrete Fourier transform of the M-vector φ. Then $\varphi_S(x)$ interpolates the discrete data, and (76) for any fixed q as $M \to \infty$.We can define a spectrally accurate (approximate) solution on M points to be the exact solution to (1), with initial data $\varphi_S(x)$:

$$u_S(x, t) = \varphi_S(x - at). \tag{90}$$

From equation (76) we find that,

$$u_S(x, t) - u(x, t) = \varphi_S(x - at) - \varphi(x - at) = O(\Delta x^{q+1}).$$

Thus, for sufficiently large q, the spectrally accurate solution differs from the exact solution by an amount that is much smaller that the difference between either of them and the solutions obtained from the finite difference methods described above. In particular, $u_S(j\Delta x, n\Delta t) - u_j^n$ is an excellent approximation to $u(j\Delta x, n\Delta t) - u_j^n$.

Second, we can also apply Fourier interpolation to the numerical solution u_j^n. We can define, for each n, the function

$$u_I^n(x) = \sum_{-M/2+1}^{M/2} b_k \lambda(\beta)^n W^k(x) \tag{91}$$

Since the discrete Fourier transform diagonalizes the difference operator L, and by the interpolation property (75), we have $u_I^n(j\Delta x) = u_j^n$. Thus we can view the finite difference method as defining a discrete time dynamics on the set of all sums of finite Fourier series of the form (75).

Finally, we can write $u_s(x, n\Delta t)$ in such a way so as to easily compare it to $u_I^n(x)$:

$$u_S(x, n\Delta t) = \sum_{-M/2+1}^{M/2} b_k e^{i\sigma\beta n} W^k(x) \tag{92}$$

where $\beta = 2\pi k\Delta x/D$ and $\sigma = a\Delta t/\Delta x$. From this expression for u_S, it is clear that the spectrally accurate discrete time dynamics on the Fourier modes is given by multiplying each amplitude by $e^{i\sigma\beta}$, and that the numerical solution gives an error exactly to the extent that the symbol $\lambda(\beta)$ differs from $e^{i\sigma\beta}$.

To illustrate the utility of these ideas, we can use the symbol to compute the local truncation error of a finite difference method. For initial data in the form of a finite Fourier sum (75), the LTE is given by

$$u_e^{n+1} - Lu_e^n = \sum_{-M/2+1}^{M/2} b_k (e^{-i\sigma\beta} - \lambda(\beta)) e^{-i\sigma\beta n} w^k. \tag{93}$$

In particular, for a single Fourier mode of wavenumber *k*, with *k* fixed, the statement that the LTE is $O((\Delta x)^{l+1})$ is the same as saying that $e^{-i\sigma\beta} - \lambda(\beta) = O(\beta^{l+1})$. In fact, we can infer the form of the truncation error from the symbol. If

$$e^{-i\sigma\beta} - \lambda(\beta) = C(i\beta)^{l+1} + O(\beta^{l+2})$$

then the LTE can be written in the form

$$u_e^{n+1} - Lu_e^n = \Delta t\left(\frac{Ca}{\sigma}(\Delta x)^l \left.\frac{\partial^{l+1} u_e}{\partial x^{l+1}}\right|_{(j\Delta x,\, n\Delta t)} + O((\Delta x)^{l+1})\right) \tag{94}$$

This may look like a somewhat singular expression, due to the division by σ, but it is not since $C = O(\sigma)$ if the scheme is consistent.

1.5.1 Amplitude error

To determine the difference between the amplitudes of the numerical and exact solutions, we must determine the degree to which the symbol $\lambda(\beta)$ approximates $e^{-i\sigma\beta}$. The amplitude error, *AE*, is:

$$AE = 1 - |\lambda(\beta)|$$

since $|e^{-i\sigma\beta}| = 1$. Note that for a single Fourier mode $|e^{-i\sigma\beta} b_k| = |b_k|$ and that $|\lambda(\beta)\, b_k| \le |b_k|$, so that the amplitudes of Fourier modes tend to decrease as a function of time. In general, the symbol can be expressed as a function of β:

$$|\lambda(\beta)| = 1 + C_q \beta^q + \ldots \tag{95}$$

As we have previously discussed, we want C_q to be negative in order to yield a stable numerical scheme. Taylor expansions for several common discretizations yield the results shown in Table 2.

Table 2. Amplitude error for various schemes

Scheme	$\lambda(\beta)$
Upwind	$1 + \frac{\sigma^2 - \sigma}{2}\beta^2 + \ldots$
Lax-Wendroff	$1 + \frac{\sigma^4 - \sigma^2}{8}\beta^4 + \ldots$
Fromm	$1 + \frac{(\sigma - 1)\,\sigma\,(\sigma^2 - \sigma + 1)}{8}\beta^4 + \ldots$

Since β is proportional to the mesh spacing Δx for fixed k, the amplitude error for both Fromm and Lax-Wendroff is quite small, of $O(\Delta x^4)$. For reasonable CFL numbers (e.g., for compressible flow, one might set the optimal σ at 0.9), upwind differencing does indeed give a negative C_q, and one can predict that the Fourier modes will be damped by $O(\Delta x^2)$.

1.5.2 Phase errors

The phase error is a measure of how accurately the finite difference scheme is translating each Fourier mode in space. The exact solution at time level $n+1$ looks like a translation of the Fourier mode by $\sigma\beta/k = a\Delta t$:

$$u_s(x, n\Delta t) = \sum_{-M/2+1}^{M/2} b_k W^k (x - an\Delta t) \tag{96}$$

We wish to compute the corresponding quantity for u_I^n. Define η by:

$$\frac{\lambda(\beta)}{|\lambda(\beta)|} = e^{-i\eta(\sigma,\beta)} \tag{97}$$

The range of η is from $-\pi$ to π. Multiplication by $e^{-i\eta(\sigma,\beta)}$ corresponds to shifting the argument of W^k by $\eta(\sigma,\beta)/(2\pi k/D)$, so that

$$u_I^n(x) = \sum_{-M/2+1}^{M/2} b_k |\lambda(\beta)|^n W^k(x - n\eta(\sigma,\beta)/(2\pi k/D)) \tag{98}$$
$$= \sum_{-M/2+1}^{M/2} b_k |\lambda(\beta)|^n e^{-i\eta(\sigma,\beta)n} W^k(x)$$

While the exact solution would yield a multiplier of $e^{-i\sigma\beta}$ in equation (xx) above, the numerical solution gives a multiplier of $e^{-i\eta(\sigma,\beta)}$. However, this definition of phase error does not permit us to treat the time step as a parameter, comparing phase error for different values of σ. A definition that allows us to make such comparisons is to define the error in the phase after a length of time required for the solution to move a fixed distance. A natural choice of such a distance is Δx, the length of a finite difference cell. In that case, we can define the phase error using the following expression:

$$\left(e^{i\sigma\beta}\frac{\lambda(\beta)}{|\lambda(\beta)|}\right)^{1/\sigma} = e^{-i\delta} \tag{99}$$

In this case, $\delta/(2\pi k/D)$ is the error in the displacement of the k-th Fourier component after the solution has evolved for a time $\Delta x/a$.

We can use Taylor expansions of the left-hand side of (98) to evaluate δ in the limit of $\beta \to 0$. To leading order,

$$\left(e^{i\sigma\beta}\frac{\lambda(\beta)}{|\lambda(\beta)|}\right)^{1/\sigma} = 1 - iC_{2q+1}\beta^{2q+1} + O(\beta^{2q+3}) = 1 - i\delta + O(\delta^2) \tag{100}$$

where $2q = l$ if l is even, and $2q = l+1$ if l is odd. Given sufficient effort (or a symbolic manipulation package), one can compute $C_{2q+1}\beta^{2q+1}$, the leading order term in the expansion for each of the schemes is shown in Table 3.

Table 3. Phase error for various schemes

Scheme	Phase error (100)
Upwind	$\left(\frac{-(2\sigma^2 - 3\sigma + 1)}{6}\right)\beta^3 + O(\beta^5)$
Lax-Wendroff	$\frac{(\sigma^2 - 1)}{6}\beta^3 + O(\beta^5)$

Table 3. Phase error for various schemes

Scheme	Phase error (100)
Fromm	$(2\sigma^2 - 3\sigma + 1)\frac{\beta^3}{12} + O(\beta^5)$

It is particularly instructive to plot the polynomials in σ that multiply the leading order terms in the phase error (figure needed here) for the case of Fromm's scheme and the Lax-Wendroff scheme. Overall, we see that the phase error for Fromm's scheme is smaller than that for the Lax-Wendroff scheme. In addition, we see that, for $0.5 \le \sigma \le 1$, the phase error for Fromm's scheme is much smaller that for Lax-Wendroff, as well as for that for either scheme for $0 \le \sigma \le 0.5$. In particular, for $\sigma = 0.75$, the value in that range for which the phase error for Fromm has the largest magnitude, the coefficients differ by a factor of 8. Since the leading order term in the phase error is cubic in Δx, this implies that, for this value of the CFL number, it is necessary to reduce Δx by a factor of two in the Lax-Wendroff scheme to obtain the same accuracy as that obtained in Fromm's scheme. This conclusion is supported by the convergence study described at the beginning of the section. As a practical matter, it is usually not possible to run an algorithm at a specified CFL number, since wave speeds vary in space over the grid. However, it is typical of a large class of practical applications that the CFL condition is constrained by the behavior of the solution in the region of greatest interest, and that one runs using a CFL number that is as large as possible. This analysis provides an explanation of what has been observed in much more complicated nonlinear problems in several space dimensions [Woodward and Colella, 1984]: that the improvement in going from Lax-Wendroff to Fromm is worth about a factor of two mesh refinement.

The classical definition of phase error is given by $\lim_{\sigma \to 0} \delta = \delta_c(\beta)$. In terms of the definition (98), it is given by

$$\delta_c = \frac{1}{-i}\frac{d}{d\sigma}\left(e^{i\sigma\beta}\frac{\lambda(\beta)}{|\lambda(\beta)|}\right)\Bigg|_{\sigma=0} = \frac{d}{d\sigma}\left(Im(\lambda)\right)\Bigg|_{\sigma=0} \tag{101}$$

where $Im(\lambda) = \frac{\bar{\lambda} - \lambda}{2i}$. It is easy to see from the table above that the classical phase error for Lax-Wendroff is, to leading order in β, only twice as large as that for Fromm. Thus it gives only a partial, and sometimes misleading picture of how schemes will behave. The advantage is that it is precisely this part of the phase error that is most easily improved. It is not difficult to show that δ_c depends only on the terms in the finite difference scheme that are first order in σ

$$(Lu^n)_j = u_j^n + \sigma (u_{j-1/2}^n - u_{j+1/2}^n) + O(\sigma^2).$$

In particular, one can reduce the classical phase error by improving the accuracy with which $\frac{u_{j-1/2}^n - u_{j+1/2}^n}{\Delta x}$ approximates $\left.\frac{\partial u}{\partial x}\right|_{(j\Delta x,\, n\Delta t)}$.

We can use Richardson extrapolation to construct higher order approximations to spatial derivatives. Given a smooth function *f(x)*, say that we want to compute $f'(0)$, given values on a grid with grid spacing Δx. We can use Taylor expansions to obtain the following error estimates for centered difference approximations to $f'(0)$

$$\frac{f_{j+1} - f_{j-1}}{2\Delta x} = f'(0) + \frac{(\Delta x)^2}{6} f'''(0) + O(\Delta x)^4$$

$$\frac{f_{j+2} - f_{j-2}}{4\Delta x} = f'(0) + \frac{(2\Delta x)^2}{6} f'''(0) + O(\Delta x)^4$$

We can take linear combinations of these two expressions to obtain a fourth-order accurate expression for $f'(0)$:

$$f'(0) = \frac{4}{3}\frac{f_{j+1} - f_{j-1}}{2\Delta x} - \frac{1}{3}\frac{f_{j+2} - f_{j-2}}{4\Delta x} + O(\Delta x)^4 \tag{102}$$

We can write this expression in flux form, as follows:

$$f'(0) = \frac{f_{j+1/2} - f_{j-1/2}}{\Delta x}$$

where

$$f_{j+1/2} = \frac{7}{12}(f_j + f_{j+1}) - \frac{1}{12}(f_{j-1} + f_{j+2})$$

This process can be continued to any order of accuracy, and flux forms derived; for details, see, e.g., Zalesak (1984).

We can now use this to define a Lax-Wendroff scheme with fourth-order spatial accuracy (LW4) and lower classical phase error. It is given by:

$$u_j^{n+1} = u_j^n + \sigma\left(u_{j-\frac{1}{2}}^{n+\frac{1}{2}} - u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right) \tag{103}$$

where

$$u_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{7}{12}(u_j + u_{j+1}) - \frac{1}{12}(u_{j-2} - u_{j-1}) + \frac{\sigma}{2}(u_j^n - u_{j+1}^n) \tag{104}$$

As compared to second-order Lax-Wendroff (LW2), the first two terms on the right-hand side of equation (104) have replaced $(u_j + u_{j+1})/2$. The global phase error in the long-wavelength limit is given by $\delta = \frac{\sigma^2}{6}\beta^3 + O(\beta^5)$. Thus the use of higher order differencing eliminates the leading order term in σ of the phase error, as claimed at the outset. In general, the use of $2q$-th order centered differences in (xxx) leads to a classical phase error $\delta_c = O(\beta^{2q+1})$.

A Taylor expansion of the amplitude error indicates that the scheme as stated is unstable in the long-wavelength limit: $|\lambda(\beta)| = 1 + C\beta^4$ where $C > 0$. A little artificial viscosity can be added to damp out the linear instability. Even-order derivatives are dissipative, so we might try adding a term like $\Delta t \Delta x \frac{\partial^2 u}{\partial x^2}$; however, this would wreak havoc with our error.- Instead, try adding something like $\Delta t \Delta x^3 \frac{\partial^4 u}{\partial x^4}$. Consider the effect of fourth-order derivatives:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + C\frac{\partial^4 u}{\partial x^4} = 0 \tag{105}$$

The last term on the left-hand side of equation(94) above acts to reduce the amplitudes of the Fourier modes. The numerical method is then given as follows:

$$u^{n+\frac{1}{2}}_{j+\frac{1}{2}} = \frac{7}{12}(u_j + u_{j+1}) - \frac{1}{12}(u_{j-2} - u_{j-1}) + \frac{\sigma}{2}(u^n_j - u^n_{j+1}) +$$

$$+ \left(\frac{\sigma^3}{4} + \frac{1}{16}\right)(u^n_{j+2} - u^n_{j-1} - 3(u^n_{j+1} - u^n_j)) \tag{106}$$

In the long-wavelength limit, $C < 0$ for $|\sigma| \le 1$. This does not mean that we can neglect the short wavelengths however; if one computes $|\lambda(\beta)|$ for the full range $0 \le \beta \le \pi$, one finds that this scheme is stable only for $|\sigma| \le 0.6$.

Table 4 summarizes the phase and amplitude errors for a number of numerical schemes, as well as ranges of CFL number for which stability is anticipated.

Table 4. Comparison of accuracy, error and stability of various schemes

Scheme	Accuracy	Phase Error	Amplitude Error	Stability
Upwind	$O(\Delta x)$	$O(\beta^3)$	$O(\beta^2)$	$0 \le \sigma \le 1$
LW2	$O(\Delta x^2)$	$O(\beta^3)$	$O(\beta^4)$	$0 \le \sigma \le 1$
Fromm	$O(\Delta x^2)$	$O(\beta^3)$	$O(\beta^4)$	$0 \le \sigma \le 1$

Table 4. Comparison of accuracy, error and stability of various schemes

Scheme	Accuracy	Phase Error	Amplitude Error	Stability
LW4	$O(\Delta x^2)$	$O(\beta^5)$ ($\sigma \to 0$)	$O(\beta^4)$	$0 \le \sigma \le 0.6$
		$O(\beta^3)$ (finite σ)		

With the exception of upwinding, the phase error swamps the amplitude error for the above schemes. So, from the standpoint of accuracy, it is the phase error which is of the most concern.

1.6 The modified equation.

Discrete systems are not objects that we have a great deal of intuition about, while we do have a substantial understanding of how solutions to differential equations behave. Thus, in this section, we look to develop an intuition about the behavior of errors in difference schemes by deriving a differential equation - called the modified equation - that models the behavior of the error. We start by expressing the local truncation error in the following form

$$u_{e,j}^{n+1} - (Lu_e^n)_j = \Delta t \left(C (\Delta x)^l \frac{\partial^{l+1} u_e}{\partial x^{l+1}} \bigg|_{(j\Delta x,\, n\Delta t)} + O\left((\Delta x)^{l+1} \right) \right) \tag{107}$$

Where the coefficient can be determined from the leading order term in the Taylor expansion of the symbol, for example. From this expression, one observes that locally in time, at least, the effect of replacing the exact solution operator by the result from the difference scheme is to add to the result from the exact solution a quantity of the form Δt times the spatial derivatives in the brackets. For that reason, we define the modified solution u_M as the solution to the following differential equation:

$$\frac{\partial u_M}{\partial t} + a \frac{\partial u_M}{\partial x} - C \Delta x^l \frac{\partial^{l+1} u_M}{\partial x^{l+1}} = 0 \tag{108}$$

for which the expression for Lu_e implied by (107) is a consistent discretization in time.- Thus, u_M will better describe what the difference scheme is doing than u_e. If we define $u_{M,j}^n$ to be the solution to this differential equation evaluated at grid points, analogously to $u_{e,j}^n$, then the difference between the modified solution at time level *n+1* and the evolved modified solution is:

$$u_M^{n+1} - Lu_M^n = \Delta t O(\Delta x^{l+1}) \tag{109}$$

For the discrete solution

$$u^{n+1} - Lu^n = 0 \tag{110}$$

For the exact solution

$$u_E^{n+1} - Lu_E^n = \Delta t O(\Delta x^l) \tag{111}$$

thus we find that the discrete dynamics are approximated to one higher order of accuracy by the solution to the modified equation that by the solution to the original differential equation, as illustrated in the figure below.

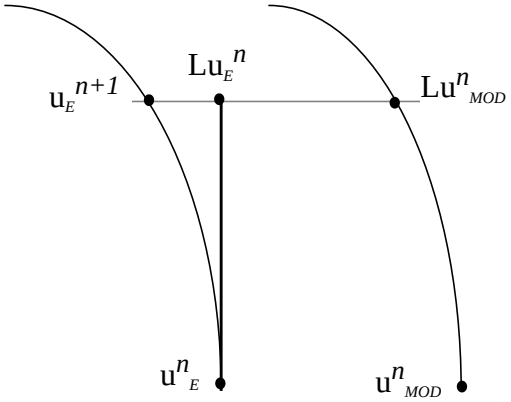


Figure 8. A qualitative picture of the evolution of the modified and numerical solutions..

Example: Upwind differencing The modified equation for upwind differencing is given as

$$\frac{\partial u_M}{\partial t} + a\frac{\partial u_M}{\partial x} = \frac{a\Delta x}{2}(1-\sigma)\frac{\partial^2 u_M}{\partial x^2}. \tag{112}$$

The effect of the numerical error in upwind differencing is, to leading order, to add a diffusion term to the PDE with a diffusion coefficient of magnitude $\frac{(1+\sigma)\,a\Delta x}{2}$. Second-order schemes are found to be dispersive in an analogous fashion.

1.7 Discontinuities

We would like to know whether we can design finite difference methods for the case where φ is a discontinuous function, i.e., a function that is continuous, except at a finite number of points, $x_1 \ldots x_p$ such that the following limits exist

$$\lim_{x \to x_p,\, x < x_p} \varphi(x) = \varphi_L(x_p) \neq \varphi_R(x_p) = \lim_{x \to x_p,\, x > x_p} \varphi(x)$$

For such initial data, the exact solution formula (xx) is still well-defined, even though the derivatives appearing in the differential equation are not. Consequently, we can take that formula as the definition of what we mean by a solution to the differential equation. We can construct the discretized initial data φ_j by evaluating φ at grid points, and taking $\varphi_j = \varphi_L(j\Delta x)$ at points for which the data is discontinuous. Using that discrete data, we can define both the Fourier interpolant $\varphi_S(x)$, the spectrally accurate approximate solution $u_S(x,t)$, and the numerical solution u_j^n. We consider first the convergence properties of the spectrally accurate solution. In that case, it is well-known (Dym and McKean) that Fourier interpolation / approximation of discontinuities does not converge uniformly, i.e., that $max_x|\varphi(x) - \varphi_S(x)|$ does not approach zero as $M \to \infty$. Instead, one obtains an overshoot / undershoot in the neighborhood of the discontinuity, with magnitude of approximately $\pm 14\%$. This failure of Fourier approximations to converge, and the nature of the lack of convergence, was first observed by J.W. Gibbs, and is known as Gibb's phenomenon; an example is shown in Figure 9.Thus, as the solution evolves in time, $u_S(x,t) = \varphi_S(x-at)$, and we will sample the Gibbs oscillations at the grid points, depending on the location of the grid points relative to the oscillations. In spite of this problem, the Gibbs' oscillations occupy a decreasingly narrow interval in the neighborhood of the discontinuity as $M \to \infty$, so that the solution converges in L^1 or L^2.

Overall, finite difference methods have the same convergence properties as spectral methods in the presence of discontinuities. The numerical solutions converge in L^1 or L^2, but not uniformly. However, the qualitative behavior of these methods varies considerably, depending on the method. Upwind differencing, as we have discussed above, does not introduce any new maxima or minima that were not already present in the solution.

Figure 9. Gibbs' phenomenon for 64 and 256 interpolation points. We plot the Fourier interpolant of the step function at ten times the number of interpolation points.

This is considerably different from the behavior of the spectrally accurate solution, which produces new extrema. Both Lax-Wendroff and Fromm also produce oscillations near discontinuities. One can think of these oscillations as coming from two sources. First, to the extent that the finite difference solution $u_I^n(x)$ approximates $u_S^n(x)$, we expect to see Gibbs oscillations in the solution (these do not appear in the upwind differencing results because the short-wavelength components that lead to these oscillations are sufficiently strongly damped). In addition, the amplitudes of the short-wavelength Fourier components are sufficiently large ($b_k \sim k^{-1}$ for k large) that errors in the locations of these Fourier modes, i.e., phase errors, can lead to oscillations that are larger that the Gibbs oscillations.

In Figures 10-12 we show several examples of the behavior of difference schemes in the presence of discontinuities. In all cases, we compare the results with the exact solution, as well as to the continuous Fourier interpolant. The upwind differencing represents the discontinuity as a smooth monotone transition; however, there are many mesh points in the transition. In fact, one can analytically estimate the width of the transition for this scheme; at time t, it is given by $O((t\Delta x)^{1/2})$, i.e. a transition region which is, for fixed t, $O(M^{1/2})$ mesh points wide. This is consistent with the model of the upwind difference

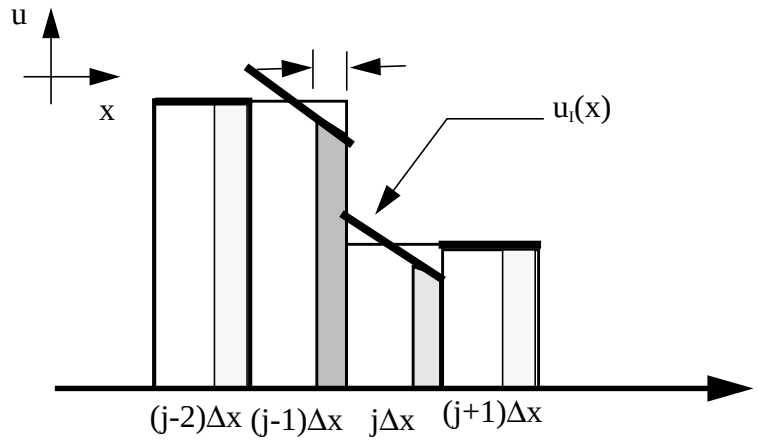
method that comes from the modified equation analysis that models the error in upwind differencing acting like an added diffusion term in the differential equation, with diffusion coefficient proportional to Δx. For both the Fromm and Lax-Wendroff schemes, we see oscillations whose amplitudes and spatial extent exceed those due to Gibbs' phenomenon in the spectrally accurate calculation. In addition, the qualitative character of the oscillation is consistent with the phase error properties of the schemes discussed previously. For both schemes, the phase error is a lagging phase error, causing short-wavelength to propagate a speeds less than the exact propagation speed. This leads to enhanced oscillatory behavior in back of the exact location of the shock; in addition, there is smoothing of the solution in front of the discontinuity due to the displacement of the short wavelengths that had initially represented the sharp transition.

Figure 10. Lax Wendroff, the Fourier interpolant and the exact solution on a grid with 128 mesh points. The Fourier interpolant is plotted as in Figure 9.

A somewhat different, and more spatially localized explanation of what oscillations occur can be obtained by looking in detail at what the various finite difference schemes produce

Figure 11. Fromm, the Fourier interpolant and the exact solution on a grid with 128 mesh points. The Fourier interpolant plotted as in Figure 9.

Figure 12. Upwind differencing and the exact solution on a grid of 128 mesh points.



(113)

Figure 13. Over / undershoot of the numerical solution near a discontinuity using Fromm's scheme

for discontinuous initial data at the end of one time step. In figure 13, we show the case of Fromm's scheme. The shaded regions represent the total mass crossing the various cell edges. In the $j-1$ cell, the amount of mass leaving the cell is smaller than the amount of mass coming from the right. since the average of u is already at the maximum value to the left of the discontinuity, this leads to an overshoot. Similarly, in the $j+1$ cell, the amount of mass exiting from the left is greater than the amount of mass entering from the right. Since the initial value of the average is equal to the minimum value to the right of the discontinuity, this leads to an undershoot. in that cell.

In Figure 14 we show the corresponding result for the Lax-Wendroff scheme. In this case, we observe only the an overshoot in the $j-1$ cell due to the lower value of the mass moving out across the right edge.

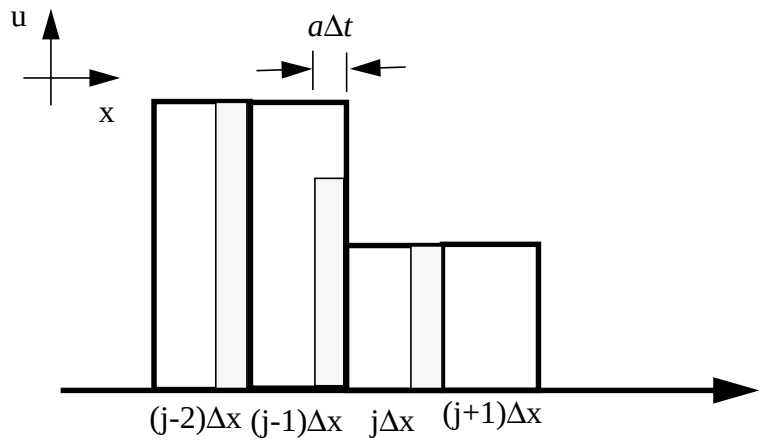


Figure 14. Overshoot of the numerical solution near a discontinuity using Lax-Wendroff

On the other hand, upwinding avoids this problem altogether.

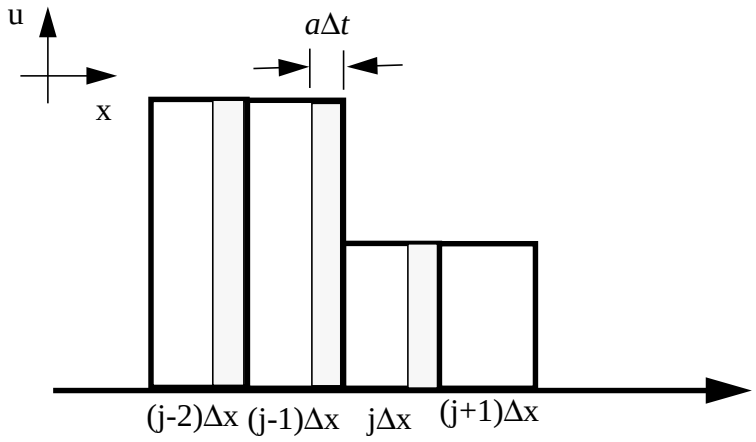


Figure 15. Behavior of numerical solution near a discontinuity using upwinding.

1.7.1 Max-norm boundedness and Godunov's Theorem

On physical grounds, it is often important that the solution to an advection equation of the form (1) not exceed the range of values defined by the initial data. If the solution represents the density of a fluid in a compressible flow calculation, the density must remain positive; otherwise, various quantities derived from the density may not be well-defined. Similarly, if the solution represents a concentration or mass fraction, it is important that the values remain between 0 and 1.

There are three ways of specifying this requirement. We say that a scheme is positivity preserving, if the $u_j^n \geq 0$ implies that $u_j^{n+1} \geq 0$. Second, we say that a scheme is monotonicity - preserving if $u_j^n \geq u_{j+1}^n$ for all j implies that $u_j^{n+1} \geq u_{j+1}^{n+1}$. finally, we say that a scheme is max-norm bounded if

$$\| Lu \|_\infty \leq \| u \|_\infty \tag{114}$$

where the max norm is defined by:

$$\| u \|_\infty = max_j (u_j) \tag{115}$$

For linear schemes, all three of these conditions are equivalent.

It is clear from the examples given above, that the Lax-Wendroff and Fromm schemes are not max-norm bounded, while upwind differencing is. On the other hand, the observed empirical accuracy of upwind differencing is much lower than that of the other two schemes, a reflection of the fact that the other two schemes are second-order accurate, while upwind differencing is first order accurate. Could it be that we have been insuffi-

ciently clever in constructing our methods, and that if we looked further, we could come up with a second-order accurate linear scheme which is also max-norm bounded? The answer, unfortunately, is no, as indicated by the following theorem.

Godunov's theorem: Let

$$(Lu)_j = \sum_s c_s u_{j+s} \tag{116}$$

be a linear scheme that is consistent and max - norm bounded and assume at least two of the coefficients c_s are nonzero. Then the scheme is first-order accurate.

.**Proof:** We shall prove this in two steps. First, we show that L being stable in the max norm is equivalent to saying that the coefficients of the scheme are nonnegative; i.e. $c_s \geq 0$ for all s.

Assume that this is *not* true, in other words that equation (116) holds for some c_s which is negative. For example, if:

$$u_s = \begin{cases} 1 \ldots (c_s > 0) \\ -1 \ldots (c_s < 0) \end{cases} \tag{117}$$

Then $u_j = 0$ if $j \in \{ s | (c_s = 0) \}$. Then, $\| u \|_\infty = 1$ and

$$(Lu)_0 = \sum c_s u_s = \sum |c_s| .$$

Since $\sum c_s = 1$ and $\exists s$ such that $c_s < 0$, this implies that:

$$\sum |c_s| > 1$$

and

$$\| u \|_\infty < \| Lu \|_\infty$$

which is a contradiction.

The second part of the proof is a simple calculation using the symbol. By (82)

$$|\lambda(\beta)|^2 = 1 + C_q \beta^q + O(\beta^{q+2}) \tag{118}$$

If the scheme is to be better than first order, then $q \geq 4$. For first order

$$|\lambda(\beta)|^2 = 1 + C_2 \beta^2 + O(\beta^4) \tag{119}$$

with $C_2 < 0$ for stability. This can be written as

$$|\lambda(\beta)|^2 = \sum_s c_s e^{i\beta s} \sum_{s'} c_{s'} e^{-i\beta s'}$$

Performing a Taylor expansion on the exponential, we obtain

$$|\lambda(\beta)|^2 = \sum_{s,\,s'} (c_s c_{s'})\ (1 - i\beta s - \frac{\beta^2}{2} s^2)\ (1 + i\beta s' - \frac{\beta^2}{2} s'^2) + O(\beta^4)$$

$$1 - \sum_{s,\,s'} \frac{(s - s')^2}{2} c_s c_{s'} \beta^2 + O(\beta^4).$$

If at least two of the c_s are nonzero, then the coefficient multiplying β^2 is nonzero, and the result is proven.

By consistency, in the form of equation (82), we have that $\sum c_s s = -\sigma$; consequently, any scheme that is consistent must have at least two nonzero c_s^s for σ not equal to an integer.Thus for almost all choices of σ, a linear finite difference scheme that is max-norm bounded will be first-order accurate. In addition, it will behave qualitatively like upwind differencing, in the sense that it will tend to smooth out discontinuities. This is easily seen, for example, by performing a modified equation analysis following that of equation (82).

So, how can we develop schemes that are higher-order accurate but still satisfy a maximum principle? The answer is to allow the schemes to be nonlinear, even for the linear problem (1); i.e., to be of the form

$$u^{n+1} = £u^n \tag{120}$$

where £ is a nonlinear operator. However, we pay a price in doing this, in that we lose our linear analysis machinery - equivalence of stability and boundedness, and the use of Fourier analysis. Our strategy will be to allow us to use the results from linear analysis “almost everywhere”, in the sense that, for regions where the solution is smooth, our nonlinear schemes will reduce to linear ones, and whose properties we understand. However, to talk about convergence for all parts of the solution, we need to develop some machinery.

1.8 Weak solutions, conservative finite difference methods and the Lax-Wendroff theorem.

Given initial data $U(x,0)$ for a general nonlinear system of equations:

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0 \tag{121}$$

where $U = U(x, t) \in \Re^n$ and $F = F(U(x, t)) \in \Re^n$, assume that a smooth solution exists over the domain. The engineering definition of a weak solution is that the solution

must be integrable. All of the methods (so far) satisfy the discrete analog of the weak solution. By the divergence theorem,

$$\int_{x_L}^{x_R} U(x, t^N)\, dx = \int_{x_L}^{x_R} U(x, t^0)\, dx + \int_{t^0}^{t^N} F(U(x_L, t))\, dt + \int_{t^0}^{t^N} F(U(x_R, t))\, dt \qquad (122)$$

For any x_L, x_R, t^O, t^N. We can take this to be the definition of a solution, even in the case when the derivatives in (110) are not defined. Incidentally, this is just an expression in mathematical terms, of the familiar control-volume formulation of continuum mechanics. We can think of the integral of U over an interval in space as being the total amount of conserved quantity at some time - a total "mass", in some generalized sense. Then (111) can be interpreted as the usual flux-balance relation over a control volume: mass at the new time is equal to the mass at the old time plus the net fluxes on the left and right edges.

A finite-difference method is in *conservation form* if it can be written as,

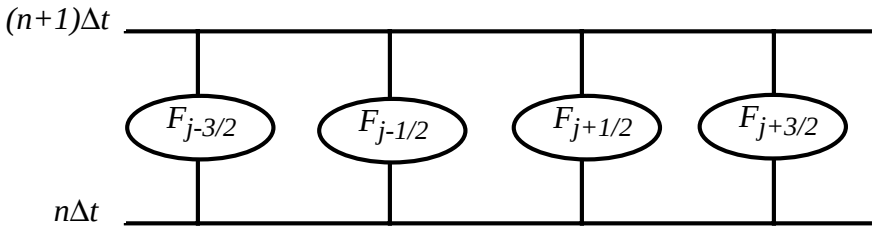


Figure 16. Flux through three cells

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x}\left(F_{j-\frac{1}{2}} - F_{j+\frac{1}{2}}\right)$$

where F represents the numerical fluxes at the cell boundaries. The weak consistency condition says that if we express the flux as

$$F_{j+\frac{1}{2}} = \Psi(U_{j-s}^n, \ldots, U_{j+s}^n)$$

and if we stuff a constant value U_o into Ψ, i.e., $\Psi(U_o, \ldots, U_o) = F(U_o)$. The discrete analogue of equation (111) above is

$$\sum_{j=j_L}^{j_R} U_j^{n+1}\Delta x = \sum_{j=j_L}^{j_R} U_j^n \Delta x + \Delta t \sum_{j=j_L}^{j_R}\left(F_{j-\frac{1}{2}} - F_{j+\frac{1}{2}}\right) \qquad (123)$$

At any *j*th cell, the flux coming from the *j-1*th cell is equal to the flux leaving the *j-1*th cell. Consider a domain which is three cells wide as below. The rate of change of the mass of in these three cells is given by

$$\left(F_{j-\frac{3}{2}} - F_{j-\frac{1}{2}}\right) + \left(F_{j-\frac{1}{2}} - F_{j+\frac{1}{2}}\right) + \left(F_{j+\frac{1}{2}} - F_{j+\frac{3}{2}}\right) = F_{j-\frac{3}{2}} - F_{j+\frac{3}{2}}$$

Notice that all the terms on the right-hand side cancel except for the first and last terms. This implies that the only fluxes which count are those at the edges of the domain. More generally, we have the following mass balance relation:

$$\Delta x \sum_{j=j_L}^{j_R} U_j^{N_n} = \Delta x \sum_{j=j_L}^{j_R} U_j^{N_o} + \Delta t \sum_{N_O}^{N_N - 1} F_{j_L - 1/2} - \Delta t \sum_{N_O}^{N_N - 1} F_{j_R + 1/} \tag{124}$$

This is a consistent discretization of the integral relation (111) defining weak solution. In particular, this enables us to compute discontinuous solutions. In figure 17 below, we show the exact solution to a discontinuous initial value problem plotted with the solution obtained by a conservative finite difference scheme. The fact that the scheme is conservative implies that the integral under the numerical solution curve is identical to that for the exact solution.

Figure 17. Conservative finite difference calculation of a discontinuity versus the exact solution.

The Lax-Wendroff theorem: Assume that the finite-difference scheme under consideration is consistent and in conservation form. If Ujn converges to some function, say U∞(x,t), in some norm such as L^1 or L^2, then U is a weak solution to

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0 \tag{125}$$

Note that this is *not* equivalent to saying that if the scheme is consistent and in conservation form, then it will converge; Lax-Wendroff doesn't guarantee existence of a weak solution.

1.9 Limiters

Consider discontinuous solutions to the hyperbolic equation:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{126}$$

Reasonable design criteria for a numerical scheme which solves this equation would need to include consistency and conservation. The discretized equation could be written in the form:

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{\Delta x}\left(F_{j-\frac{1}{2}} - F_{j+\frac{1}{2}}\right) \tag{127}$$

where $F_{j+\frac{1}{2}} = au_{j+\frac{1}{2}}$. The consistency condition indicates that:

$$\Phi(u_{j-s}, \ldots, u_{j+s}) = F_{j+\frac{1}{2}}$$

$$\Phi(u_0, \ldots, u_0) = au_0$$

We will attempt to obtained the properties of second- order accuracy for smooth solutions and positivity preservation by hybridizing first order and second order schemes. Given a second order scheme with scheme with flux F^H and a first-order scheme with flux F^L, we can take a linear combination of the two fluxes to obtain a hybrid scheme, as follows.

$$F_{j+\frac{1}{2}} = \alpha_{j+1/2}F^H_{j+\frac{1}{2}} + (1-\alpha_{j+1/2})F^L_{j+\frac{1}{2}}$$

$$= F^L_{j+\frac{1}{2}} + \alpha_{j+1/2}\left(F^H_{j+\frac{1}{2}} - F^L_{j+\frac{1}{2}}\right)$$

Here, $0 \le \alpha \le 1$. The high-order flux comes from our arsenal of linear difference schemes with order greater than two, while the low-order flux satisfies the maximum principle. At the discrete level, this looks like having an exact and a diffusive flux. The goal of linear

hybridization is to keep the solution from straying outside some maximum/minimum bounds. Godunov's theorem shows that $\alpha=\alpha(u)$ for the hybridized scheme to be second-order accurate in regions where the solution is smooth; in such regions, we would expect $\alpha = 1$

1.9.1 Flux-corrected transport

Flux-corrected transport was an early version of this approach. The initial work was done by Boris and Book in the early 1970's. Zalesak (1979) has a good discussion of this method. The basic idea is to choose α so that no overshoot or undershoot can occur. Since we don't want *any* high-frequency amplification (a nontrivial requirement), we require F^H to be Fourier stable. Since notation is free, let's denote the anti-diffusive flux as follows.

$$A_{j+\frac{1}{2}} = F^H_{j+\frac{1}{2}} - F^L_{j-\frac{1}{2}} \tag{128}$$

The quantity $A_{j+1/2}$, to leading order in Δx, looks like a flux for a backwards diffusion term, in the sense of a modified equation analysis; hence its name. We also define the transported and diffused velocity by:

$$u^{TD} = u_j^n + \frac{\Delta t}{\Delta x}\left(F^L_{j-\frac{1}{2}} - F^L_{j+\frac{1}{2}}\right) \tag{129}$$

In other words, in order to obtain u^{TD}, simply evolve u for one time step using the lower-order scheme. Finally, we define the corrected flux A^c as follows.

$$A^c_{j+\frac{1}{2}} = \alpha_{j+1/2}\left(F^h_{j+\frac{1}{2}} - F^l_{j-\frac{1}{2}}\right) \tag{130}$$

With all of these new definitions in place, the discrete evolution equation for u is given by

$$u_j^{n+1} = u^{TD} + \frac{\Delta t}{\Delta x}\left(A^c_{j-\frac{1}{2}} - A^c_{j+\frac{1}{2}}\right) \tag{131}$$

The strategy for computing the hybridization parameter $\alpha_{j+1/2}$ or, equivalently, $A^c_{j+1/2}$, is based on the following ideas. First, we want to compute the corrected flux so that it as nearly equal to $A_{j+1/2}$ as possible, subject to the constraint that the outcome of applying (130) will not exceed the range of values defined by nearby values of u^{TD}. Thus, if the low-order scheme satisfies a maximum principle, then the hybrid scheme will do so, as well. In addition, if we choose a sufficiently large window over which to define the range of minimum and maximum values of u^{TD}, the scheme should reduce to the linear high-order scheme in regions where the solution is sufficiently smooth to be resolved on the grid.

Finally, one would like an algorithm that looks at each edge independently from the others. The problem with this last criterion is that it is possible for two fluxes, acting together, to cause a value to exceed the max and min limits, although each flux, acting separately, might remain in bounds. The solution to this problem obtained by Boris and Book is to use the intuition that $A_{j+1/2}$ is an antidiffusive flux, so that it has the same sign as the local gradient.

We illustrate this idea in the following figure. If the antidiffusive fluxes are all in the direction of the local solution gradient, as defined by u^{TD}, the two antidiffusive fluxes will have the same sign, and for that reason partially cancel.

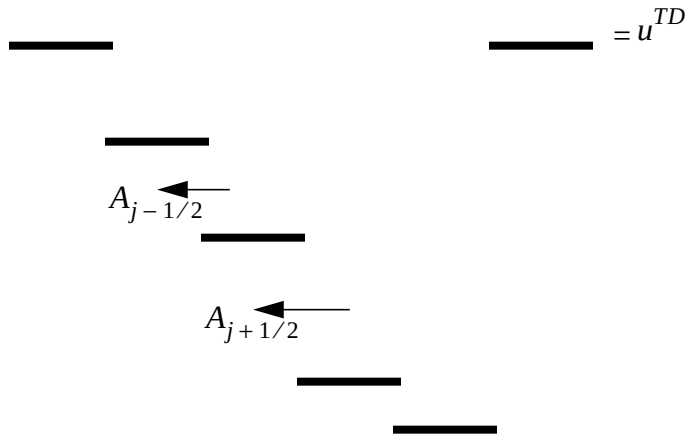


Figure 18. Partial cancellation of the antidiffusive fluxes at adjacent cell edges.

On the other hand, if any antidiffusive flux fails to satisfy this condition, the corresponding corrected flux will be set to zero.

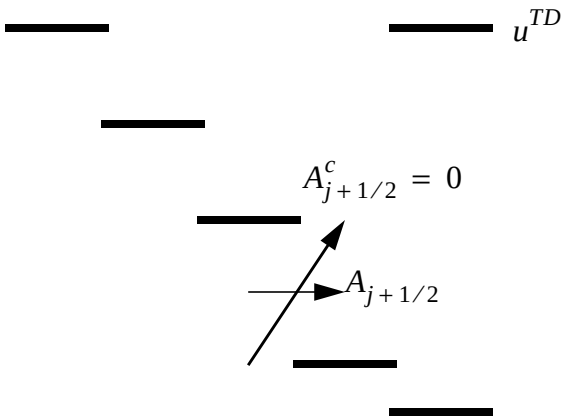


Figure 19. Cancellation of a corrected flux that is not antidiffusive.

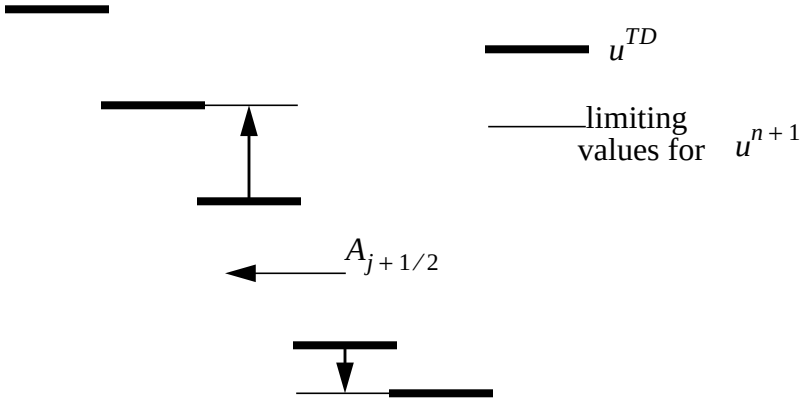


Figure 20. Limiting of the corrected antidiffusive flux.

The combination of these two conditions allows us to limit each corrected flux separately, since the extent to which it will cause the solution to exceed the range of values defined by u^{TD} will be bounded by the extent to which that will occur with the fluxes at the other edges set to zero. Specifically, we will take the corrected flux to be as nearly equal to $A_{j+1/2}$, subject to the constraint that its use in (131) with the fluxes set to zero at the other

edges does not allow u_j^{n+1} to go outside the range defined by u_{j-1}^{TD}, u_j^{TD}, nor u_{j+1}^{n+1} to go outside the range defined by u_{j+1}^{TD}, u_{j+2}^{TD} (Figure 20). Algebraically, this constraint is given as follows.:

$$\left|A^c_{j+\frac{1}{2}}\right| = min\left(\left|\frac{\Delta x}{\Delta t}(u_j^{TD} - u_{j-1}^{TD})\right|, \left|\frac{\Delta x}{\Delta t}(u_{j+2}^{TD} - u_{j+1}^{TD})\right|, |A_{j+1/2}|\right) \tag{132}$$

Combining these two conditions, and accounting for all the possible combinations of signs, the corrected flux can be written as follows.

$$A^c_{j+\frac{1}{2}} = S_{j+\frac{1}{2}} max(0, K) \tag{133}$$

where

$$K = min\left(\left|A_{j+\frac{1}{2}}\right|, S_{j+\frac{1}{2}}\frac{\Delta x}{\Delta t}(u_{j+2}^{TD} - u_{j+1}^{TD}), S_{j+\frac{1}{2}}\frac{\Delta x}{\Delta t}(u_j^{TD} - u_{j-1}^{TD})\right) \tag{134}$$

and

$$S_{j+\frac{1}{2}} = sign\left(A_{j+\frac{1}{2}}\right) \tag{135}$$

1.9.2 Geometric limiters

The basic idea of geometric limiters is to first apply a limiter to an interpolation function $u_l(x)$, and then use an upwind method to calculate the fluxes and advance the solution. One example is van Leer's scheme, which combines Fromm's method with a geometric limiter. Van Leer's scheme adjusts the slopes; i.e., the Δu_j's, so that the interpolation function $u_l((j+1/2)\Delta x)$ stays in the range defined by u_j^n and $u_{j\pm1}^n$ as shown in Figure 21.

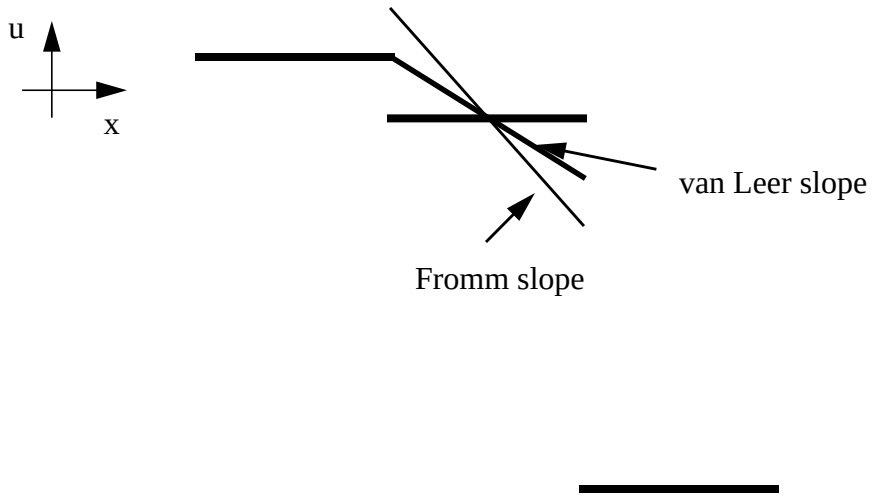


Figure 21. An example of slope limiting.

. We can write the flux as:

$$F_{j+\frac{1}{2}} = a u_{j+\frac{1}{2}} \tag{136}$$

where

$$u_{j+\frac{1}{2}} = u_j^n + (1-\sigma)\,\Delta u_j^{VL} \tag{137}$$

In the above equation, σ is the CFL number and the van Leer slopes are:

$$\Delta^{VL} u_j = \begin{cases} S_j \bullet min\,(2|u_{j+1}-u_j|,\, 2|u_j-u_{j-1}|,\, \frac{1}{2}|u_{j+1}-u_{j-1}|)\ \ldots if \ldots \varphi > 0 \\ 0 \ldots otherwise \end{cases} \tag{138}$$

where S_j now denotes the sign of the gradient

$$S_j = sign\,(u_{j+1}^n - u_{j-1}^n) \tag{139}$$

and the quantity φ is given by

$$\varphi = (u_{j+1}^n - u_j^n) \bullet (u_j^n - u_{j-1}^n) \tag{140}$$

This scheme “naturally” detects a discontinuity and modifies its behavior accordingly. The implications of this are that this method retains the high-order accuracy of Fromm’s scheme in smooth regions of the solution, but where discontinuities are detected, the discretized evolution equation drops to first-order accuracy. (This is a reasonable thing to do since you probably don’t have enough derivatives in this region, anyways.)

We note finally that the van Leer scheme is actually a linear hybridization of Fromm's scheme and upwinding with a special choice of the hybridization parameter α. Consider

$$F_{j+\frac{1}{2}} = a u_{j+\frac{1}{2}}$$

where the cell edge velocity is defined for van Leer's scheme as

$$u_{j+\frac{1}{2}} = u_j + \frac{1}{2}(1-\sigma)\,\Delta^{VL} u_j \tag{141}$$

The van Leer slope $\Delta^{VL} u_j$ was defined previously in equation (127). For Fromm's scheme, the analogous quantity is

$$u_{j+\frac{1}{2}} = u_j + \frac{1}{2}(1-\sigma)\,\Delta^{c} u_j \tag{142}$$

where the slope is defined as simply

$$\Delta^{c} u_j = \frac{1}{2}(u_{j+1} - u_{j-1}) \tag{143}$$

To conform to the notation adopted at the beginning of this section on limiters, if we consider $F^{H}_{j+\frac{1}{2}} \leftrightarrow F^{Fromm}$ and $F^{L}_{j+\frac{1}{2}} \leftrightarrow a u_j^n$, then we can write the van Leer flux as,

$$\begin{aligned} F^{VL} &= F^{L} + \alpha\,(F^{Fromm} - F^{L}) \\ &= a u_j^n + a\frac{1}{2}(1-\sigma)\,\Delta^{VL} u_j \\ &= a u_j^n + a\frac{1}{2}(1-\sigma)\,\Delta^{C} u_j \left[\frac{\Delta^{VL} u_j}{\Delta^{C} u_j}\right] \end{aligned}$$

If we choose α to be the term in brackets above, then we can write F^{VL} as:

$$F^{VL} = a u_j^n + \alpha\,(F^{Fromm} - F^{L}) \tag{144}$$

1.9.3 Design criteria for schemes with limiters

Thus far, we have identified a general set of techniques for constructing schemes with limiters. The ingredients are:

(i) A linear low-order scheme that satisfies a maximum principle.

(ii) A linear high-order scheme that is linearly stable.

(iii) A hybridization switch between the two, designed so that it the resulting hybrid scheme introduces no new local extrema, and reduces to the high-order scheme for smooth solutions. In order for this last condition to be satisfied, the hybridization coefficient $\alpha_{j+1/2}$ must depend on the solution values.

The way we have accomplished the third criterion is to construct the hybridization switch so that it blends in, by some measure, the minimum amount of the low-order scheme necessary to prevent new extrema from forming. In the FCT case, the measure was that the solution must remain within bounds set by the transported and diffused solution. For the van Leer scheme, the interpolation profile is constrained so that, independent of the CFL number σ, the overall scheme would general no new extrema. On of the consequences of this kind of construction is that, after a few time steps, the discrete profile representing a discontinuity approaches something very close to a discrete travelling wave solution, with the number of mesh points in the transition across the discontinuity increasing only very slowly as a function of time. This is in contrast with the first-order linear schemes, for which the number of mesh points in the transition representing a discontinuity increases like $O(n^{1/2})$, where n is the number of time steps.

Next, we would like to make the argument that the accuracy with which hybrid schemes of the form discussed above represent discontinuities is intimately tied to the phase error properties of the scheme. In section 1.7, we saw that the effect of the leading-order dispersive errors of second order schemes was to cause a lagging or leading oscillatory wave train (depending on the sign of the phase error) due to the errors in the transport of the large wavenumber Fourier modes. In addition, the extraction of the high-wavenumber modes from one side of the discontinuity caused the side opposite that of the oscillations to be smoothed out. Overall, the lower the phase error of the scheme, the narrower this oscillatory / smooth structure representing the discontinuity was.

When we combine these second-order schemes with limiters, we find that the interaction of the phase error properties of the high-order scheme with the limiter dominate the accuracy with which discontinuities are represented on the grid. Roughly speaking, in the region in which the linear high-order scheme would otherwise be oscillating, the solution is chopped off to be flat, due to the influence of the limiter. However, that high-wavenumber component of the solution is still being extracted from the side opposite the oscillatory region, leading to a smoothed out profile. Thus to a high degree of accuracy, the effect of the limiter is to replace the oscillatory region near a discontinuity with a flat profile, while

leaving the smoothed-out portion of the solution from the second-order linear scheme unchanged. In Figure 22 we show comparisons between the Lax-Wendroff and FCT / Lax-Wendroff schemes, and between the Fromm and van Leer schemes. The solution profiles support this scenario for the interaction of the phase error of the high-order scheme with the treatment of discontinuities. Finally, if we compare the FCT/ Lax-Wendroff and van Leer schemes for this run, we find that the number of mesh points in the van Leer representation of the discontinuities is about half that in the FCT / Lax Wendroff representation of the discontinuity. Thus, even in the case of hybrid schemes and discontinuous solutions, the lower phase error of the Fromm scheme is worth about a mesh refinement over the Lax-Wendroff scheme.

Figure 22. Fromm / van Leer comparison (top), Lax-Wendroff / FCT comparison (bottom) on a grid with 128 mesh points, CFL = 0.75 and time t = 5.

2. Nonlinear scalar problems

Consider the nonlinear PDE for the scalar function *u:*

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \tag{1}$$

with initial data $\phi(x)$,

$$u(x, 0) = \phi(x) \tag{2}$$

In what follows we will assume that *f(u)* is a convex function of *u*; i.e., we assume that

$$\frac{\partial^2 f}{\partial u^2} > 0$$

As an example we consider Burger's equation for which

$$f(u) = \frac{1}{2}u^2$$

Let the wavespeed *a* be defined by $a(u) = \frac{\partial f}{\partial u}$. Note that our convexity assume implies that $\frac{\partial a}{\partial u} > 0$. Assuming that *u(x,t)* is differentiable, equation (1) can be rewritten in the quasilinear form:

$$\frac{\partial u}{\partial t} + a(u)\frac{\partial u}{\partial x} = 0 \tag{3}$$

Now consider solutions *x(t)* to the ODE

$$\frac{\partial x}{\partial t} = a(u(x, t), t) \tag{4}$$

(5)

By differentiating *u(x(t),t)* with respect to*t* one can see that *u(x(t),t) = u(x-at,0) =* $\phi(x_o - at)$ is a solution of (3). The function *x(t)* is called a characteristic and the ODE (4)-(5) is called a characteristic equation. The characteristic can be thought of as either a trajectory in space or as a curve in space/time along which information (the solution *u*) propagates.- Let's examine what happens to *u* along characteristics.Using the chain rule, we find that the material derivative of *u(x(t),t)* is:

$$\frac{d}{dt}(u(x(t), t)) = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}\frac{\partial x}{\partial t}$$

$$\frac{d}{dt}(u(x(t), t)) = \frac{\partial u}{\partial t} + a(u(x, t), t)\frac{\partial u}{\partial x} \tag{6}$$

However, the differential equation (3) tells us that the r.h.s. of equation (4) vanishes. Therefore, the derivative of u is zero along a characteristic. This implies that u is constant along characteristics.

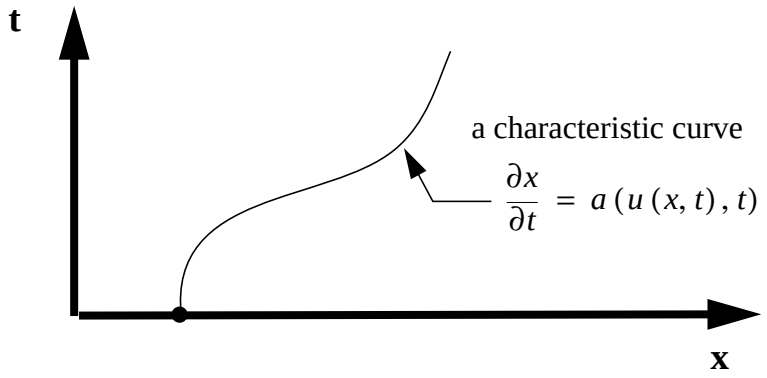


Figure 23. A characteristic curve in space-time.

Now consider how the wavespeed a for Burgers equation (1) changes along a characteristics. We have

$$\frac{dx}{dt} = a(u(x,t),t) = a(u(x_o,0))$$

$$\frac{dx}{dt} = \phi(x_o) \tag{7}$$

since $u(x,t) = u(x_o,0)$ on the characteristic. Thus, for Burgers equation, the characteristics in space/time are straight lines with slope $\phi(x_o)$, and the information which propagates along the characteristics is $u(x_o,0)$. The difference between the linear and nonlinear cases is that the slopes of the characteristics are not necessarily constant for the latter case.

For example, it is apparent from Figure 24 that if the lines are extended further in time then eventually they will cross. Thus, if at time $t = 0$, there are some values of $u(x,0) = \phi(x)$ are positive and some are negative, or more generally if $\phi(x_1) > \phi(x_2)$ for some $x_1 < x_2$, then they will cross at some finite time. In general,whenever the initial data $\phi(x)$ is smooth and has compact support (i.e., $\phi(x) = 0$ outside of some finite interval), then there will be some critical time T_c at which the solution u is no longer single valued. At T_c is the value of $a(\phi)$ equivalent to the value carried with the left-running characteristic, the right-running characteristic, or both?

.

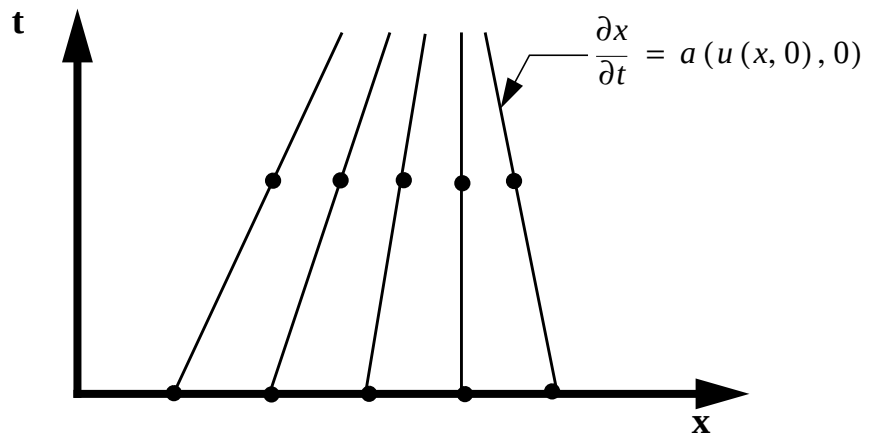


Figure 24Characteristic curves for the nonlinear scalar problem

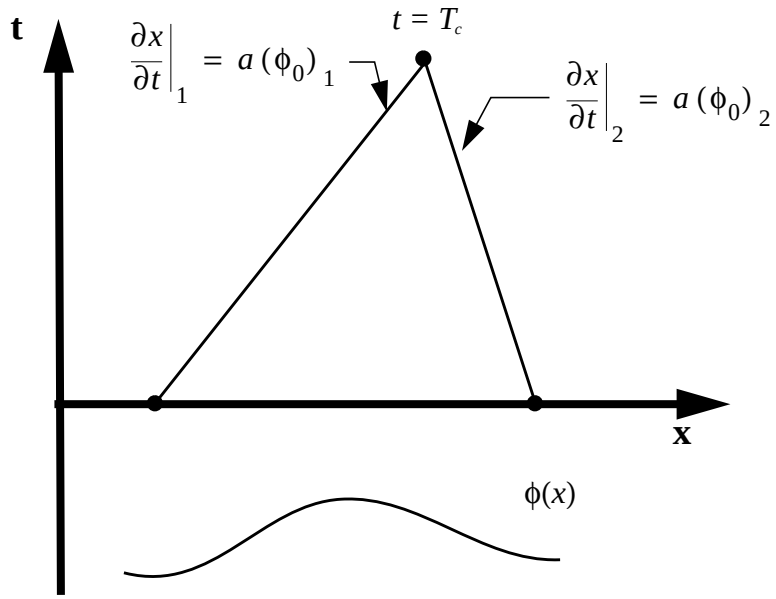


Figure 25Characteristics collide at time t= T_c

To examine this more closely, consider the behavior of *u* in space. We know that the exact solution at any time *t* is given by:

$$u(x,t) = \phi(x - a(u(x,t))\,t) \tag{8}$$

The partial derivative of *u(x(t),t)* with respect to *x* is:

$$\frac{\partial u}{\partial x} = \phi' - t\frac{\partial a(u(x,t))}{\partial x}\phi'$$

where the prime indicates differentiation with respect to x. Next use the chain rule to obtain

$$\frac{\partial u}{\partial x} = \phi' - t\frac{da}{du}\frac{\partial u}{\partial x}\phi'$$

or

$$\frac{\partial u}{\partial x} = \frac{\phi'}{1 + t\frac{da}{du}\phi'} \tag{9}$$

If $\phi' < 0$ anywhere, then the spacial derivative of u blows up, since the denominator in equation (8) goes to zero at time $t = T_c$ where

$$T_c = -\frac{1}{\frac{da}{du}\phi'} \tag{10}$$

This analysis might klead on to believe that the solution cannot be continued beyond the time $t = T_c$. However, from a physical standpoint, we know that there exist solutions to nonlinear hyperbolic equations beyond the critical time. In order to understand how one can extend the solutions to times $t > T_c$ we need to introduce the concept of weak solutions.

2.1 Weak solutions of nonlinear hyperbolic problems

A weak solution to the nonlinear scalar equation (1) satisfies

$$\int_{x_L}^{x_R} u(x, t^N)\,dx = \int_{x_L}^{x_R} u(x, t^0)\,dx + \int_{t^0}^{t^N} f(u(x_L, t))\,dt - \int_{t^0}^{t^N} f(u(x_R, t))\,dt \tag{11}$$

between initial time t^0 and a later time t^N and between the left and right boundaries of the physical domain, x_L and x_R respectively. Consider the particular case of a discontinuous solution, specifically, an isolated discontinuity propagating at speed s. From Figure 4, we can evaluate the integrals:

$$\int_{x_L}^{x_R} (u(x, t^N) - u(x, t^0))\,dx = u_L(\lambda_L - s(t^N - t^0)) + u_R(\lambda_R - s(t^N - t^0)) - (u_L\lambda_L + u_L\lambda_L)$$

$$\int_{x_L}^{x_R} (u(x, t^N) - u(x, t^0))\,dx = (u_L - u_R)\,s\,(t^N - t^0) \tag{12}$$

. and similarly

$$\int_{t^0}^{t^N} (f(u(x_L, t)) - f(u(x_R, t))) \, dt = (f(u_L) - f(u_R)) \, (t^N - t^0) \tag{13}$$

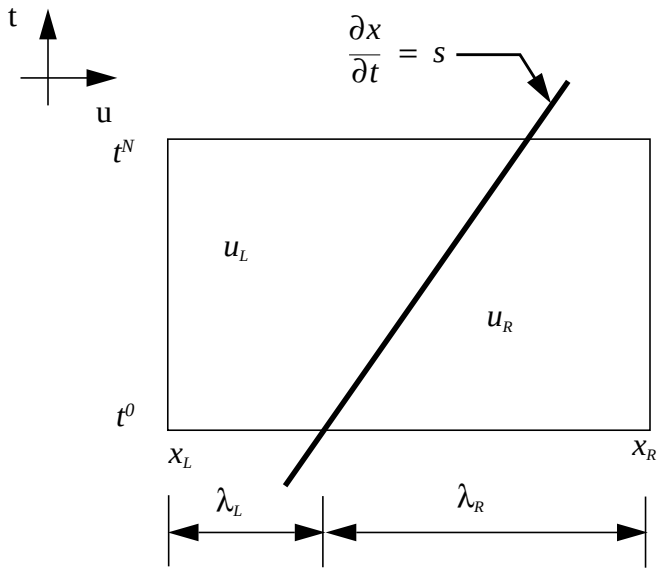


Figure 26. A discontinuity propagating in space-time with speed s.

Using equations (11) and (12) above, we can solve for the propagation speed which the discontinuity is traveling at,

$$s = \frac{f(u_L) - f(u_R)}{u_L - u_R} \tag{14}$$

This is the known as the Rankine-Hugoniot jump relation. If f is a smooth function, then by the mean value theorem

$$\frac{f(u_L) - f(u_R)}{u_L - u_R} = \left.\frac{\partial f}{\partial u}\right|_{\xi} \tag{15}$$

where ξ is some intermediate value between u_L and u_R. Thus we see that as $u_L \to u_R$, then

$$s \to \left.\frac{df}{du}\right|_{\frac{u_L + u_R}{2}} = a\left(\frac{u_L + u_R}{2}\right) \tag{16}$$

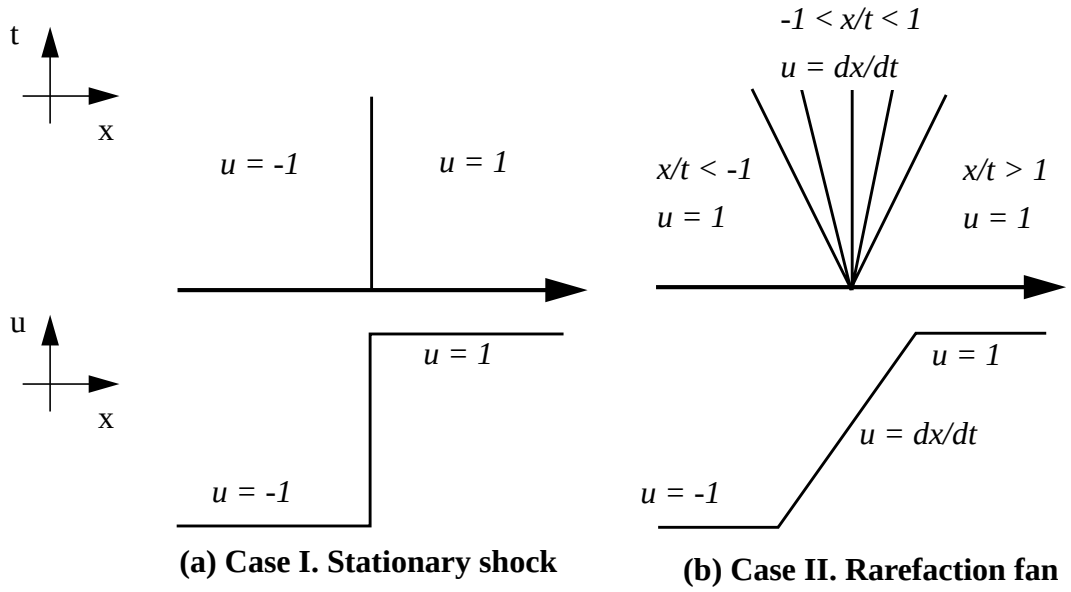
This indicates that weak waves are propagated along characteristics.

2.1.1 Nonuniqueness of weak solutions

One problem with weak solutions is that they are not unique, i.e., for the same initial data, we can obtain more than one weak solution. For example, consider Burgers equation for which $f(u) = \frac{1}{2}u^2$ with the following initial data:

$$u(x, 0) = \begin{cases} -1 \ldots (x < 0) \\ 1 \ldots (x > 0) \end{cases}$$

The trivial solution is *s = 0*, shown in figure 5a. Another weak solution in which characteristics fan out from the point *x = 0, t = 0* is presented in Figure 27bt



(a) Case I. Stationary shock **(b) Case II. Rarefaction fan**

Figure 27. Two weak solutions of Burgers equation: (a) a stationary shock; (b) a rarefaction fan.

2.1.2 The entropy condition

We are thus left with the dilemma of deciding how to choose the physically correct solution. The answer to this conundrum is to require that the 'correct' weak solution be the limit of solutions to the associated viscous equation in the limit as the viscosity tends to zero. In other words, let u^ε be a solution of

$$\frac{\partial u^\varepsilon}{\partial t} + \frac{\partial f(u^\varepsilon)}{\partial x} = \varepsilon \frac{\partial^2 u^\varepsilon}{\partial x^2} \tag{17}$$

where ε is the viscosity coefficient. We require that *u* satisfy

$$\lim_{\varepsilon \to 0} u^\varepsilon = u \tag{18}$$

We claim that a weak solution which satisfies (17) leads to a well-posed problem. In other words, the solution exists, is unique, and has continuous dependence on initial data. Solutions which satisfy (17) are said to satisfy the entropy condition. This terminology comes from the fact that entropy can only increase across a gas dynamic shock and that (17) is one mechanism we can use to identify the solution to which has this property. These solutions are often referred to as 'entropy solutions'.

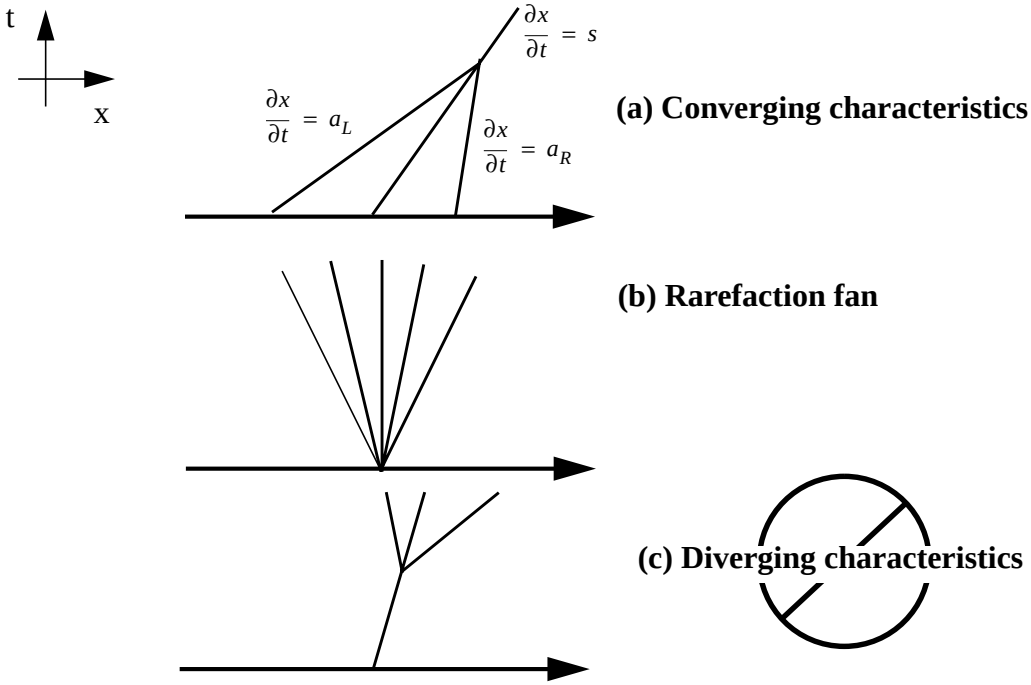


Figure 28. Weak solutions to Burgers equation which satisfy the entropy condition: (a) Converging characteristics; and (b) Rarefaction fan; and a weak solution which violates the entropy condition: (c) Diverging characteristics

There is an equivalent criterion that one can use to select the physically correct weak solutio which turns out to be easier to apply in practice. This is a condition that must be satisfied at discontinuities in the weak solution. Recall that the speed s of a discontinuity is given by

$$s = \frac{f(u_L) - f(u_R)}{u_L - u_R} \tag{19}$$

where L and R denote left and right sides of the wave. For convex flux functions *f; i.e.*, $\frac{\partial^2 f}{\partial u^2} > 0$, the entropy condition is satisfied if:

$$a(u_L) > s > a(u_R) \tag{20}$$

(For Burgers equation, recall that $a(u) = u$.) Geometrically, this means that the information must propagate forward in time as shown in Figure 26.

2.2 Strategies to enforce the entropy condition

To, in order to develop an adequate numerical scheme to solve the nonlinear scalar equation (1), we require that the numerical solution:

(i) be a weak solution of the problem;

(ii) be reasonably accurate in smooth regions using the design criteria for linear advection;

(iii) be stable, where we place more stringent requirements on the stability of the scheme for the case of nonlinear discontinuities (e.g., LW4 is stable for purely linear problems, but it spreads out the Fourier components of the wave since they travel at different speeds. This may cause problems such as mode-mode coupling in nonlinear problems);

(iv) satisfy the entropy condition.

Let's examine what happens when the entropy condition is not satisfied. Consider the simple case of upwind differencing. The discretized form of the governing equation in conservation form is:

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{\Delta x}\left(F^n_{j-\frac{1}{2}} - F^n_{j+\frac{1}{2}}\right) \tag{21}$$

where

$$F_{j+\frac{1}{2}} = \begin{cases} f(u_j) \ldots if \ldots a\left(\frac{u_j + u_{j+1}}{2}\right) > 0 \\ f(u_{j+1}) \ldots if \ldots a\left(\frac{u_j + u_{j+1}}{2}\right) \le 0 \end{cases} \tag{22}$$

Desirable properties of this scheme include the facts that it is stable and total-variation-diminishing; however, it does not satisfy the entropy condition. To see this suppose we have the initial data at $t = 0$:

$$u_j^0 = \begin{cases} -1 \ldots j < 0 \\ 1 \ldots j \ge 0 \end{cases}$$

Since this particular discontinuity violates the entropy condition, we would like the numerical simulation to dissipate the discontinuity with time. For Burgers equation, $f(u_j) = u_j^2/2$ and $a(u_j) = \partial f/\partial u = u_j$ and a simple calculation shows that all the

fluxes cancel each other, and the solution reproduces itself at each time step. As Δx and Δt go to zero, we would converge to an entropy-violating shock using upwind differencing.

2.2.1 Artificial viscosity

There are a number of ways to cope with this problem. One solution is to add in a small amount of artificial viscosity, ε. The error in smooth regions is of order ε, while it is order $\sqrt{\varepsilon}$ at discontinuities. In general, if one chooses $\varepsilon = O(\Delta x)$ everywhere this will work. However, this will negate all of the careful improvements we have made in designing a high resolution numerical method. Therefore, to preserve the low phase error and other desireable properties of the methods discussed in Chapter I, we would like $\varepsilon = O(\Delta x)$ at the discontinuities, and $\varepsilon = O(\Delta x^p)$ in smooth regions, where p is the order of the scheme. The truncation error for upwinding is:

$$\frac{\partial u_{MOD}}{\partial t} + \frac{\partial f(u_{MOD})}{\partial x} = a(u_{MOD})\left(\frac{1-\sigma}{2}\right)\Delta x \frac{\partial^2 u_{MOD}}{\partial x^2} \tag{23}$$

2.2.2 The first-order Godunov method

Another strategy is to utilize Godunov's method,which is a "real" geometric upwinding scheme. The solution process is broken down into three steps:

(i) Interpolate $u_I(x)$ from u_j^n. Choose a piecewise linear interpolation function so that $u_I(x) = u_j^n$ for $(j-\frac{1}{2})\Delta x < x < (j+\frac{1}{2})\Delta x$. (See Figure 2 in Chapter 1);

(ii) Solve the above equation exactly; i.e., solve the Riemann problem at each discontinuity, with the left and right states given by u_j and u_{j+1} respectively. This is always solvable because of our assumption that $\partial^2 f/\partial u^2 > 0$ and $da/du > 0$. We will obtain compression waves or expansion fans as shown by Figure 6a and b above;

(iii) Average the results back on to the grid.

Step (ii) above ensures that this procedure produces a solution which satisfies the entropy condition, while step (iii) adds artificial viscosity. We will later prove that if we start out with something which satisfies the entropy condition and average it, the result will still satisfy the entropy condition. Each piece of the solution satisfies the entropy condition by itself and is a weak solution to the governing equation.

We now describe this procedure in more detail. We wish to solve the nonlinear scalar problem:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \tag{24}$$

The first-order Godunov finite-difference method for (24) is given by:

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{\Delta x}\left(F_{j-\frac{1}{2}} - F_{j+\frac{1}{2}}\right) \tag{25}$$

where the flux is defined by

$$F_{j+\frac{1}{2}} = f\left(\hat{u}_{j+\frac{1}{2}}\right) \tag{26}$$

and $\hat{u}_{j+\frac{1}{2}}$ is the solution to the Riemann problem; i.e., the solution of (24) with initial data

(27)

evaluated along the ray $x/t = 0$. This satisfies the entropy condition since by definition the exact solution ofthe Riemann problem is the entropy solution.

Let's compare this method to classical upwinding. The flux for the latter is:

$$F_{j+\frac{1}{2}}^{upwind} = \begin{cases} f(u_j) \ldots if \ldots \left(\frac{a(u_j) + a(u_{j+1})}{2} > 0\right) \\ f(u_{j+1}) \ldots if \ldots \left(\frac{a(u_j) + a(u_{j+1})}{2} < 0\right) \end{cases} \tag{28}$$

For cases in which the entropy condition is not violated, the result reduces to the same answer as upwinding would give. If $a(u_j) > 0 > a(u_{j+1})$, the Riemann problem would choose

$$F_{j+\frac{1}{2}}^{Godunov} = \begin{cases} f(u_j) \ldots if \ldots (s > 0) \\ f(u_{j+1}) \ldots if \ldots (s < 0) \end{cases} \tag{29}$$

By the Rankine-Hugoniot relation for Burgers equation, $f(u_j) = u_j^2/2$, the speed is simply:

$$s = \frac{u_j + u_{j+1}}{2} \tag{30}$$

as for upwinding. On the other hand, if $a(u_j) < 0 < a(u_{j+1})$, solving the Riemann problem looks something like adding a viscous flux at the transonic point, so that we can write

$$F_{j+\frac{1}{2}}^{Godunov} = F_{j+\frac{1}{2}}^{upwind} + \alpha(u_{j+1} - u_j) \tag{31}$$

where $\alpha > 0$ and is of order $|u_j|$ and $|u_{j+1}|$. This adds a flux in the $-x$ direction which tends to smooth out rarefaction shocks. Godunov's method modifies the direction of informa-

tion transfer, depending on the sign of the wavespeed, just as upwinding does. However, Godunov's method also tends to smooth out physically unrealistic rarefaction shocks.

2.2.3 The second-order Godunov method

The second-order Godunov's method is a two-sided predictor/corrector scheme. This allows information to travel from both sides of the cell. First we must develop the concept of the approximate Riemann problem.

For convex functions,[1] consider the case where the wavespeed in adjacent cells is increasing as $a(u_j) < 0 < a(u_{j+1})$. We would like to obtain $\hat{u}_{j+\frac{1}{2}}$, which we define as the exact solution to $a\left(\hat{u}_{j+\frac{1}{2}}\right) = 0$. We can approximate the solution by linear interpolation:

$$\hat{u}_{j+\frac{1}{2}} \approx u_j^n - \frac{a(u_j^n)}{a(u_{j+1}^n) - a(u_j^n)} (u_j^n - u_{j+1}^n) \tag{32}$$

For this case, there will be only one value of u for which $a(u) = 0$, as shown in figure +1, below.

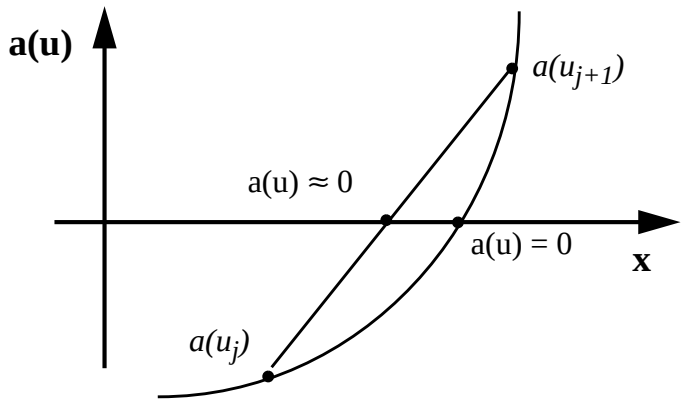


Figure 29Geometric representation of approximate solution

Then the flux can be written

$$f\left(\hat{u}_{j+\frac{1}{2}}\right) = F^{upwind}_{j+\frac{1}{2}} + \alpha (u_j - u_{j+1}) \tag{33}$$

where $\alpha > 0$, and is of order u_j and u_{j+1}. To make this scheme second-order, we follow the same guidelines as for linear equations. Given u_j^n at time t^n, we must find the interpolation

1. In other words, $f'' > 0$. This condition is met for the case of gasdynamics, but almost everything else in the real world is not convex.Examples of nonconvex functions are solutions to flow through porous media, magnetohydrodynamics, shocks in solids and phase change.

function; solve the initial-value problem for $u_I(x)$ exactly; and average the results back down to the grid.

2.2.3.1 Outline of the method

Let's handle each of these steps sequentially. Assume that we know u_j^n at the cell centers, and that Δx and Δt are constants. We would like to evolve u such that

$$u_j^{n+1} = \pounds(u^n) \tag{34}$$

where $\pounds$ is a nonlinear operator.

(i) **Find the interpolation function $u_I(x)$**. The constraint on $u_I(x)$ is that

$$u_j^n = \frac{1}{\Delta x} \int_{(j-\frac{1}{2})\Delta x}^{(j+\frac{1}{2})\Delta x} u_I(x)\, dx$$

To find $u_I(x)$, calculate

$$u_I(x) = u_j^n + \frac{(x - j\Delta x)}{\Delta x} \Delta^{VL} u_j \tag{35}$$

where Δ^{VL} is the van Leer slope. To construct the van Leer slopes, recall the definition from the discussion of geometric limiters for the linear scalar problem:

$$\Delta^{VL} u_j = \begin{cases} S_j \cdot min\left(2|u_{j+1} - u_j|, 2|u_j - u_{j-1}|, \frac{1}{2}|u_{j+1} - u_{j-1}|\right) \ldots if \ldots (\varphi > 0) \\ 0 \ldots otherwise \end{cases} \tag{36}$$

where

$$S_{j+\frac{1}{2}} = sign(u_{j+1} - u_{j-1}) \tag{37}$$

and

$$\varphi = (u_{j+1} - u_j) \cdot (u_j - u_{j-1}) \tag{38}$$

The usage of flux limiters makes the scheme second order and, furthermore, helps us to avoid the nuisance of Gibb's phenomenon at discontinuities.

(ii) **Solve the initial-value problem for $u_I(x)$ exactly.** To accomplish this, calculate the left and right states for the Riemann problem, depending on the sign of the wavespeed:

$$u_{j+\frac{1}{2},L} = \begin{cases} u_j^n + \frac{1}{2}\left(1 - a\left(u_j^n\right)\frac{\Delta t}{\Delta x}\right)\Delta^{VL}u_j \ldots if \ldots \left(a\left(u_j\right) > 0\right) \\ u_j^n + \frac{1}{2}\Delta^{VL}u_j \ldots if \ldots \left(a\left(u_j\right) < 0\right) \end{cases} \tag{39}$$

$$u_{j+\frac{1}{2},R} = \begin{cases} u_{j+1}^n - \frac{1}{2}\left(1 + a\left(u_{j+1}^n\right)\frac{\Delta t}{\Delta x}\right)\Delta^{VL}u_{j+1} \ldots if \ldots \left(a\left(u_j\right) < 0\right) \\ u_{j+1}^n - \frac{1}{2}\Delta^{VL}u_{j+1} \ldots if \ldots \left(a\left(u_j\right) > 0\right) \end{cases} \tag{40}$$

These formulas were derived by considering upstream-centered Taylor series expansions on both sides. Consider a Taylor's expansion for $u_{j+1/2,L}$.

$$u_{j+\frac{1}{2},L} = u_j^n + \frac{\Delta x}{2}\frac{\partial u}{\partial x}\bigg|_{j\Delta x} + \frac{\Delta t}{2}\frac{\partial u}{\partial t}\bigg|_{j\Delta x} + \ldots$$

Transform the temporal derivative to a spatial one by using the PDE in nonconservation form (i.e., equation (3)):

$$u_{j+\frac{1}{2},L} = u_j^n + \left(\frac{\Delta x}{2} - a\left(u_j^n\right)\frac{\Delta t}{2}\right)\frac{\partial u}{\partial x}\bigg|_{j\Delta x}$$

where $a(u_j^n) > 0$. Finally we replace the spacial derivative of u with a slope limited derivative,

$$u_{j+\frac{1}{2},L} = u_j^n + \frac{1}{2}\left(1 - a\left(u_j^n\right)\frac{\Delta t}{\Delta x}\right)\Delta^{VL}u_j \tag{41}$$

Had we been traveling to the left, $a(u_j^n) < 0$, and

$$u_{j+\frac{1}{2},R} = u_{j+1}^n - \frac{1}{2}\left(1 + a\left(u_{j+1}^n\right)\frac{\Delta t}{\Delta x}\right)\Delta^{VL}u_{j+1} \tag{42}$$

After having calculated the left and right states, solve the Riemann problem, i.e., find the half-step value $u_{j+1/2}^{n+1/2}$ along $x/t = 0$.

(iii) **Average the results back down to the grid.** To find the equivalent flux form, compute $u_{j+1/2}$ for $t^n < t < t^n + \Delta t$ and find

$$F_{j+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t^n}^{(t^n+\Delta t)} f\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right)dt \tag{43}$$

Finally, calculate the updated value of the solution by conservative differencing

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{\Delta x}\left(F_{j-\frac{1}{2}} - F_{j+\frac{1}{2}}\right) \tag{44}$$

2.2.3.2 Analysis of the method

Let's examine how this method performs in the following general categories of solution regimes:

(i) If all of the wave speeds are moving in one direction, will this scheme do the right thing? Will the wavespeeds $a(u_j^n)$'s have unambiguous sign?

(ii) What happens at shocks?

(iii) What happens at transonic expansions? Can this scheme generate an entropy-violating discontinuity?

One can see that that the upwind state is always chosen by this method. When the wavespeed is positive $a(u_j^n) > 0$, information propagates from the left and conversely, when $a(u_j^n) < 0$ information propagates from the right.

.

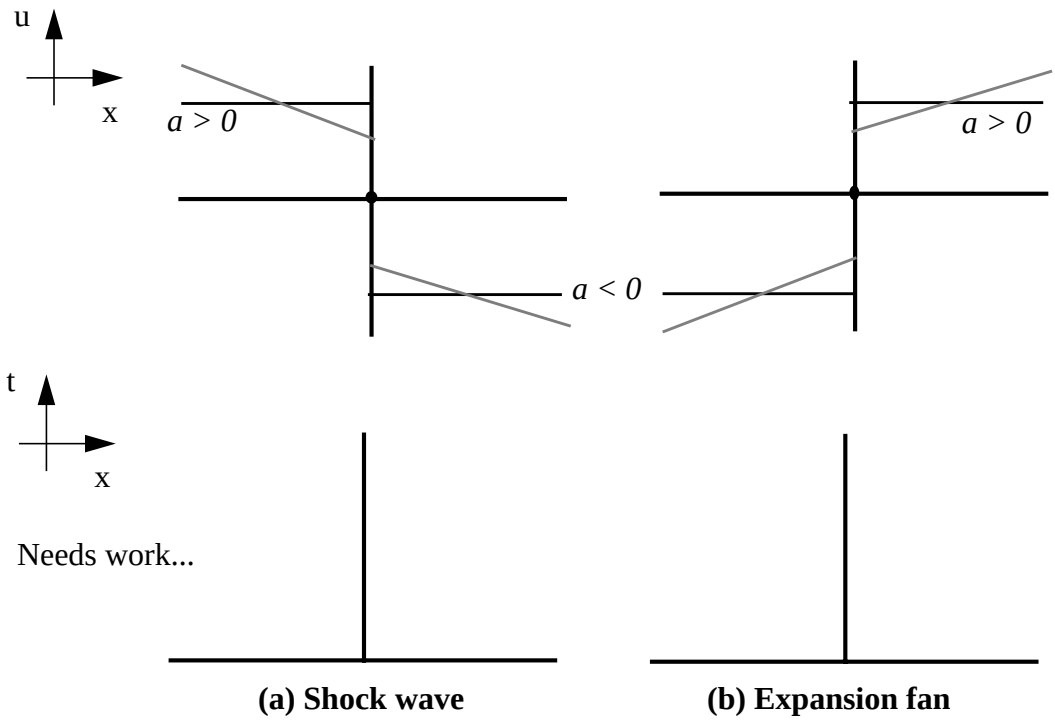


Figure 30Geometrical interpretation of Godunov's method at discontinuities

This method is well behaved at shocks. Imagine the situation shown in Figure 8 above. The wavespeed a changes sign across the shock. To deal with this, we extrapolate to the cell edge from both sides using a Taylor series expansion. This yields two different states at the cell edge. Next, treating these two extrapolated values as the left and right states of the Riemann problem, let the Riemann solver sort out the appropriate upwind value

Now let's examine the accuracy issue. We begin by examining the local truncation error in regions where the solution is smooth. Assume that the exact solution at the cell center is given by $u^n{}_{E_j} = u_E(j\Delta x, n\Delta t)$ and that u_E satisfies the PDE

$$\frac{\partial u_E}{\partial t} + a\,(u_E)\,\frac{\partial u_E}{\partial x} \;=\; 0 \tag{45}$$

Further assume that u_E is smooth enough (say it has two or three derivatives) and that the wave speed $a(u_E) > 0$. So far, we will make no restrictions on the convexity of u_E. We claim that, for second-order Godunov:

$$u_E{}^{n+1} - \mathcal{L} u_E{}^n = O(\Delta x^3) \tag{46}$$

Define the predictor step by

$$\tilde{u}_{E_{j+\frac{1}{2}}} \;=\; u^n_{E_j} + \frac{1}{2}\,(1 - a\,(u^n_{E_j})\,\frac{\Delta t}{\Delta x})\,\Delta u^n_{E_j} \tag{47}$$

and define

$$u_{E_{j+\frac{1}{2}}}^{n+\frac{1}{2}} \;=\; u_E\,(\,(j+\frac{1}{2})\,\Delta x,\;(n+\frac{1}{2})\,\Delta t) \tag{48}$$

or, rewriting as a Taylor series expansion about $j\Delta x, n\Delta t$:

$$\begin{aligned} u_{E_{j+\frac{1}{2}}}^{n+\frac{1}{2}} \;=\; & u^n_{E_j} + \frac{\Delta x}{2}\frac{\partial u_E}{\partial x}\bigg|_{j\Delta x,\,n\Delta t} + \frac{\Delta x^2}{8}\frac{\partial^2 u_E}{\partial x^2}\bigg|_{j\Delta x,\,n\Delta t} + \frac{\Delta t}{2}\frac{\partial u_E}{\partial t}\bigg|_{j\Delta x,\,n\Delta t} + \\ & \frac{\Delta t^2}{8}\frac{\partial^2 u_E}{\partial t^2}\bigg|_{j\Delta x,\,n\Delta t} + \frac{\Delta x \Delta t}{4}\frac{\partial^2 u_E}{\partial x \partial t}\bigg|_{j\Delta x,\,n\Delta t} + O\,(\Delta x^3, \Delta t^3) \end{aligned} \tag{49}$$

Since u_E is a smooth function and satisfies the PDE:

$$\frac{\partial u_E}{\partial t}\bigg|_{j\Delta x,\,n\Delta t} \;=\; -a\,(u_E)\,\frac{\partial u_E}{\partial x}\bigg|_{j\Delta x,\,n\Delta t} \tag{50}$$

Then, substituting equation (50) into (49) and regrouping, we obtain,

$$u_{E_{j+\frac{1}{2}}}^{n+\frac{1}{2}} = u_{E_j}^n + \left(\frac{\Delta x}{2} - a\left(u_{E_j}^n\right)\frac{\Delta t}{2}\right)\left.\frac{\partial u_E}{\partial x}\right|_{j\Delta x,\, n\Delta t} + \left(\frac{\Delta x^2}{8} - a\left(u_{E_j}^n\right)\frac{\Delta t^2}{8}\right)\left.\frac{\partial^2 u_E}{\partial x^2}\right|_{j\Delta x,\, n\Delta t} + \left.\frac{\Delta x \Delta t}{4}\frac{\partial^2 u_E}{\partial x \partial t}\right|_{j\Delta x,\, n\Delta t} + O\left(\Delta x^3, \Delta t^3\right) \tag{51}$$

We know that as $\Delta x \to 0$ and $\Delta t \to 0$, the ratio of Δt to Δx goes to a constant value so that $O(\Delta t^3) \to O(\Delta x^3)$. Comparing the expressions for (51) and twiddle (47) above, and noting that

$$\Delta u_{E_j}^n - (\Delta x)\left.\frac{\partial u_E}{\partial x}\right|_{j\Delta x,\, n\Delta t} \approx O\left(\Delta x^3\right) \tag{52}$$

if the derivative is approximated with central differencing, we conclude that

$$u_{E_{j+\frac{1}{2}}}^{n+\frac{1}{2}} - \tilde{u}_{E_{j+\frac{1}{2}}} = C\left[\left(\left(j+\frac{1}{2}\right)\Delta x\right)\Delta x^2 + O\left(\Delta x^3\right)\right] \tag{53}$$

where C indicates some smooth function. Notice that the term in brackets is of order Δx^3. Now we need to show that the local truncation error of the evolved solution is of order Δx^3. Using central differencing, we know that:

$$\frac{u_{E_j}^{n+1} - u_{E_j}^n}{\Delta t} = \left.\frac{\partial u_E}{\partial t}\right|_{j\Delta x,\ (n+1/2)\Delta t} + O\left(\Delta t^2\right) \tag{54}$$

and

$$f\left(u_{E_{j+1/2}}^{n+1/2}\right) - f\left(u_{E_{j-1/2}}^{n+1/2}\right) = \left.\frac{\partial f}{\partial x}\right|_{(j+1/2)\Delta x,\ (n+1/2)\Delta t} + O\left(\Delta x^2\right) \tag{55}$$

Then the evolved solution is

$$L\left(u_{E_j}^n\right) = u_{E_j}^n + \frac{\Delta t}{\Delta x}\left[f\left(\tilde{u}_{j-1/2}\right) - f\left(\tilde{u}_{j+1/2}\right)\right]$$

$$L\left(u_{E_j}^n\right) = u_{E_j}^n + \frac{\Delta t}{\Delta x}\left[f\left(u_{E_{j-1/2}}^{n+1/2}\right) - f\left(u_{E_{j+1/2}}^{n+1/2}\right)\right] + C_1\left[\left(\left(j+\frac{1}{2}\right)\Delta x\right)\Delta x^2\right] + C_2\left[\left(\left(j-\frac{1}{2}\right)\Delta x\right)\Delta x^2\right] + O\left(\Delta x^3\right) \tag{56}$$

2.2.4 The convexification of the Riemann problem

The next question to ask is what happens if f'' changes sign? (If $f'' < 0$, the condition for convexity is not met.) The answer is that weak solutions still exist, but entropy conditions become much harder to enforce. The Lax entropy condition ($a(u_L) \geq s \geq a(u_R)$) is a necessary but not sufficient condition.

Let's examine the Riemann problem with some *f(u)* which is monotonically increasing but has a change in curvature. In pictures:

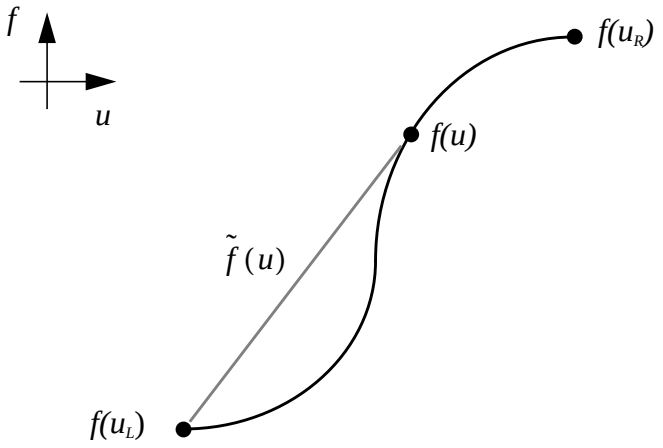


Figure 31.Pictorial representation of *f(u)*

The goal is to construct a new function, $\tilde{f}(u)$ between u_L and u_R which is the smallest convex function which is greater than or equal to *f*. Then solve the Riemann problem with the Lax entropy condition on the new convex function $\tilde{f}(u)$.

This function $\tilde{f}(u)$ turns out to be either equal to *f* or is linear. For example

,

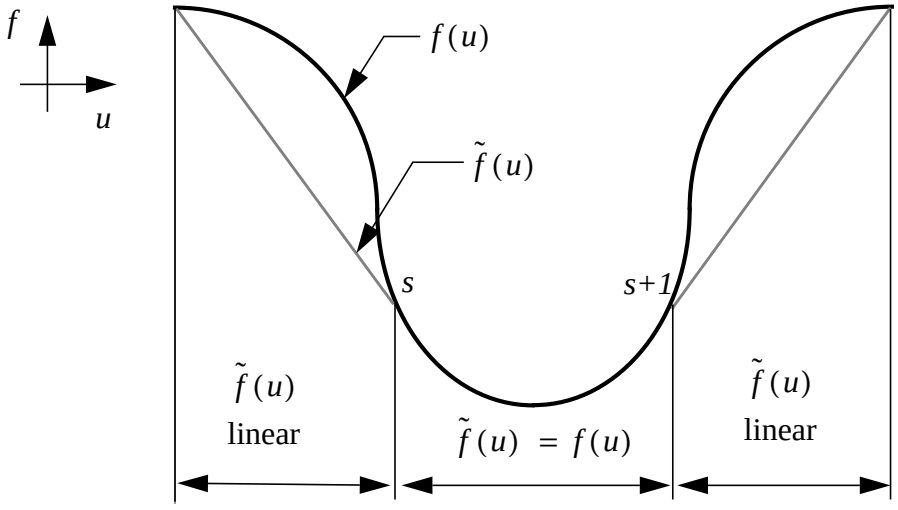


Figure 32Convexification of the function *f(u)*

Next, place discontinuities at the intersections between the regions where $\tilde{f}(u)$ follows f and where it is linear (locations s and $s+1$ in Figure 10). The wavespeed of the discontinuity is

$$\frac{d\tilde{f}}{du} = \frac{f(u_{s+1}) - f(u_s)}{u_{s+1} - u_s} \tag{57}$$

If $u_L > u_R$, this will look like a shock; if the opposite holds true, then $\tilde{f}(u) = f$ and behavior like an expansion fan will be seen. This however is a lot of work. We discuss an approach in the next section.

2.2.5 The Engquist-Osher flux

Consider the flux obtained for first-order Godunov:

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{\Delta x}\left[f(u_{RP_{j-1/2}}) - f(u_{RP_{j-1/2}})\right] \tag{58}$$

where u_{RP} is the solution for the Riemann problem with left and right states u_j^n and u_{j+1}^n. We would like to find a flux function that is:

(i) upwind when the sign of a is unambiguous;

(ii) always more dissipative than Godunov;

(iii) easier to deal with than Godunov.

One solution is to use the Engquist-Osher flux, f_{EO}, where

$$f_{EO} = f(u_L) + \int_{u_L}^{u_R} min(a(u), 0)\, du \tag{59}$$

where $a(u) = df/du$. The term f_G is the Godunov flux. This indicates that f_{EO} is simply the Godunov flux plus some diffusive function of (u_L-u_R). Let's examine the specific cases which arise when $f(u)$ is convex ($f'' > 0$):

(i) $a(u) > 0$: $f_{EO} = f_G = f(u_L)$

(ii) $a(u) < 0$: $f_{EO} = f_G = f(u_R)$

(iii) $a(u_L) < 0 < a(u_R)$: $f_{EO} = f_G = f(a^{-1}(0))$ where $a^{-1}(0)$ is the value corresponding to u=0?

(iv) $a(u_L) > 0 > a(u_R)$: This transonic case is the interesting one. For this case, $f_G = f(u_L)$ or $f(u_R)$, depending on the sign of s. However f_{EO} is given by:

$$f_{EO} = f(u_L) + \int_{u_L}^{u_R} min(a(u), 0)\, du$$

$$f_{EO} = f(u_L) + \int_{u_L}^{a^{-1}(0)} min(a(u), 0)\, du + \int_{a^{-1}(0)}^{u_R} min(a(u), 0)\, du$$

$$0$$

$$f_{EO} = f(u_L) + \int_{a^{-1}(0)}^{u_R} min(a(u), 0)\, du$$

$$f_{EO} = f(u_L) + f(u_R) - f(a^{-1}(0)) \tag{60}$$

Note that for this case $f_{EO} \neq f_G$. For Burger's equation, $f(u) = u^2/2$ and

$$f_G = \begin{cases} \frac{u_L^2}{2} \ldots if \ldots (s > 0) \\ \frac{u_R^2}{2} \ldots if \ldots (s < 0) \end{cases} \tag{61}$$

and

$$f_{EO} = \frac{u_L^2}{2} + \frac{u_R^2}{2} \tag{62}$$

Since f_{EO} is larger than f_G, it is more diffusive and can push more "stuff" down gradients. This indicates that the Engquist-Osher flux will spread out discontinuities, but only for the case of transonic flow; however, the numerical diffusion will be somewhat muted by the characteristics which will drive the discontinuities back in. This is advantageous over the convexification approach described in the last section, since there is much less logic involved. In addition, the logic that is involved is simply algebra; once we find *a(0)* for the first time, we can simply store it and we're set. For further details, see Bell, Colella and Trangenstein (1989).

The drawback to this approach is the difficulty in extending the procedure tohigher order. Osher has shown that the only known schemes which are known to produce entropy-satisfying shocks for a general *f* are first order while Bell and Shubin have made the empirical observation that using van Leer flux limiting will lead to entropy-violating discontinuities. We know that for convex *f*, entropy violations are very unstable, however for nonconvex *f*, entropy violations can actually steepen the shock. The answer lies in changing the limiter. For example, one might try some kind of modified van Leer approach. If either $f'' > 0$ or $f'' < 0$ over large intervals of *u*, one might be able to reduce the scheme to first order only near points where $f'' = 0$ and retain higher-order accuracy elsewhere.

3. Systems of conservation laws

Let's now turn our attention to systems of equations which define conservation laws in one spatial dimension. We would like to solve an initial-value problem for $U(x, t) \in \Re^N$ with $F(U) \in \Re^N$ such that:

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0 \tag{1}$$

with $U(x,0) = U_o$ given. We now know that, even for smooth U_o, the solution can develop discontinuities in finite time. So, we need a numerical method that will recognize and accommodate this limitation. For example, a weak solution will do:

$$\int_{x_L}^{x_R} [U(x, t^N) - U(x, t^0)]\, dx + \int_{t^0}^{t^N} [F(U(x_R, t)) - F(U(x_L, t))]\, dt = 0 \tag{2}$$

Note that U now represents a vector, with each component of U corresponding to a conserved quantity. For the case of gas dynamics, the conserved quantities are density ρ, momentum, ρu, and energy ρE:

$$U = (\rho, \rho u, \rho E)^T \quad F(U) = (\rho u, \rho u^2 + p, \rho u E + u p)^T$$

The pressure p is assumed to be a function of density and internal energy e, where e is defined by:

$$e = E - \frac{u^2}{2} \tag{3}$$

so that pressure may be given by, for example:

$$p = \frac{\rho e}{\gamma - 1} \tag{4}$$

where γ is the ratio of specific heats at constant pressure to that at constant volume.

3.1 The linearized perturbation equations

Assume that U can be expressed as some base state, U_o, with a small perturbation around that state, U', i.e.,

$$U = U_o + U' \tag{5}$$

The state U_o is taken to be constant. We can substitute the expansion of equation (5) into the PDE (1):

$$\frac{\partial U'}{\partial t} + \frac{\partial F(U_o + U')}{\partial x} = 0$$

To write equation (5) in linearized form, first use the definition of the gradient of F with respect to U to denote:

$$A(U_o) = \nabla_U F|_{U = U_o} = \left.\frac{\partial F_i}{\partial U_j}\right|_{U = U_o} \tag{6}$$

then we can expand F by

$$F(U_o + U') = F(U_o) + A(U_o)\,U' + \ldots \tag{7}$$

Neglecting terms of higher order than the perturbation U', we can form the linearized perturbation equation:

$$\frac{\partial U'}{\partial t} + A(U_o)\frac{\partial U'}{\partial x} = 0 \tag{8}$$

After this point, we will drop the prime on U. Notice that $A(U_o)$ is a constant-valued matrix since the gradient of F is evaluated at the base state U_o. What conditions on A are necessary for us to be able to solve this system of equations? Suppose that the system is *hyperbolic*. In this case, A has n linearly independent right eigenvectors r_k:

$$Ar_k = \lambda_k r_k \tag{9}$$

where λ_k are real eigenvalues. This will allow us to diagonalize A and decouple the system of equations (8).

Define R to be the rectangular matrix made up of columns of n right eigenvectors:

$$R = (r_1, \ldots, r_k, \ldots, r_n) \tag{10}$$

Then by equation (9)

$$AR = RD$$

or

$$R^{-1}AR = D \tag{11}$$

where D is the diagonal matrix comprised of the eigenvalues:

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ldots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \tag{12}$$

First, premultiply equation (8) by R^{-1} (where RR^{-1} is the identity matrix):

$$R^{-1}\frac{\partial U}{\partial t} + R^{-1}ARR^{-1}\frac{\partial U}{\partial x} = 0$$

Since R^{-1} is a constant, it can be taken inside the derivatives. Now use equation (11) above for D to obtain:

$$\frac{\partial (R^{-1}U)}{\partial t} + D\frac{\partial (R^{-1}U)}{\partial x} = 0$$

If we define $W = R^{-1}U = (w_1, \ldots, w_k, \ldots, w_n)$, then we obtain the system of equations,

$$\frac{\partial W}{\partial t} + D\frac{\partial W}{\partial x} = 0 \tag{13}$$

which can also be written as n uncoupled scalar advection equations,

$$\frac{\partial w_k}{\partial t} + \lambda_k\frac{\partial w_k}{\partial x} = 0 \tag{14}$$

Note that the eigenvalues represent the wavespeed of the kth component of W. The initial conditions on W are

$$W(x, 0) = R^{-1}U(x, 0) \tag{15}$$

The idea is to work for as long as we can in U variables.

3.1.1 Perturbations of the Riemann problem

Let's now consider some small perturbations $\delta U_{L,R}$ to the Riemann problem about the base states $U_{L,R}$. Let

$$U_L^P = U_L + \delta U_L \tag{16}$$

$$U_R^P = U_R + \delta U_R \tag{17}$$

Pick some point x/t. If

$$l_k\delta U_L = 0 \ldots for \ldots (\lambda_k > \frac{x}{t}) \tag{18}$$

and

$$l_k\delta U_R = 0 \ldots for \ldots (\lambda_k < \frac{x}{t}) \tag{19}$$

then

$$U(x, t) = U^P(x, t) \tag{20}$$

Proof:

$$U^P(x,t) = U_L + \delta U_L + \sum_{\lambda_k < \frac{x}{t}} \alpha_k^P r_k \tag{21}$$

where

$$\alpha_k^P = l_k (U_R + \delta U_R - U_L - \delta U_L)$$

but $l_k \delta U_R = 0$ for $\lambda_k < x/t$ so that the above equation is

$$\begin{aligned} \alpha_k^P &= l_k (U_R - U_L - \delta U_L) \\ &= l_k (U_R - U_L) - l_k \delta U_L \end{aligned}$$

Or, using the definition of α_k

$$\alpha_k^P = \alpha_k - l_k \delta U_1 \tag{22}$$

Substituting this expression into equation (21) we find

$$U^P(x,t) = U_L + \delta U_L + \sum_{\lambda_k < \frac{x}{t}} \alpha_k r_k - \sum_{\lambda_k < \frac{x}{t}} l_k \delta U_L r_k$$

and, using the fact that $l_k \delta U_L = 0$ for $\lambda_k > x/t$

$$U^P(x,t) = U_L + \delta U_L + \sum_{\lambda_k < \frac{x}{t}} \alpha_k r_k - \sum_k (l_k \delta U_L) r_k \tag{23}$$

But since

$$\begin{aligned} \sum_k (l_k \delta U_L) r_k &= \sum_k (L \delta U_L)_k r_k \\ &= RL \delta U_L \end{aligned}$$

By construction, $RL = I$, so that

$$\sum_k (l_k \delta U_L) r_k = \delta U_L \tag{24}$$

Putting equation (24) into (23)

$$U^P(x,t) = U_L + \delta U_L + \sum_{\lambda_k < \frac{x}{t}} \alpha_k r_k - \delta U_L$$

or finally

$$U^P(x,t) = U(x,t) \tag{25}$$

3.1.2 The first order Godunov method

The methodology for solving a scalar uncoupled system of equations like this one is very similar to the approach we used for a single scalar equation. At each time step, we know how much conserved "stuff" is in each cell, i.e.

$$U_j^n \approx \frac{1}{\Delta x} \int_{(j-\frac{1}{2})\Delta x}^{(j+\frac{1}{2})\Delta x} U(x, n\Delta t)\, dx \tag{26}$$

We then:

(i) find the interpolation function $U_I(x)$ subject to the constraint that

$$U_j^n \Delta x = \int_{(j-\frac{1}{2})\Delta x}^{(j+\frac{1}{2})\Delta x} U_I(x)\, dx; \tag{27}$$

(ii) solve the PDE exactly along the cell edge;

$$\frac{\partial U_I^{n,\,n+1}}{\partial t} + A\frac{\partial U_I^{n,\,n+1}}{\partial x} = 0 \tag{28}$$

where the double superscript indicates that this solution for $U_I(x)$ exists between time levels $n\Delta t$ and $(n+1)\Delta t$.

(iii) use conservative finite differencing to interpolate the new solution onto the grid:

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x}\left[F_{j-1/2} - F_{j+1/2}\right] \tag{29}$$

where the flux F is given by:

$$F_{j+1/2} = AU_{j+1/2} \tag{30}$$

For first-order Godunov, we use a piecewise linear interpolation function; we can think of the function U as a sequence of constant states separated by jumps at the cell edges. We then solve the Riemann problem at the cell edges with the values in adjacent cells corresponding to the left and right states (depending on the sign of the wavespeed λ_k):

$$U(x,0) = \begin{cases} U_L \ldots (x<0) \\ U_R \ldots (x>0) \end{cases} \tag{31}$$

or

$$W(x,0) = \begin{cases} R^{-1}U_L \ldots (x<0) \\ R^{-1}U_R \ldots (x>0) \end{cases}; \qquad w_k(x,0) = \begin{cases} w_{k_L} \ldots (x<0) \\ w_{k_R} \ldots (x>0) \end{cases}$$

The exact solution to the hyperbolic equation for w_k given by equation (14) above is

$$w_k(x,t) = w_k(x - \lambda_k t, 0) \tag{32}$$

Consider the case of an isolated wave with constant speed λ_k. What happens to U? First, define a left-eigenvector matrix, L, analogous to R, which is made up of n rows of left eigenvectors which satisfy:

$$l_k A = \lambda_k l_k \tag{33}$$

so that L looks like:

$$L = R^{-1} = \begin{bmatrix} l_1 \\ \ldots \\ l_n \end{bmatrix} \tag{34}$$

Find the left and right states for the Riemann solver:

$$U_L = \begin{cases} U_{j-1} \ldots if \ldots (\lambda_k > 0) \\ U_j \ldots if \ldots (\lambda_k < 0) \end{cases} \tag{35}$$

and

$$U_R = \begin{cases} U_j \ldots if \ldots (\lambda_k > 0) \\ U_{j+1} \ldots if \ldots (\lambda_k < 0) \end{cases} \tag{36}$$

Then solve the Riemann problem with:

$$U(x,0) = \begin{cases} U_L \ldots (x<0) \\ U_R \ldots (x>0) \end{cases} \tag{37}$$

so that

$$U(x,t) = U_L + \sum_{\lambda_k < \frac{x}{t}} \alpha_k r_k \tag{38}$$

where

$$\alpha_k = l_k (U_R - U_L) \tag{39}$$

We are now in a position to show that multiplying U on the left by l_k will yield the components of w that travel at speed λ_k.

$$l_k U(x,t) = l_k U_L + l_k \sum_{\lambda_{k'} < \frac{x}{t}} \alpha_{k'} r_{k'} \tag{40}$$

For $\lambda_k \geq x/t$:

$$\begin{aligned} l_k U(x,t) &= l_k U_L \\ &= w_{k_L} \end{aligned}$$

And for $\lambda_k < x/t$:

$$\begin{aligned} l_k U(x,t) &= l_k U_L + l_k (U_L - U_R) \\ &= l_k U_R \\ &= w_{k_R} \end{aligned}$$

We can think of w_k as the amplitude of the traveling disturbance for the kth wave. The way to extract that amplitude from the "primitive" variables U is to perform the vector multiplication $l_k U$. In this analysis, we would like to keep things in terms of U as far as possible, since there is no global analog like w_k for general nonlinear problems.

3.1.3 The first-order Godunov method (continued)

If we define U^{RP} as the solution to the Riemann problem with left and right states as U_j^n and U^n_{j+1}, we can update the solution by

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x}\left[F_{j-1/2} - F_{j+1/2}\right] \tag{41}$$

where

$$F_{j+1/2} = A U^{RP}_{j+1/2} \tag{42}$$

Define for discrete values

$$w^n_{k_j} = l_k U_j^n \tag{43}$$

Then

$$w^{n+1}_{k_j} = \begin{cases} w^n_{k_j} + \dfrac{\Delta t}{\Delta x}\lambda_k (w^n_{k_{j-1}} - w^n_{k_j}) \ldots if \ldots (\lambda_k > 0) \\ w^n_{k_j} + \dfrac{\Delta t}{\Delta x}\lambda_k (w^n_{k_j} - w^n_{k_{j+1}}) \ldots if \ldots (\lambda_k < 0) \end{cases} \tag{44}$$

The second terms on the right represent the flux in and out of the cell. To see this multiply equation (42) on the left by l_k and observe what happens to the flux:

$$
\begin{aligned}
l_k F_{j+1/2} &= l_k A U^{RP}_{j+1/2} \\
&= \lambda_k l_k U^{RP}_{j+1/2} \\
&= \begin{cases} \lambda_k w^n_{k_j} \ldots if \ldots (\lambda_k > 0) \\ \lambda_k w^n_{k_{j+1}} \ldots if \ldots (\lambda_k < 0) \end{cases}
\end{aligned}
$$

3.1.4 The second-order Godunov method

By now, the routine should be fairly clear. The steps which we must follow are:

(i) Interpolate for $U_I(x)$.

(ii) Solve for $U_I(x)$ exactly.
  (a) Compute that left and right states at the cell edges.
  (b) Solve the Riemann problem.

(iii) Update the solution onto the grid with conservative finite differencing.

The big difference between first- and second-order Godunov is the use of limiting. We interpolate in a piecewise linear fashion by:

$$U_I(x) = U^n_j + \frac{(x - j\Delta x)}{\Delta x} \Delta U_j \tag{45}$$

where ΔU_j is the finite-difference approximation to

$$\Delta U_j \approx \Delta x \left. \frac{\partial U}{\partial x} \right|_{j\Delta x} \tag{46}$$

when the solution is smooth and which includes limiting when it is not, similar to van Leer. In terms of the amplitudes w_k we have:

$$
\begin{aligned}
w_{k_I}(x) &= l_k U^n_j + l_k \frac{(x - j\Delta x)}{\Delta x} \Delta U_j \\
&= w^n_{k_j} + \frac{(x - j\Delta x)}{\Delta x} l_k \Delta U_j \\
&= w^n_{k_j} + \frac{(x - j\Delta x)}{\Delta x} \Delta w_{k_j}
\end{aligned}
$$

We use a slope limiter to prevent oscillations in w_k. To calculate the limiter we must define right, left and center $\alpha_{k,j}$. Let

$$\alpha^L_{k_j} = l_k (U^n_j - U^n_{j-1}) = w^n_{k_j} - w^n_{k_{j-1}} \tag{47}$$

$$\alpha^C_{k_j} = l_k (U^n_{j+1} - U^n_{j-1}) = w^n_{k_{j+1}} - w^n_{k_{j-1}} \tag{48}$$

$$\alpha^R_{k_j} = l_k (U^n_{j+1} - U^n_j) = w^n_{k_{j+1}} - w^n_{k_j} \tag{49}$$

Further define

$$\alpha_{k_j} = \begin{cases} (sign(\alpha^C_{k_j}))\, min(2|\alpha^L_{k_j}|, |\alpha^C_{k_j}|, 2|\alpha^R_{k_j}|) \ldots if \ldots (\alpha^L_{k_j}\alpha^R_{k_j} > 0) \\ 0 \ldots otherwise \end{cases} \tag{50}$$

When the solution is smooth, we would expect $|\alpha^C_{k_j}|$ to be a minimum. To compute the left and right states, use an upstream-centered Taylor expansion:

$$U^{n+1/2}_{j+1/2_L} = U^n_j + \frac{\Delta t}{2}\frac{\partial U}{\partial t} + \frac{\Delta x}{2}\frac{\partial U}{\partial x} + \ldots$$

$$\approx U^n_j + \frac{1}{2}(I - A\frac{\Delta t}{\Delta x})\,\Delta x \frac{\partial U}{\partial x}$$

so that

$$U^{n+1/2}_{j+1/2_L} = U^n_j + \frac{1}{2}(I - A\frac{\Delta t}{\Delta x})\,\Delta U_j \tag{51}$$

$$U^{n+1/2}_{j+1/2_R} = U^n_{j+1} - \frac{1}{2}(I + A\frac{\Delta t}{\Delta x})\,\Delta U_{j+1} \tag{52}$$

Using equations (51) and (52) above as input states, compute the solution to the Riemann problem, evaluated at *x/t = 0* to obtain $U^{n+1/2}_{j+1/2}$. To compute the predictor for second-order Godunov on the w_k's, we would like to claim that

$$l_k U^{n+1/2}_{j+1/2} = w^{n+1/2}_{k_{j+1/2}} = \begin{cases} w^n_{k_j} + \frac{1}{2}(1 - \lambda_k \frac{\Delta t}{\Delta x})\,\Delta w^n_{k_j} \ldots if \ldots (\lambda_k > 0) \\ w^n_{k_{j+1}} + \frac{1}{2}(1 - \lambda_k \frac{\Delta t}{\Delta x})\,\Delta w^n_{k_{j+1}} \ldots if \ldots (\lambda_k < 0) \end{cases}$$

However, if we premultiply equation (51) for the left state above by l_k, we obtain:

$$l_k (U^{n+1/2}_{j+1/2_L} = U^n_j + \frac{1}{2}(I - A\frac{\Delta t}{\Delta x})\,\Delta U_j) \tag{53}$$

$$w^{n+1/2}_{k_{j+1/2_L}} = w^n_{k_j} + \frac{1}{2}(1 - \lambda_k \frac{\Delta t}{\Delta x})\, l_k \Delta U_j \tag{54}$$

For $\lambda_k > 0$, this is precisely the Godunov predictor that we want.

3.1.4.1 Stability of the method

Since we are dealing with a system of uncoupled scalar equations the stability of this scheme can be proved exactly as for scalars. We know that

$$l_k U_j^{n+1} = w_{k_j}^{n+1} \tag{55}$$

and that the w_k's satisfy

$$w_{k_j}^{n+1} = w_{k_j}^n + \frac{\Delta t}{\Delta x}\left(\lambda_k w_{k_{j-1/2}}^{n+1/2} - \lambda_k w_{k_{j+1/2}}^{n+1/2}\right) \tag{56}$$

3.1.4.2 Local truncation error

Assume that we know the exact solution evaluated at discrete points in space and time

$$U_{E_j}^n = U(j\Delta x, n\Delta t) \tag{57}$$

We further assume the solution is smooth. Let's analyze the case for $\lambda_k \neq 0$. The local truncation error, *LTE*, is

$$LTE = U_E^{n+1} - L(U_E^n) \tag{58}$$

where $L(U_E^n)$ is given by

$$L(U_E^n) = U_E^n + \frac{\Delta t}{\Delta x}\left(A\tilde{U}_{E_{j-1/2}} - A\tilde{U}_{E_{j+1/2}}\right) \tag{59}$$

The second and third terms on the RHS of equation (59) come from the evaluation of the Riemann problem with left and right states given by

$$\tilde{U}_{E_{j+1/2_L}} = U_{E_j}^n + \frac{1}{2}\left[I - A\frac{\Delta t}{\Delta x}\right](\Delta U_E^n)_j \tag{60}$$

$$\tilde{U}_{E_{j+1/2_R}} = U_{E_j}^n - \frac{1}{2}\left[I + A\frac{\Delta t}{\Delta x}\right](\Delta U_E^n)_j \tag{61}$$

We would like to use the Lax equivalence theorem to show that this method converges and that the rate of convergence is $O(\Delta x^2)$. Thus we need to prove that the following difference is the sum of a smooth function of order Δx^2 plus higher-order terms:

$$\tilde{U}_{E_{j+1/2}} - U_{E_{j+1/2}}^{n+1/2} = C\left[\left((j+\frac{1}{2})\Delta x\right)\Delta x^2\right] + O(\Delta x^3) \tag{62}$$

It is easy to show that $\tilde{U}_{E_{j+1/2_L}} - U_{E_{j+1/2}}^{n+1/2}$ is $O(\Delta x^2)$ by keeping a few more terms of the Taylor expansion and using the PDE, as was done for the linear scalar equation previously. Note that $\tilde{U}_{E_{j+1/2_L}}$ picks out the positive eigenvalues, while $\tilde{U}_{E_{j+1/2_R}}$ picks out the negative eigenvalues. Thus,

$$\tilde{w}_{k_{j+1/2}} = \begin{cases} \tilde{w}_{k_{j+1/2_L}} \ldots for \ldots (\lambda_k > 0) \\ \tilde{w}_{k_{j+1/2_R}} \ldots for \ldots (\lambda_k < 0) \end{cases} \tag{63}$$

so that $\tilde{w}_{k_{j+1/2}} - w^{n+1/2}_{E_{j+1/2}}$ is $O(\Delta x^2)$.

3.2 The effect of a nonlinear change of variables

Consider the following question. How do the eigenvectors and eigenvalues transform under a nonlinear changes of variable? Let's begin by examining the general case. Then, as an illustration of why this is useful, we will examine the specific case of gasdynamics.

3.2.1 The general case

Given the system of n equations in the vector U,

$$\frac{\partial U}{\partial t} + A(U)\frac{\partial U}{\partial x} = 0 \tag{64}$$

where $A(U) = \nabla_U F$. Recall from our previous discussion that have n eigenvalues λ_k and right eigenvectors r_k. We would like to find a nonlinear change of variables $V(U)$. The derivatives can then be transformed by:

$$\frac{\partial U}{\partial t} = \nabla_V U \frac{\partial V}{\partial t} \tag{65}$$

$$\frac{\partial U}{\partial x} = \nabla_V U \frac{\partial V}{\partial x} \tag{66}$$

From the inverse function theorem

$$(\nabla_V U)^{-1} = \nabla_U V \tag{67}$$

Let $A^U = A(U)$. We can rewrite equation (64) as,

$$\nabla_V U \frac{\partial V}{\partial t} + A^U \nabla_V U \frac{\partial V}{\partial x} = 0$$

Premultiplying by $(\nabla_V U)^{-1}$

$$\frac{\partial V}{\partial t} + (\nabla_V U)^{-1} A^U \nabla_V U \frac{\partial V}{\partial x} = 0$$

If we define A^V by

$$A^V = (\nabla_V U)^{-1} A^U (\nabla_V U) \tag{68}$$

then we have obtained the equation:

$$\frac{\partial V}{\partial t} + A^V \frac{\partial V}{\partial x} = 0 \tag{69}$$

If we can easily find A^V, $r^V{}_k$, and $\lambda^V{}_k$, how can we figure out what $r^U{}_k$, and $\lambda^U{}_k$ are supposed to be? We first consider what the r_k's tell us by examining the propagation of an isolated wave. The eigenvalues represent the amplitude of that wave traveling in space. Changing our dependent variables does not change the physical wave speed; it only changes what we measure it to be. Therefore, it makes sense that

$$r_k^U = (\nabla_V U)\, r_k^V \tag{70}$$

or equivalently

$$(\nabla_V U)^{-1} r_k^U = r_k^V \tag{71}$$

Multiply equation (68) on the right by $r^V{}_k$ to obtain,

$$\begin{aligned} A^V r_k^V &= (\nabla_V U)^{-1} A^U (\nabla_V U)\, r_k^V \\ &= (\nabla_V U)^{-1} A^U (\nabla_V U)\, (\nabla_V U)^{-1} r_k^U \\ &= (\nabla_V U)^{-1} A^U r_k^U \\ &= \lambda_k (\nabla_V U)^{-1} r_k^U \\ &= \lambda_k r_k^V \end{aligned}$$

Thus we find that the eigenvalues do not change under this transformation. Note that as a consequence we can save ourselves some work by finding acoordinate system in which the eigenvalues are easy to find.

3.2.2 Gasdynamics

We begin with a quick review. In nonconservate form the governing equations of continuity, momentum and energy in one dimension are:

$$\frac{D\rho}{Dt} + \rho \frac{\partial u}{\partial x} = 0 \tag{72}$$

$$\frac{Du}{Dt} + \frac{1}{\rho} \frac{\partial p}{\partial x} = 0 \tag{73}$$

$$\frac{De}{Dt} + \frac{p}{\rho} \frac{\partial u}{\partial x} = 0 \tag{74}$$

where ρ is the density of the fluid medium, u is the velocity, e is the internal energy and the pressure p is given by a constitutive relation $p = p(\rho,e)$, e.g., $p = (\gamma-1)\rho e$. Recall that

the conserved quantities are density, momentum ρu, and energy ρE, and that the energy is related to the internal energy by

$$E = e + \frac{u^2}{2} \tag{75}$$

and that the material derivative is defined by

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} \tag{76}$$

For this case, U is

$$U = \begin{bmatrix} \rho \\ u \\ e \end{bmatrix} \tag{77}$$

Compare to the form used previously

$$\begin{bmatrix} \rho \\ u \\ e \end{bmatrix}_t + A \begin{bmatrix} \rho \\ u \\ e \end{bmatrix}_x = 0 \tag{78}$$

where A results from linearization about the base state of ρ_o, u_o, e_o and the subscripts denote differentiation with respect to that variable. You can verify for yourself that

$$A(U) = \begin{bmatrix} u & \rho & 0 \\ \frac{1}{\rho} p_\rho & u & \frac{p_e}{\rho} \\ 0 & \frac{p}{\rho} & u \end{bmatrix} \tag{79}$$

We would now like to get from these variables to the "primitive" variables by a nonlinear transformation. We defined the U variables as:

$$U = (\rho, u, e)^T \tag{80}$$

We now define the transformed variables to be:

$$V(U) = (\rho, u, p)^T \tag{81}$$

We can relate the two by

$$V(U) = V(U_o) + \nabla_U V U' \tag{82}$$

How can we accomplish this change? The governing equations for density and velocity above do not change, but we need to do something different for the pressure. Since $p = p(\rho, e)$:

$$\frac{Dp}{Dt} = \left.\frac{\partial p}{\partial e}\right|_{\rho} \frac{De}{Dt} + \left.\frac{\partial p}{\partial \rho}\right|_{e} \frac{D\rho}{Dt}$$

by the chain rule. Denoting differentiation by subscripts

$$\frac{Dp}{Dt} = -p_e \frac{\dot{p}}{\rho^2} \rho \frac{\partial u}{\partial x} - p_\rho \rho \frac{\partial u}{\partial x}$$

Since the speed of sound is defined by

$$c^2 = p_e \frac{\dot{p}}{\rho^2} + p_\rho \tag{83}$$

the material derivative of the pressure can be written as

$$\frac{Dp}{Dt} = -c^2 \frac{\partial u}{\partial x} \tag{84}$$

or, using continuity

$$\frac{Dp}{Dt} = -\rho c^2 \frac{\partial u}{\partial x} \tag{85}$$

so, in V variables, the matrix A^V can be written as:

$$A^V = u \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & \rho & 0 \\ 0 & 0 & \frac{1}{\rho} \\ 0 & \rho c^2 & 0 \end{bmatrix} \tag{86}$$

The eigenvectors of A^V can be easily found by considering only the eigenvectors of the second matrix on the RHS, since the first term on the RHS simply represents a shift in the propagation speed. This is the beauty of this change of variables; the second matrix on the RHS reduces to a simple 2 by 2 system.

As an exercise one can verify that the eigenvalues of both A^U and A^V are $u+c$, u, and $u-c$. Physically, if we think of the initial data as an isolated disturbance (or 'blip'), then these eigenvalues correspond to three waves: one moves backward relative to the disturbance's speed, one moves with the disturbance, and one moves ahead of the disturbance. The right eigenvectors for A^V are given by:

$$R = \begin{bmatrix} 1 & 1 & 1 \\ -\frac{c}{\rho} & 0 & \frac{c}{\rho} \\ c^2 & 0 & c^2 \end{bmatrix} \tag{87}$$

4. Nonlinear systems of conservation laws

If the system of equations under consideration is nonlinear, our task is more difficult. We would like to solve an initial-value problem for the vector $U(x, t) \in \Re^N$ with $F(U) \in \Re^N$ such that:

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0 \tag{1}$$

with $U(x,0) = U_o$ given and where F is now a nonlinear function of U. We can still talk about hyperbolicity, although we will require some special restrictions. As a first attempt, we will try to do the same thing that we did with linear systems; diagonalize the system and decouple the equations.

If F is differentiable, then we can still define

$$A(U_o) = \nabla_U F = \left.\frac{\partial F_i}{\partial U_j}\right|_{U = U_o} \tag{2}$$

and write (1) in quasilinear form

$$\frac{\partial U}{\partial t} + A(U)\frac{\partial U}{\partial x} = 0 \tag{3}$$

We say that the nonlinear system (1) is hyperbolic if A has n linearly independent right eigenvectors r_k,

$$Ar_k = \lambda_k r_k \tag{4}$$

and n real eigenvalues λ_k. In addition we assume that the eigenvalues can be put in strictly ascending order so that

$$\lambda_1 < \ldots < \lambda_k < \ldots < \lambda_n \tag{5}$$

for the right eigenvectors $r_1 \ldots r_k \ldots r_n$. We can still define R to be a rectangular matrix made up of columns of n right eigenvectors,

$$R = (r_1, \ldots, r_k, \ldots, r_n) \tag{6}$$

and a left-eigenvector matrix, L, analogous to R, which is made up of n rows of left eigenvectors which satisfy:

$$l_k A = \lambda_k l_k \tag{7}$$

so that L looks like:

$$L = R^{-1} = \begin{bmatrix} l_1 \\ \cdots \\ l_n \end{bmatrix} \tag{8}$$

Even though the structure of this system of PDE's and its associated eigenvalues and eigenvectors looks familiar, we must handle them in a way that is fundamentally different from the linear system. This is because the fact that $A = A(U)$ is nonlinear implies that $\lambda_k = \lambda_k(U)$; $r_k = r_k(U)$; and $l_k = l_k(U)$. This prevents us from diagonalizing the system and writing,

$$U(x, t) = \sum \alpha_k(x, t)\, r_k(x, t) \tag{9}$$

for

$$\alpha_k(x, t) = l_k U(x, t) \tag{10}$$

To see this, substitute the expansion for U in (9) above into (3) to obtain,

$$\frac{\partial}{\partial t}\sum \alpha_k r_k + A(U)\frac{\partial}{\partial x}\sum \alpha_k r_k = 0$$

$$\sum_k \left(\frac{\partial \alpha_k}{\partial t} r_k + \frac{\partial r_k}{\partial t}\alpha_k\right) + A(U)\sum_k \left(\frac{\partial \alpha_k}{\partial x} r_k + \frac{\partial r_k}{\partial x}\alpha_k\right) = 0$$

If we premultiply this expression by l_k and use equation (7) we find that,

$$\sum_k \left(l_k \cdot \left(r_k \frac{\partial \alpha_k}{\partial t} + \lambda_k r_k \frac{\partial \alpha_k}{\partial x}\right)\right) + \sum_k \alpha_k l_k \left(\frac{\partial r_k}{\partial t} + A\frac{\partial r_k}{\partial x}\right) = 0 \tag{11}$$

The second term on the right is problematic since it contains derivatives of the eigenvectors $r_k$with respect to the components of U. For example,

$$\frac{\partial r_k}{\partial t} = \nabla_U r_k \frac{\partial U}{\partial t} \tag{12}$$

For certain special systems one may be able to find a nonlinear change of variables which will allow us find quantities analogous to the wave amplitudes w_k which are constant along certain paths. These variables are known as Riemann invariants. See Smoller (1986) for a discussion of the theory of Riemann invariants.

The general statement for nonlinear systems is that signals can be interpreted as small disturbances or perturbations of some background solution and that these disturbances propagate along characteristics. We therefore look for weak solutions which satisfy the conditions associated with a single individual disturbance such as single-wave solutions separated by constant states; e.g., shocks and rarefaction waves. Since such solutions have

no distinguished length scale, the solution $U(x,t)$ is a function of only one variable x / t. We will then piece together N individual waves of this type to solve the Riemann problem, as shown in Figure 33. Although this is probably the simplest way of finding a solution of (1) problem, it is certainly not the only one. However, if we find an answer with this technique, then we can stop searching for alternate solution methods since solutions of (1) are unique.

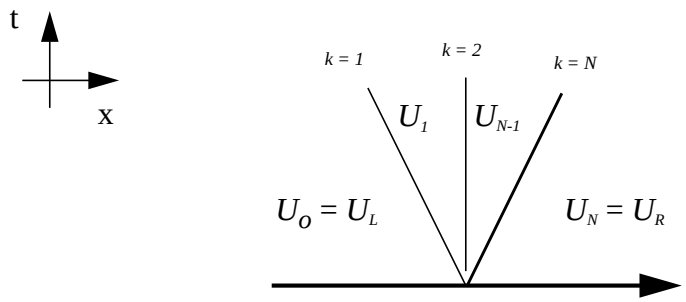


Figure 33. Graphical representation of the solution to the to a nonlinear system (1) as a sequence of constant states separated by discontinuities.

4.1 The Riemann problem

We suppose that the left and right states U_l, U_R are close in the sense that their difference U_l - U_R is small. Then the constant states U_o, U_1, ... U_N are each separated by small jumps. The kth discontinuity satisfies the Rankine-Hugoniot jump conditons,

$$F(U_{k-1}) - F(U_k) = s(U_{k-1} - U_k) \tag{13}$$

where s is the speed of propagation. We want to find (U_k - U_{k-1}) and s for each k. We assume that the waves are weakly nonlinear, so that in the limit of small U_k - U_{k-1},

$$U_k - U_{k-1} = \alpha_k r_k \tag{14}$$

The quantity $\alpha_k r_k$ can be interpreted as the *increment* of change in U. The information which propagates along characteristics isthe amplitude of the jump in α_k; the direction in which it propagates is given by the kth right eigenvector r_k.

4.2 The entropy condition

We still need an entropy condition that applies to nonlinear systems tso that we can select a unique weak solution of (1). In order to formulate an analog of the scalar version of the Lax entropy condition, we need to find the multivariable analog of a convex functions of

one variable. To begin,we require that admissible solutions still be the limit of viscous solutions of (1). As for the nonlinear scalar case, if the viscous solution U^ε is defined by:

$$\frac{\partial U^\varepsilon}{\partial t} + \frac{\partial F[U^\varepsilon]}{\partial x} = \varepsilon \frac{\partial^2 U^\varepsilon}{\partial x^2} \tag{15}$$

then we want the solution U of (1) to satisfy,

$$U = \lim_{\varepsilon \to 0} U^\varepsilon \tag{16}$$

For the nonlinear scalar case the convexity of f implies that the wave speed $a = a(u)$ increases with increasing u, $\partial a/\partial u > 0$. This means that as the strength of the wave increases, the jump in a also increases. The analog of a for nonlinear systems is the speed with which kth disturbance propagates, λ_k. The strength of the wave is measured by the amplitude α_k. So for nonlinear systems, the quantity we want to examine is $\partial \lambda_k / \partial u$. Specifically, we want $\partial \lambda_k / \partial u > 0$. Given $\lambda_k(U)$ and $r_k \cdot \nabla_U \lambda_k$, we have the following two possible cases:

(a) **Linearly degenerate:** If $r_k \cdot \nabla_U \lambda_k \equiv 0$ for all U, then the kth wave family is said to be linearly degenerate. Note that by definition, if the kth family is linearly degenerate, then r_k is perpendicular to $\nabla_U \lambda_k$, which looks like the linear scalar case. This is depicted in Figure 34(a).

(b) **Genuinely nonlinear:** If $r_k \cdot \nabla_U \lambda_k \neq 0$ for all U, then the kth wave family is said to be genuinely nonlinear. In this case r_k is never perpendicular to $\nabla_U \lambda_k$, as shown in Figure 34(b).

Note that if

$$U_k - U_{k-1} = \alpha_k r_k$$

then

$$\lambda_k(U_k) - \lambda_k(U_{k-1}) = \lambda_k(U_{k-1} + \alpha_k r_k) - \lambda_k(U_{k-1})$$

so that to second order in $(U_k - U_{k-1})$ we have,

$$\lambda_k(U_k) - \lambda_k(U_{k-1}) = \alpha_k (r_k \cdot \nabla_U \lambda_k) \tag{17}$$

If the kth wave family is genuinely nonlinear, the RHS of (17) is nonzero. This indicates that as something crosses the kth discontinuity, its speed λ_k changes.

We assume that all waves are either linearly degenerate or genuinely nonlinear. Whether or not a wave is genuinely nonlinear or linearly degenerate depends on which wave family it belongs to. For example, in gas dynamics the *k*th family is always either genuinely nonlinear or linearly degenerate. Wave falimies do not change type. The entropy condition is always satisfied for linearly degenerate waves. For genuinely nonlinear waves, we add the requirement that

$$\lambda_k(U_{k-1}) > s > \lambda_k(U_k) \tag{18}$$

This guarantees that characteristics impinge on a shock; i.e., no entropy-violating rarefaction shocks are allowed.

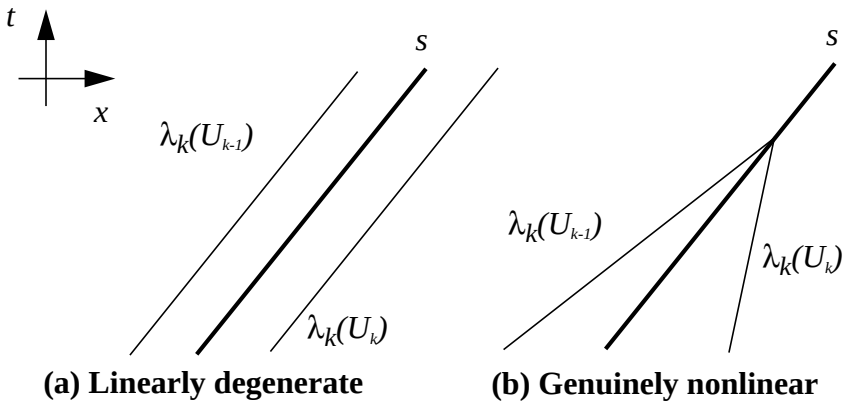


Figure 34. Linearly degenerate and genuinely nonlinear wave families.

4.3 Solution procedure for the approximate Riemann problem

We would like to solve the Riemann problem for a nonlinear system of equations. We are given initial data on the left and right states:

$$U(x,0) = \begin{cases} U_L = U_{k-1} \ldots if \ldots (x<0) \\ U_R = U_k \ldots if \ldots (x>0) \end{cases} \tag{19}$$

as shown in Figure 33. We seek a solution that satisfies the following assumptions:

(a) *U(x, t)* is a function of *x / t* only.

(b) *U(x, t)* is the limit of the viscous solution and hence satisfies the entropy condition.

(c) Each wave family *k* is either linearly degenerate or genuinely nonlinear.

We will now discuss the solution procedure.

4.3.1 The solution in phase space

The first step in the solution procedure is to find the phase space solution U_k. To begin we need to compute the left and right eigenvector matrices, L and R at a point. A logical starting place would be to calculate them as functions of $(U_L + U_R)/2$. There will be a number of intermediate constant states in between these left and right states with incremental discontinuities between them. Graphically

:

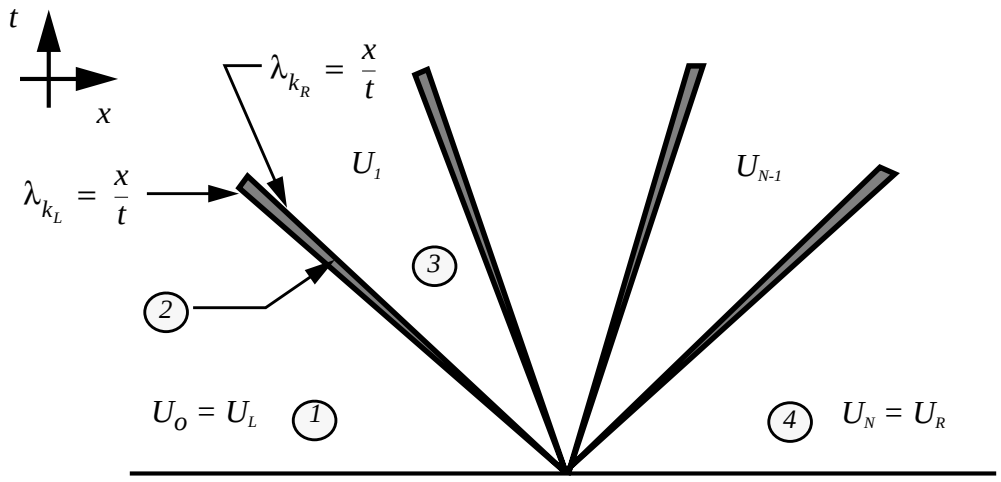


Figure 35. Separation of the problem into a series of constant states separated by incremental discontinuities.

Figure 36 depicts the *k*th wave which separates the *(k-1)*st and *k*th state.

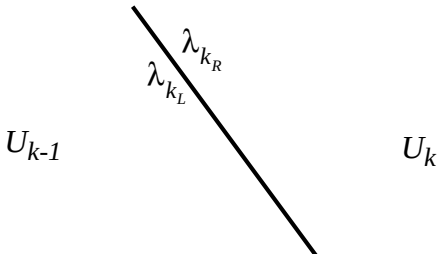


Figure 36. Left and right states and wavespeeds for the kth wave

The wave amplitudes are defined by:

$$\alpha_k = l_k \cdot (U_R - U_L) \tag{20}$$

or, equivalently:

$$U_R - U_L = \sum \alpha_k r_k \tag{21}$$

We have evaluated the eigenvectors and amplitudes at the average of the left and right states, $(U_L + U_R)/2$. If the kth wave is genuinely nonlinear and $\lambda_k(U_{k-1}) < \lambda_k(U_k)$, then we define the kth eigenvalues by,

$$\lambda_{k_L} = \lambda_k(U_{k-1}) \tag{22}$$

$$\lambda_{k_R} = \lambda_k(U_k) \tag{23}$$

Otherwise, if the kth wave family is linearly degenerate we set,

$$\lambda_{k_L} = \lambda_{k_R} = \frac{1}{2}(\lambda_k(U_{k-1}) + \lambda_k(U_k)) \tag{24}$$

4.3.2 The solution in physical space

Given a value of x/t, we can now find $U(x, t)$. Recall that U_k can be found by

$$U_k = U_L + \sum_{k' \le k} \alpha_{k'} r_{k'} \tag{25}$$

There is a jump carried with each wave. Consider the kth wave. If it is linearly degenerate, treat it as a discontinuity traveling at the speed $s = [\lambda_k(U_{k-1}) + \lambda_k(U_k)]/2$. If it is genuinely nonlinear, treat it as a discontinuity which satisfies the analog of the Lax entropy condition. Four cases are possible, as shown in Figure 35 above:

(a) If $x/t < \lambda_{k_L}$ then, $U(x, t) = U_L$, since we are downstream of the wave.

(b) If $\lambda_{k_L} \le x/t < \lambda_{k_R}$, then the point x / t lies in a rarefaction fan. In this case we linearly interpolate to find the value of U inside the rarefaction wave.

$$U(x, t) = U_{k-1} + \frac{x/t - \lambda_{k_L}}{\lambda_{k_R} - \lambda_{k_L}} [U_k - U_{k-1}] \ldots k = 1, \ldots, N$$

(c) If $\lambda_{k-1_R} \le x/t < \lambda_{k_L}$, then, $U(x, t) = U_{k-1} \ldots k = 1, \ldots, N$.

(d) If $x/t > \lambda_{k_R}$, then $U(x, t) = U_R$ since we are upstream of the wave.

4.3.3 Miscellaneous tricks

Although this is a nonlinear problem, it looks very similar to the linear case. The difficulty lies in ensuring that the entropy condition is not violated, so we must check carefully for

genuinely nonlinear waves and rarefaction shocks. To accomplish this one can use the following procedures:

(a) **Choose the right nonlinear change of variables.** Work in terms of the variables $V = V(U)$; e.g., the "primitive" variables for gasdynamics

$$V(U) = \begin{bmatrix} \rho \\ u \\ p \end{bmatrix} \tag{26}$$

The goal is to find a change of variables which makes it easier to compute the eigenvectors, especially the right-eigenvector matrix, R. It should also be simpler to verify the ordering of the wavespeeds λ_k.

(b) **Augment the dependent variables.** For the case of gasdynamics, the nonlinear change of variables is as shown in equation (26) above and the corresponding flux is:

$$F = \begin{bmatrix} \rho \\ \rho u^2 + p \\ \rho u\,(e + u^2/2) + up \end{bmatrix} \tag{27}$$

The concern now arises as to what to do about the internal energy e, which is not explicitly solved for in the system of three equations. One solution is to solve the system redundantly, adding an extra equation for e as discussed in Colella, Glaz and Ferguson (1994). This essentially adds an equation of state to the system of equations. For continuum problems, we might want to adopt this approach to add other quantities, such as stresses, into the Riemann problem.

(c) **Make the approximation slightly nonlinear.** So far, we've talked about computing R at a single point to obtain $R((U_L + U_R)/2)$. Adding a small amount of nonlinear information can make the scheme much more robust.

4.4 Temporal evolution

4.4.1 First-order Godunov

As we've done before, the solution can be updated by first-order Godunov, an upwinding technique which also satisfies the entropy condition by dissipating rarefaction shocks. The solution looks like a series of constant states separated by discrete jumps:

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x}\left[F_{j-1/2} - F_{j+1/2}\right] \tag{28}$$

where the flux is given by:

$$F_{j+1/2} = F\left(U_{RP_{j+1/2}}\right) \tag{29}$$

and $U_{RP_{j+1/2}}$ is the solution to the Riemann problem along the ray $x / t = 0$ with $(U_L, U_R) = (U_{j-1}^n, U_{j+1}^n)$.

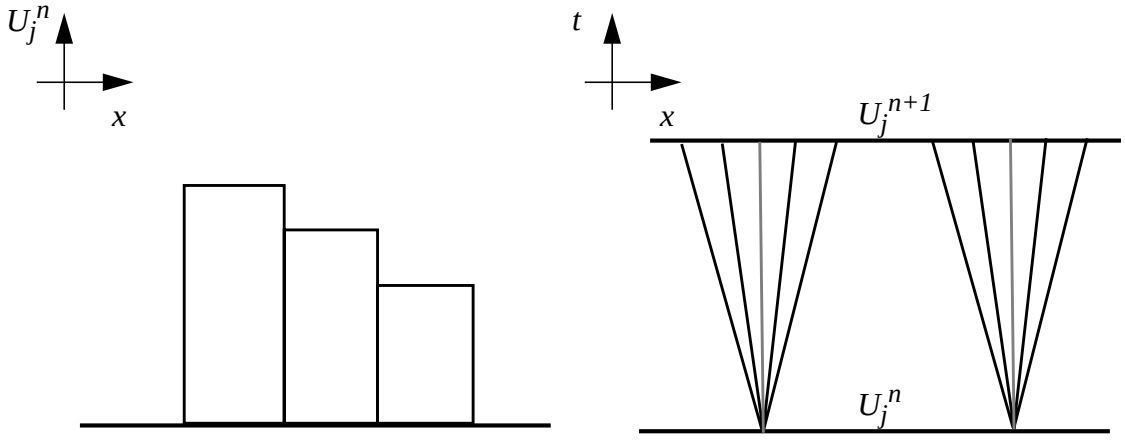


Figure 37. A Graphical representation of the the first-order Godunov solution procedure in the x-U and x-t planes.

4.4.2 Second-order Godunov

We can also solve this problem with higher-order accuracy by utilizing the predictor / corrector approach of second-order Godunov. The discrete evolution equation for the predictor step is found by taking an upstream-centered Taylor series expansion:

$$U_j^{n+1} = U_j^n + \frac{\Delta x}{2}\frac{\partial U}{\partial x}\bigg|_{j\Delta x,\, n\Delta t} + \frac{\Delta t}{2}\frac{\partial U}{\partial t}\bigg|_{j\Delta x,\, n\Delta t} + \ldots$$

$$\approx U_j^n + \frac{\partial U}{\partial x}\left[\frac{\Delta x}{2} - A(U)\frac{\Delta t}{2}\right]$$

$$U_j^{n+1} = U_j^n + \frac{\partial U}{\partial x}\Delta x \frac{1}{2}\left[I - A(U_j^n)\frac{\Delta t}{\Delta x}\right] \tag{30}$$

This is the same procedure that we used before. However, at this point we need to be a little more creative, because we cannot diagonalize the nonlinear operator on the RHS of the equation. Note that the form of the equation looks like a perturbation around the state U_j^n and recall from our previous discussion concerning the propagation of perturbations to the solution along characteristics that we can think about writing the first part of the second term as:

$$\frac{\partial U}{\partial x}\Delta x = \sum \alpha_k r_k \tag{31}$$

Using this in equation (30), we find that

$$U_j^{n+1} = U_j^n + \frac{1}{2}\left[I - A(U_j^n)\frac{\Delta t}{\Delta x}\right]\sum \alpha_k r_k$$

$$U_j^{n+1} = U_j^n + \sum_k \frac{1}{2}\left[I - \lambda_k \frac{\Delta t}{\Delta x}\right]\alpha_k r_k \tag{32}$$

This indicates that for $\lambda > 0$, signals propagate to the right along characteristics. Exactly as for the linear case, define the limiter as:

$$\Delta U_j = \Delta x \left.\frac{\partial U}{\partial x}\right|_{j\Delta x} \tag{33}$$

and the left, right and center amplitudes as

$$\alpha_{k_j}^L = l_k \cdot (U_j^n - U_{j-1}^n) \tag{34}$$

$$\alpha_{k_j}^C = \frac{1}{2} l_k \cdot (U_{j+1}^n - U_{j-1}^n) \tag{35}$$

$$\alpha_{k_j}^R = l_k \cdot (U_{j+1}^n - U_j^n) \tag{36}$$

and

$$\alpha_{k_j} = \begin{cases} (sign(\alpha_{k_j}^C))\, min(2|\alpha_{k_j}^L|, |\alpha_{k_j}^C|, 2|\alpha_{k_j}^R|) \ldots if \ldots (\alpha_{k_j}^L \alpha_{k_j}^R > 0) \\ 0 \ldots otherwise \end{cases} \tag{37}$$

Now compute the left and right states at the half step,

$$U_{j+1/2_L}^{n+1/2} = U_j^n + \frac{1}{2} P_+ \left(\left(I - A(U_j^n)\frac{\Delta t}{\Delta x}\right)(\Delta U_j)\right) \tag{38}$$

$$U_{j+1/2_R}^{n+1/2} = U_{j+1}^n - \frac{1}{2} P_- \left(\left(I + A(U_{j+1}^n)\frac{\Delta t}{\Delta x}\right)(\Delta U_{j+1})\right) \tag{39}$$

where the projection operators P_+ and P_- are defined by,

$$P_+(W) = \sum_{k|\,\tilde{\lambda}_k > 0} [l_k(U_j^n) \cdot W]\, r_k(U_j^n) \tag{40}$$

$$P_-(W) = \sum_{k|\,\tilde{\lambda}_k < 0} [l_k(U_j^n) \cdot W]\, r_k(U_j^n) \tag{41}$$

(We need to enlage our discussion of the projection operators.) Now solve the Riemann problem, using the left and right states defined by the equations above. Finally compute the fluxes and update using conservative finite differencing. We may want to use primitive variables or some other nonlinear change of variables and/or augment the variables to properly compute $U_{j+1/2}^{n+1/2}$

II. Incompressible Flow

5. Introduction

We now consider finite-difference methods for incompressible flow. The governing equations are conservation of mass,

$$\nabla \cdot \vec{u} = 0 \tag{1}$$

and conservation of momentum,

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\,\vec{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \vec{u} \tag{2}$$

where $\vec{u}$ is the velocity; ρ is density; p is pressure; and ν is the kinematic viscosity. The divergence operator in two-dimensional Cartesian coordinates is:

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right) \tag{3}$$

and the Laplacian is:

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{4}$$

The approach to the full momentum equation may be simplified by considering separately the following three subcategories:

(i) **Advection.** This is the problem which we studied extensively in Part I. However now we are going to consider the problem in more than one space dimension. The governing equation is:

$$\frac{\partial s}{\partial t} + (\vec{u} \cdot \nabla)\, s = 0 \tag{5}$$

For the model problem, we can think of s as some scalar. This step is the design point of the full approach, since the errors introduced by the hyperbolic methodology are the most troubling. The time-step restriction enters through a limitation on the CFL number, σ:

$$\sigma = \frac{|\vec{u}|\Delta t}{\Delta x} \leq C_a \tag{6}$$

where C_a is some constant value less than 1.

(ii) **Diffusion**. The diffusion equation is,

$$\frac{\partial s}{\partial t} = \varepsilon \Delta s \tag{7}$$

where ε represents some diffusion coefficient. If we were to use an explicit method to solve this equation, the time step restriction is:

$$\frac{\varepsilon \Delta t}{\Delta x^2} \leq C_d \tag{8}$$

(iii) **The continuity equation.** The continuity equation for incompressible flow can be thought of as a constraint on the velocity field, specifically that the velocity field is divergence-free. This constraint allows us to develop another model equation to examine the influence of the field induced by the pressure gradient. If we apply the divergence operator to the momentum equation (2) above, switch the order of differentiation, and utilize the continuity equation (1), we obtain:

$$\nabla \cdot [(\vec{u} \cdot \nabla) \vec{u}] = -\frac{1}{\rho} \Delta p \tag{9}$$

or, rewriting:

$$\Delta p = -\rho \nabla \cdot [(\vec{u} \cdot \nabla) \vec{u}] \tag{10}$$

which suggests the use of the model equation:

$$\Delta \phi = \rho \tag{11}$$

A discussion of step (i) above was given in Part I. To examine steps (ii) and (iii) above, we will be looking at fundamentally different solution techniques. Rather than explicitly solving for the temporal evolution of *u*, we will be discussing *implicit* methods. For example, to solve for *p* in equation (10) above, we could discretize the Laplacian operator, and solve the resulting system of linear equations to obtain *p*. Since the inversion will likely become a rather cumbersome process involving large matrices, we would be wise to also explore iterative methods and multigrid. We first look at the Poisson equation in section 6., and in section 7. explore the model advection equation.

6. The Poisson equation.

We want to focus our attention on the Poisson equation:

$$\Delta \phi = \rho \tag{12}$$

We can think of the quantity ϕ in more general terms as representing a potential field, and ρ as the associated charge density which induces that field. Or we can think of a linear distribution of temperature at steady-state caused by diffusion and a heat source term. Although we will discuss only the two-dimensional case, the extension to three dimensions is just a matter of bookkeeping. We want to define ρ and ϕ at the cell centers on a square 2D grid with equal mesh spacing, *h*, in both dimensions, as shown in Figure 37 below.

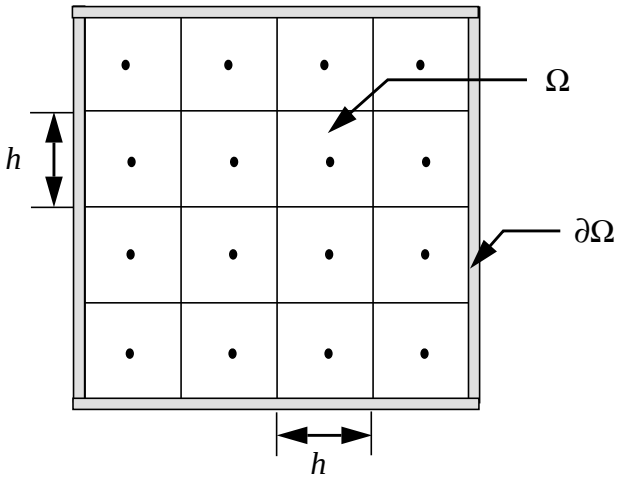


Figure 38. General setup of domain.

The dimensions in physical space will be denoted by $\underset{\sim}{x} = (x, y)$ and range from *0* to *1*, while the dimensions in logical space will be $\underset{\sim}{i} = (i, j)$ and extend from *1* to *N* in both directions. We will consider two different sets of boundary conditions: doubly periodic ($\phi(0, y) = \phi(1, y)$; $\phi(x, 0) = \phi(x, 1)$), and initially homogeneous Dirichlet conditions ($\phi = 0$ on $\partial\phi$). These boundary conditions will make Fourier analysis and numerical implementation simpler. At the moment, we will punt on the question of how to make the world conform to our choice of boundary conditions (see section xxx.xx for a discussion of boundary conditions).

To discretize the Laplacian,

$$\Delta\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}$$

$$\approx \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{h^2} + \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{h^2}$$

or, introducing new notation for the discretized form of the standard five-point Laplacian,

$$(\Delta^h \phi)_{ij} = \frac{\phi_{i+1,j} + \phi_{i-1,j} - 4\phi_{i,j} + \phi_{i,j+1} + \phi_{i,j-1}}{h^2}$$

On the interior of the domain, the discretized form of Poisson's equation is:

$$(\Delta^h \phi)_{ij} = \rho_{ij} \tag{13}$$

The boundary conditions will dictate how we deal with the points near the boundary. For periodic BC, we will extend the solution in a periodic way, e.g., $\phi_{0,j} = \phi_{N,j}$. For Dirichlet BC, we will extend the domain and linearly extrapolate so that the value of ϕ at the boundary is nil, i.e., $\phi_{o,j} = -\phi_{1,j}$, as shown in figure +1 below:

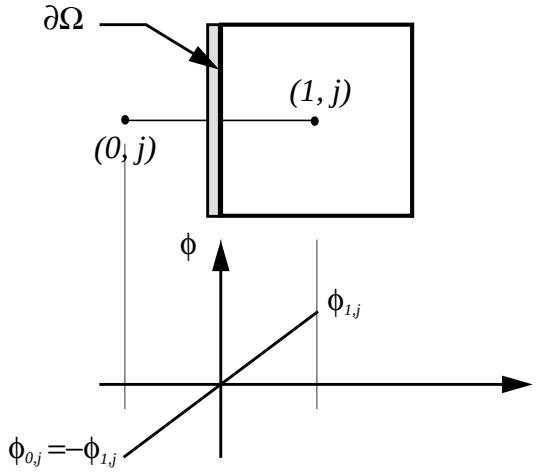


Figure 39Treatment of Dirichlet boundary conditions

Each cell in the interior must satisfy the Poisson equation, while each cell on the edge of the domain must satisfy boundary conditions. The result is that we end up with a big matrix equation of the form:

$$[A]\underset{\sim}{\phi} = \underset{\sim}{\rho} \tag{14}$$

We can solve for $\underset{\sim}{\phi}$ by calculating $\underset{\sim}{\phi} = [A]^{-1}\rho$. Therefore, we want to invert the matrix *[A]* with as little grief as possible. We next examine the merits of direct solvers, iterative and multigrid methods.

6.1 Direct solvers

One possibility is using brute force in the form of Gaussian elimination with partial pivoting and *LU* decomposition. (Further discussion of this technique is available in any number of classical books on numerical methods.) But this turns out to be a poor option since there are N^2 equations and N^2 unknowns. The number of operations for a general linear

system would be of $O((N^2)^3) = O(N^6)$. This is unacceptably large, especially if we contrast this with that of a hyperbolic problem, which is more typically of $O(N^2)$ operations. We note that the structure of the matrix depends on the order in which we write the equations. If we set up the system with a modicum of wisdom, we can obtain a matrix which is dense in nonzero values near the diagonals and full of zeroes in the upper left and lower right corners. It is easier to invert such a matrix than a fully populated one. The bandwidth, i.e., the number of lines of nonzero values off the diagonal, is of order N for Dirichlet boundary conditions, as shown in figure +1, below.

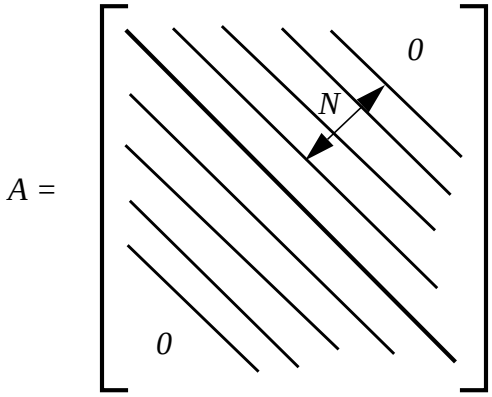


Figure 40Bandwidth structure of the matrix *A*

The number of operations needed for the LU decomposition of this matrix is roughly the number of equations multiplied by the square of the bandwidth, or approximately $O(N^4)$. We also have to consider that the result of the LU decomposition will need to be stored in $O(N^3)$ floating-point memory locations. By comparison, the explicit hyperbolic solver to which this scheme will be coupled requires $O(N^2)$ storage locations, and the same number of floating point operations. For this reason, we seek out methods for which the computational effort and storage required is closer to that of for the hyperbolic solver. We are left with the task of examining iterative solutions, the topic of the next section.

6.2 Iterative solvers

Ultimately, we want to solve the steady-state Poisson equation:

$$\Delta\phi = \rho \tag{15}$$

For iterative solutions, we form the associated time evolution equation for $\tilde{\phi}$ and carry the solution to "steady state". Instead of solving immediately for $\phi(\vec{x})$, we will first consider $\tilde{\phi}(\vec{x}, t)$ which satisfies:

$$\frac{\partial \tilde{\phi}}{\partial t} = \Delta \tilde{\phi} - \rho \tag{16}$$

We expect that as $t \to \infty$, that $\tilde{\phi}(\vec{x}, t) \to \phi(\vec{x})$. How can be convinced of this statement? Let's define the remainder as:

$$\delta = \tilde{\phi} - \phi \tag{17}$$

What equation does δ satisfy? Since $\tilde{\phi} \neq \phi(t)$ and the system is linear, it is easy to show that

$$\frac{\partial \delta}{\partial t} = \Delta \delta \tag{18}$$

For doubly periodic boundary conditions, we can use a Fourier transform to diagonalize the system and show that all Fourier modes decay as $t \to \infty$. We ought to point out at this point that "time" has a different meaning in this discussion as compared with explicit solutions of unsteady hyperbolic problems. We're not really marching in time for this case, but rather we're iterating somewhat abstractly to obtain the steady-state solution to Poisson's equation. With this clarification in mind, if we let the iterative index be l and the time step be λ, we can write the discretized iterative equation for $\tilde{\phi}$ as:

$$\tilde{\phi}^{l+1} = \tilde{\phi}^{l} + \lambda (\Delta^h \tilde{\phi}^{l} - \rho) \tag{19}$$

Since we will be solving the above equation iteratively as the simulation time $t \to \infty$, we would like the time step λ to be as large as possible.

6.2.1 Convergence and stability

If we define the remainder δ^l:

$$\delta^l = \tilde{\phi}^{l} - \phi \tag{20}$$

and use the definition $\Delta^h \phi = \rho$, we can show that δ must satisfy:

$$\delta^{l+1} = \delta^l + \lambda \Delta^h \delta^l \tag{21}$$

The question then becomes one of how quickly does the solution converge so that $\delta^l \to 0$. More formally, for some operator L where:

$$L\delta = (I + \lambda \Delta^h)\, \delta \tag{22}$$

Does $L\delta$ reduce the norm of δ? Is it true that:

$$\| L\delta \| \leq \| \delta \| \tag{23}$$

The natural norm to choose is the L_2 norm, defined by:

$$\|\Phi\|_2^2 = h^2 \sum |\Phi_{i,j}|^2 \tag{24}$$

where the summation is over the entire domain. For doubly periodic boundary conditions, Fourier analysis indicates:

$$\delta_{j_x, j_y} = \sum_{k_x = -\frac{N}{2}+1}^{\frac{N}{2}+1} \sum_{k_y = -\frac{N}{2}+1}^{\frac{N}{2}+1} a_{k_x} a_{k_y} e^{2\pi i (k_x j_x + k_y j_y) h} \tag{25}$$

where index space is temporarily denoted by (j_x, j_y) rather than (i, j) to avoid confusion with the $i = \sqrt{-1}$ in the exponential expression of equation (-1) above. Then

$$(L\delta)_{j_x, j_y} = \sum_{k_x = -\frac{N}{2}+1}^{\frac{N}{2}+1} \sum_{k_y = -\frac{N}{2}+1}^{\frac{N}{2}+1} a_{k_x} a_{k_y} \Lambda(\beta_x, \beta_y) e^{2\pi i (k_x j_x + k_y j_y) h} \tag{26}$$

where the wavenumber in the x direction is defined as:

$$\beta_x = 2\pi i k_x h \tag{27}$$

and similarly in the y direction:

$$\beta_y = 2\pi i k_y h \tag{28}$$

We also define Λ as the result of extending the shift operation to two dimensions, i.e., Λ is the symbol of the scheme (recall the discussion of symbols in section xxx.xx.x). From a simpleminded comparison of equations (-3) and (-4) above, we see that our condition for stability is met (i.e., as we slap L on δ, the norm decreases), if $\Lambda < 1$. If the operator L represents the iterative equation (22) above and using the standard five-point cell-centered Laplacian, then we can write the operator as:

$$L = I + \frac{\lambda}{h^2} [S_x + S_x^{-1} + S_y + S_y^{-1} - 4I] \tag{29}$$

where S's represent shifts and I is the identity. The symbol $\Lambda(\beta_x, \beta_y)$ of the scheme is:

$$\begin{aligned} \Lambda(\beta_x, \beta_y) &= 1 + \frac{\lambda}{h^2} [e^{i\beta_x} + e^{-i\beta_x} + e^{i\beta_y} + e^{-i\beta_y} - 4] \\ &= 1 + \frac{\lambda}{h^2} [2\cos\beta_x + 2\cos\beta_y - 4] \end{aligned} \tag{30}$$

To obtain $|\Lambda| \leq 1$ we need

$$-2 \le \frac{\lambda}{h^2} \left[2\cos\beta_x + 2\cos\beta_y - 4 \right] \le 0 \tag{31}$$

It is clear that the expression in brackets above is never positive. Therefore, we need to show that

$$\frac{\lambda}{h^2} \left[2\cos\beta_x + 2\cos\beta_y - 4 \right] \ge -2 \tag{32}$$

Since the minimum value on cosine is -1, we can show that the condition in equation (32 above is met when

$$-\frac{8\lambda}{h^2} > -2$$

or

$$\frac{\lambda}{h^2} < \frac{1}{4} \tag{33}$$

When the time step is less than $h^2/4$, all of the Fourier modes of δ eventually decay. How many iterations will this take? How much does each Fourier amplitude decrease as a function of β_x and β_y?

One thing which we can do is examine the smooth parts of the solution. This means looking at the long-wavelength limit of the solution as $\beta_x, \beta_y \rightarrow 0$. Using a Taylor expansion about $\beta_x = \beta_y = 0$:

$$\Lambda(\beta_x, \beta_y)\big|_{\beta_x = \beta_y = 0} = 1 + \frac{\lambda}{h^2} \left\{ 2\left[1 - \frac{\beta_x^2}{2} + \ldots \right] + 2\left[1 - \frac{\beta_y^2}{2} + \ldots \right] - 4 \right\}$$

$$\approx 1 - \frac{\lambda \left[\beta_x^2 + \beta_y^2 \right]}{h^2}$$

$$= 1 - \frac{\lambda \left[2\pi \right]^2 \left[(k_x h)^2 + (k_y h)^2 \right]}{h^2}$$

or

$$\Lambda(\beta_x, \beta_y)\big|_{\beta_x = \beta_y = 0} \approx 1 - 4\lambda\pi^2 \left[k_x^2 + k_y^2 \right] \tag{34}$$

From stability considerations, we have shown in equation (-2) above that $\lambda \le \frac{h^2}{4}$. We can then deduce that the dependence of the symbol on the mesh spacing is:

$$\Lambda(0, 0) \approx 1 - Ch^2 \tag{35}$$

where C is some constant. This is a slow rate of convergence.

We can also consider the behavior of high-wavenumber modes. We note that the highest wavenumber is π. This means that if $\beta_x = \beta_y = \pi$, the symbol becomes:

$$\Lambda(\pi, \pi) = 1 + \frac{\lambda}{h^2}[2(-1) + 2(-1) - 4]$$

$$= 1 - \frac{8\lambda}{h^2}$$

However, if $\lambda = \frac{h^2}{8}$, we arrive at the result:

$$\Lambda(\pi, \pi)\big|_{\lambda = h^2/8} = 0 \tag{36}$$

This finding indicates that this scheme is remarkably efficient at damping out the high-wavenumber modes with the right value of λ. In practice, we find that modes in the range $\frac{\pi}{2} \le \beta_x, \beta_y \le \pi$ decay rapidly (within a few iterations) if we choose $\lambda = h^2/8$. Therefore, it is really the long wavelengths which we must worry about.

Note that "long" and "short" wavelengths are measured relative to the mesh spacing h. We would expect short-wavelength disturbances to decay within a few iterations. The remainder δ could at that point be represented accurately on a coarser mesh, say double the original mesh spacing. If we then ran for a few iterations on this coarser mesh, we'd get rid of somewhat longer wavelengths. The idea, then, is to relax on a series of continually coarser grids in order to destroy successively longer-wavelength components of the remainder. (We have assumed (so far) that the total number of mesh points is a power of 2.)

Place somewhere relevant: For further discussion of the multigrid technique, see Briggs (19xx). "A Multigrid Tutorial." SIAM Publications.

We can also talk about the residual problem associated with Poisson's equation. Assume we have an estimate of $\tilde{\phi}$ at some time. The residual matrix R is given by:

$$R = -\Delta^h \tilde{\phi} + \rho \tag{37}$$

Using the residual as a source term, we can solve for:

$$\Delta^h \delta = R \tag{38}$$

and then use the result to update the solution:

$$\phi = \tilde{\phi} + \delta \tag{39}$$

These steps are equivalent to solving $\Delta^h \phi = \rho$. How can we be sure of this?

Proof: What is the solution to $\Delta^h \delta = R$?

$$\begin{aligned} \Delta^h \delta &= R \\ &= \rho - \Delta^h \tilde{\phi} \\ \delta &= \Delta^{-1}\rho - \tilde{\phi} \end{aligned}$$

Using the definition of δ:

$$-\tilde{\phi} + \Delta^{-1}\rho = \Delta^{-1}\rho - \tilde{\phi}$$

We are being consistent.

6.2.2 Procedure for implementation of multigrid

We would like to solve the discrete Poisson equation $\Delta^h \phi = \rho$ using multigrid on a square mesh of dimension $N \times N$, where $N = 2^m$. We will discuss the mechanics of implementing this scheme by considering the subroutine

MG_relax(ρ, h, l, $\tilde{\phi}$)

The inputs are: ρ, which is a generic forcing term on the RHS.; h is the mesh spacing, l is the current level (which represents the coarseness of the mesh); and $\tilde{\phi}$ is the first guess at the solution ϕ. The subroutine returns a new approximation to ϕ. Within the subroutine, the following steps are taken:

(i) **Relaxation scheme:** Update the guess for $\tilde{\phi}$ over all points *(i, j)* by the local relaxation scheme. So far, the only one we have considered is of the form:

$$\tilde{\phi}^{l+1} = \tilde{\phi}^{l} + \lambda\,(\Delta^h \phi^l - \rho) \tag{40}$$

Perform this operation p times. The result is that we get rid of the high-frequency components of the remainder δ.

(ii) **Calculate the residual:** As defined previously,

$$R = \rho - \Delta^h \phi \tag{41}$$

We know that the current guess satisfies $\Delta^h \delta = R$. In addition, all high-wavenumber components of R are damped if the high-wavenumber components of δ are damped, since this is an elliptic problem. Thus, R and δ can be represented accurately on a coarser grid.

(iii) **Average the residual onto a coarser grid:** Let R^c denote the averaged R on the coarser grid. As a matter of fact, let's erupt in a flurry of new notation: let the new coarse level be $l^c = l - 1$; $h^c = 2h$; and the new first guess on the coarse grid to be $\tilde{\delta}^c = 0$. Dimensionally, the source term R looks like a charge; we would like to conserve charge in forming R^c. We have specified that the mesh spacing h is equivalent in both the x and y directions. The interpolated value for the residual on the coarse mesh can then be seen as the average value of the component cells of the fine mesh, as shown in figure +1.

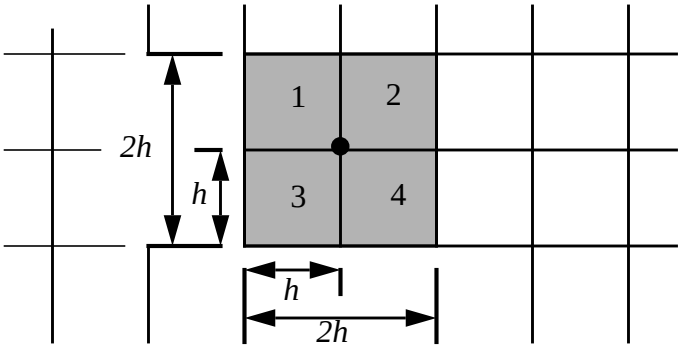


Figure 41Calculation of residual on the coarse grid

We can then calculate the residual on the coarse grid with mesh spacing *2h x 2h* from the fine-grid mesh cells 1 through 4 by:

$$R^c = \frac{R^1 + R^2 + R^3 + R^4}{4} \tag{42}$$

Note that if the mesh spacing is not uniform, a logical approach would be to use weighting factors which account for the differing areas on the fine-grid cells.

(iv) **Solve for the coarse-grid correction recursively.** As long as the level $l > 0$, we call the MG_relax routine recursively:

$$\delta^c = \text{MG\_relax}(R^c, h^c, l^c, \tilde{\delta}^c)$$

The effect of this step is to solve $\Delta^{2h}\delta^c = R^c$, thereby approximating δ on a coarser grid.

(v) **Interpolate the coarse-grid correction onto the finer grid, one level at a time.** We would like the interpolation operator I to be the adjoint of the averaging operator. For the averaging operator used above, the interpolation operator is simply $\delta^c = \delta_1 = \delta_2 = \delta_3 = \delta_4$. The next step is to update $\tilde{\phi} = \tilde{\phi} + \delta$.

(vi) **Relax.** In this context, "relax" means that we perform

$$\tilde{\phi}^{l+1} = \tilde{\phi}^{l} + \lambda\,(\Delta^h \phi^l - \rho)$$

using the updated value of $\tilde{\phi}$ of step (5).

We know from stability considerations that λ must be less than $h^2/4$, and that if $\lambda = h^2/8$, short-wavelength oscillations are readily suppressed. Can we do better? What is the optimal point relaxation scheme? What happens if Δ^h is not invertible? These topics are addressed in the following sections.

6.2.2.1 Point Jacobi iteration

To update for new values of $\tilde{\phi}_{ij}$, using point Jacobi iteration:

(43)

$$\tilde{\phi}^{l+1}_{i,j} = \tilde{\phi}^{l}_{i,j} + \lambda\left[\,(\Delta^h \tilde{\phi}^{l})_{i,j} - \rho_{i,j}\right]$$

If we choose the maximum allowable time step, $\lambda = h^2/4$. Using a five-point cell-centered Laplacian, equation (-1) becomes:

$$\begin{aligned}\phi^{l+1}_{i,j} &= \phi^{l}_{i,j} + \frac{h^2}{4}\left\{\frac{[\tilde{\phi}^{l}_{i,j+1} + \tilde{\phi}^{l}_{i,j-1} + \tilde{\phi}^{l}_{i+1,j} + \tilde{\phi}^{l}_{i-1,j} - 4\tilde{\phi}^{l}_{i,j}]}{h^2} - \rho_{i,j}\right\} \\ &= \frac{1}{4}[\tilde{\phi}^{l}_{i,j+1} + \tilde{\phi}^{l}_{i,j-1} + \tilde{\phi}^{l}_{i+1,j} + \tilde{\phi}^{l}_{i-1,j}] - \rho_{i,j}\frac{h^2}{4}\end{aligned} \tag{44}$$

If we define

$$\delta^l = \tilde{\phi}^{l} - \phi \tag{45}$$

where ϕ is the exact solution to $\Delta^h \phi = \rho$, then

$$\delta_{i,j}^{l+1} = \frac{1}{4}\left[\delta_{i,j+1}^{l} + \delta_{i,j-1}^{l} + \delta_{i+1,j}^{l} + \delta_{i-1,j}^{l}\right] \tag{46}$$

We note that we have lost one entire subtraction by using this form relative to equation (-3). The chief advantage of looking at δ rather than ϕ is that we can set the boundary values of δ to zero for the case of any Dirichlet boundary conditions, which can simplify the numerics. In either case, we would expect high frequencies to be readily damped in a simulation using this iterative procedure, thus providing lots of smoothing. This is desirable; however, there is an important drawback to this formulation. Through this choice of time step, we've removed the dependence on the current value of $\tilde{\phi}$ at point *(i, j)*; we've also opened the door to a major headache because we've removed realistic communication between adjacent cells. Consider a checkerboard pattern of red and black. With this formulation, the red cells do not "talk" to the black cells and vice versa. If, for example, $\delta\phi$ attains the following values in the cells in figure +1:

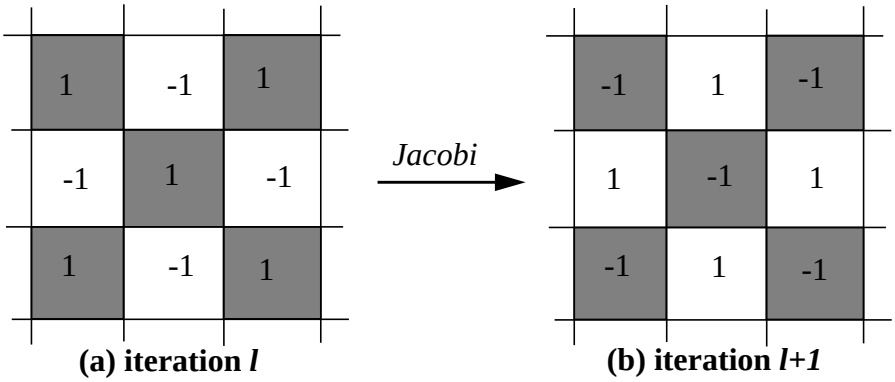


Figure 42Checkerboard pattern which spells trouble for point Jacobi

If this pattern is extended, the conscientious reader can verify that the cells flip helplessly from one value to the other with each iteration.

6.2.2.2 Gauss-Seidel relaxation with red/black ordering (GSRB)

A half-step incremental approach is used in Gauss-Seidel relaxation for which the values are updated in separate sweeps. Point-Jacobi iteration is used on half of the cells on the first pass, and on the other group on the second pass. In the first step:

$$\tilde{\phi}_{i,j}^{l+1/2} = \begin{cases} \tilde{\phi}_{i,j}^{l} + \lambda\left[\left(\Delta^{h}\tilde{\phi}^{l}\right)_{i,j} - \rho_{i,j}\right] \ldots if \ldots i+j = even \\ \tilde{\phi}_{i,j}^{l} \ldots if \ldots i+j = odd \end{cases} \tag{47}$$

Next, in step two:

$$\tilde{\phi}_{i,j}^{l+1} = \begin{cases} \tilde{\phi}_{i,j}^{l+1/2} \ldots if \ldots i+j = even \\ \tilde{\phi}_{i,j}^{l+1/2} + \lambda \left[(\Delta^h \tilde{\phi}^{l+1/2})_{i,j} - \rho_{i,j} \right] \ldots if \ldots i+j = odd \end{cases} \tag{48}$$

Or, for $\lambda = h^2/4$ and the standard five-point Laplacian:

$$\delta_{i,j}^{l+1/2} = \begin{cases} \frac{1}{4} \left[\delta_{i,j+1}^{l} + \delta_{i,j-1}^{l} + \delta_{i+1,j}^{l} + \delta_{i-1,j}^{l} \right] \ldots if \ldots i+j = even \\ \delta_{i,j}^{l} \ldots if \ldots i+j = odd \end{cases} \tag{49}$$

and

$$\delta_{i,j}^{l+1} = \begin{cases} \delta_{i,j}^{l+1/2} \ldots if \ldots i+j = even \\ \frac{1}{4} \left[\delta_{i,j+1}^{l+1/2} + \delta_{i,j-1}^{l+1/2} + \delta_{i+1,j}^{l+1/2} + \delta_{i-1,j}^{l+1/2} \right] \ldots if \ldots i+j = odd \end{cases} \tag{50}$$

The goal of using GSRB is to transfer the error from high to low wavenumbers; we will then let the coarser grids upon application of multigrid worry about the low-wavenumber errors. Ultimately, the red and black cells are linked via the boundary conditions. The checkerboard picture which goes along with Gauss-Seidel with red/black ordering is shown in figure +1 below:

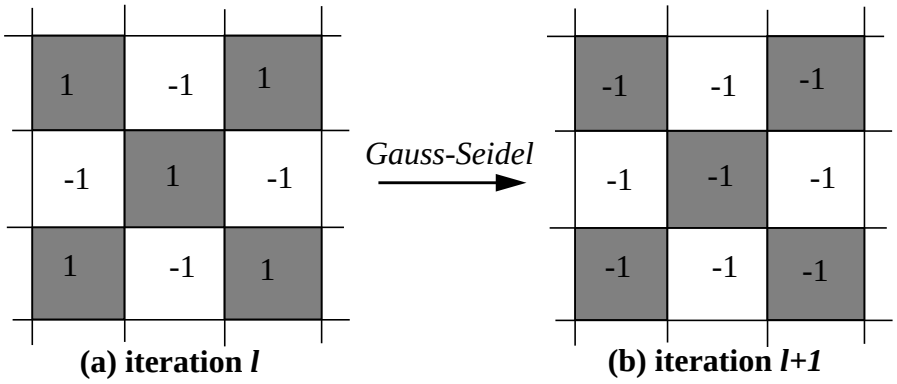


Figure 43Updating the checkerboard using Gauss-Seidel relaxation

Note that this method extremely efficient at *eliminating*, not just damping, modes in the range of $\frac{\pi}{2} \le \dot{\beta}_x, \beta_y \le \pi$. We must also note that GSRB damps all other modes to some extent. However, it does not increase the error, i.e.,

$$\| \delta^{l+1} \| < \| \delta^{l} \| \tag{51}$$

We conclude that we can live with this since we are more worried about damping high wavenumber error and, after all, GSRB is stable, which is most desirable.

6.2.2.3 Solvability conditions -- the Fredholm alternative

We want to examine the consequences if Δ^h is not invertible. Given any rectangular matrix *[A]* (not even necessarily square any more), we would like to solve:

$$A\underset{\sim}{y} = \underset{\sim}{f} \tag{52}$$

This has a solution if and only if $\underset{\sim}{f}$ is orthogonal to the null space of A^T.[1] Stated another way, a solution exists if and only if, for all w for which $A^T w = 0$, the inner product $\langle \underset{\sim}{f}, \underset{\sim}{w} \rangle = 0$.

The necessity of this condition is easy to show. Given $A\underset{\sim}{y} = \underset{\sim}{f}$

$$A^T \underset{\sim}{w} = 0$$

The inner product of $\langle \underset{\sim}{f}, \underset{\sim}{w} \rangle$ is:

$$\begin{aligned} \langle \underset{\sim}{f}, \underset{\sim}{w} \rangle &= \langle A\underset{\sim}{y}, \underset{\sim}{w} \rangle \\ &= \langle \underset{\sim}{y}, A^T \underset{\sim}{w} \rangle \\ &= \langle \underset{\sim}{y}, 0 \rangle \\ &= 0 \end{aligned}$$

It is a little trickier to show that this is sufficient for a general matrix A. However, for a symmetric matrix the proof is straightforward. Assume A is $p \times p$. We find a basis which diagonalizes A, using the right-eigenvector matrix, R so that:

$$R^{-1} A R = \begin{bmatrix} \lambda_1 & & 0 \\ & \cdots & \\ 0 & & \lambda_p \end{bmatrix} \tag{53}$$

with

$$R = (r_1, \ldots, r_p) \tag{54}$$

If, say, $\lambda_p = 0$, then we run into trouble when we try to solve $v_p = f_p / \lambda_p$.

We must also be aware that solutions are not unique. Assuming that $A\underset{\sim}{w} = 0$ (a different $\underset{\sim}{w}$ this time), we know that $A\underset{\sim}{y} = \underset{\sim}{f} = A(\underset{\sim}{y} + \underset{\sim}{w})$ for linear systems. The value of $\underset{\sim}{y}$ can be determined up to a constant. However, the nonuniqueness is not a problem in the physical application we are looking at. Recall that the motivation behind all of this is to solve Poisson's equation:

1. $\underset{\sim}{x}$ is in the null space of the matrix A if $Ax_i = 0$. E.g., for Poisson's equation with periodic BC, $\phi = Const$ is in the nullspace of Δ^h

$$\Delta\phi = \rho \tag{55}$$

However, we may not really be interested in ϕ; what we may want to find out about is the field induced by ϕ, i.e., $\nabla\phi$. For example, in the case of incompressible fluids, this is the pressure gradient. We know that the addition of a constant to ϕ does not change the value of $\nabla\phi$.

There are two important implications of this analysis relevant to our discussion of point relaxation. First, we effectively toss out the nullspace component of the solution by looking at its gradient. Secondly, even if there is some component of null space in the initial guess of ϕ, then we expect that that component will not grow. We want to iteratively update ϕ by:

$$\phi^{l+1} = \phi^l + \lambda (A\phi^l - f) \tag{56}$$

Let's examine the RHS. of equation (-1) step by step. A applied to null space is zilch; f is orthogonal to null space. Therefore, there are no terms on the RHS which could increase the null space amplitude (except for roundoff error). Checking for the increase of ϕ in null space can be used as a handy debugging tool.

Let's apply this knowledge to the solution of Poisson's equation by multigrid. In that case, the matrix A represents the discretization of the Laplacian, Δ^h. The homogenous part of the solution corresponding to null space is $\phi^h = C$, where C is some constant. The solvability condition is that the inner product

$$\langle \rho, 1 \rangle = \sum \rho_{ij} = 0$$

Physically, this means that there is no net change in the charge. Examining the multigrid procedure step by step:

(i) **Relax.** There is no term on the RHS which increases the null space of ϕ when relaxing by: $\phi^{l+1} = \phi^l + \lambda (A\phi^l - f)$, as discussed above

(ii) **Compute residual.** By calculating $R = \rho - \Delta^h\phi$, we know that R satisfies the solvability condition.

(iii) **Average residual onto coarse grid.** Solvability is not violated when $R^c = [R_1 + R_2 + R_3 + R_4] / 4$

(iv) **Relax for coarse correction**. δ^c has no null space component.

(v) **Interpolate.** The fine-grid correction $\delta = I\delta^c$ has no null space components, so that the new $\phi^{l+1} = \phi^l + \delta^c$ will not increase the null space.

6.2.2.4 Boundary conditions

So far, we've considered only homogeneous boundary conditions. What if we have inhomogeneous Dirichlet or Neumann boundary conditions? It is simple to convert the inhomogeneous boundary conditions to homogeneous BC's with a modified right-hand side. Let's examine the case of inhomogeneous Dirichlet boundary conditions. We want to solve $\Delta\phi = \rho$ on the domain D subject to the boundary conditions $\phi = g$ on ∂D. Using cell-centered discretization as usual, we write the discretized form of the differential equation as:

$$\Delta^h \phi^h = \rho^h \tag{57}$$

Consider the left edge of the domain at $i_0 - 1/2$, as shown below in figure +1:

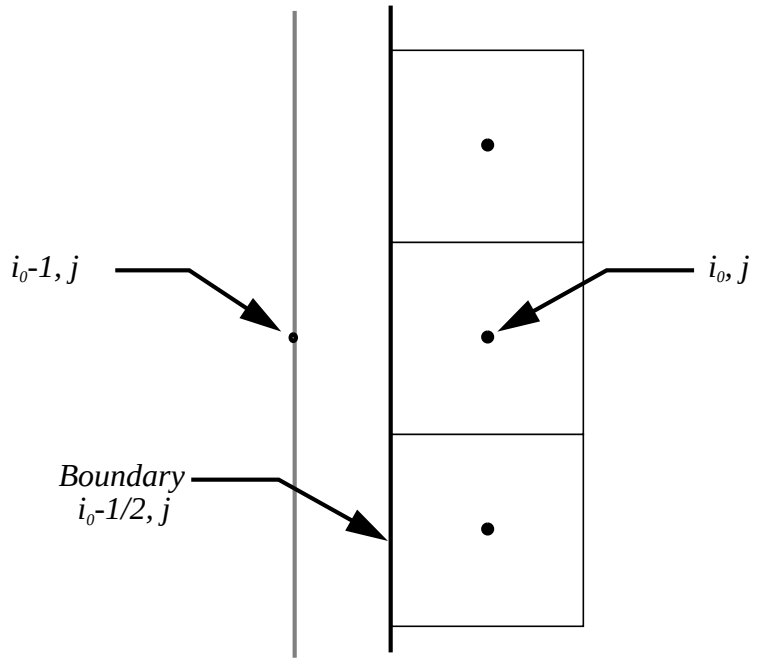


Figure 44Left boundary...

We know from the boundary conditions that in the discrete domain at the boundary, ϕ is given exactly by:

$$\phi^h_{i_0 - 1/2, j} = g^h_{i_0 - 1/2, j} \tag{58}$$

where

$$g^h_{i_0 - 1/2, j} = g\left((i_0 - 1/2)\, h, jh \right) \tag{59}$$

We can also make a first-order finite-difference approximation considering the phantom node located one point inside the boundary:

$$g^h_{i_0-1/2,j} = \frac{\phi^h_{i_0-1,j} + \phi^h_{i_0,j}}{2} \tag{60}$$

so that

$$\phi^h_{i_0-1,j} = 2g^h_{i_0-1/2,j} - \phi^h_{i_0,j} \tag{61}$$

What does all of this notation buy us (even if it doesn't cost anything)? We claim that we can find a boundary charge ρ^B_{ij} from this mess which we can simply add to the original right-hand side to obtain an equivalent problem for ϕ. For the column of nodes one point inside the domain, at $i = i_{0:}$

$$\Delta^h\phi^h = \rho^h_{ij} + \rho^B_{ij}$$

$$= \frac{1}{h^2}(\phi^h_{i_0-1,j} - 2\phi^h_{i_0,j} + \phi^h_{i_0+1,j}) + \frac{1}{h^2}(\phi^h_{i_0,j-1} - 2\phi^h_{i_0,j} + \phi^h_{i_0,j+1})$$

$$= \frac{1}{h^2}((2g^h_{i_0-1/2,j} - \phi^h_{i_0,j}) - 2\phi^h_{i_0,j} + \phi^h_{i_0+1,j}) + \frac{1}{h^2}(\phi^h_{i_0,j-1} - 2\phi^h_{i_0,j} + \phi^h_{i_0,j+1})$$

For Dirichlet boundary conditions, $\phi^h_{i_0-1,j} = -\phi^h_{i_0,j}$. We then can conclude that the left edge, the boundary charge is given by:

$$\rho^B_{i_0,j} = -\frac{2g^h_{i_0-1/2,j}}{h^2} \tag{62}$$

Similarly, on the bottom edge:

$$\rho^B_{i,j_0} = -\frac{2g^h_{i,j_0-1/2}}{h^2} \tag{63}$$

At the bottom left corner, the charge is:

$$\rho^B_{i_0,j_0} = -\frac{2}{h^2}(g^h_{i_0-1/2,j} + g^h_{i,j_0-1/2}) \tag{64}$$

We can perform a similar analysis for the right and top edges.

The same game can be played with Neumann boundary conditions, i.e., if we are given $\frac{\partial\phi}{\partial\hat{n}} = g$, where $\hat{n}$ is the outward normal. The expression for $\phi^h_{i_0-1,j}$ is in this case:

$$\phi^h_{i_0-1,j} = 2hg^h_{i_0-1/2,j} + \phi^h_{i_0,j} \tag{65}$$

6.2.3 Performance of multigrid

Now that we have the mechanics of multigrid firmly in place (presumably), we need to examine the performance of this technique. We initially start out with some guess $\tilde{\phi}$ for the solution ϕ and calculate the norm of the residual, as defined by:

$$RNORM = \left\| \Delta^h \phi - \rho \right\| \tag{66}$$

Then we must set some tolerance criterion, ε, which is a function of the machine's round-off error. Then, as long as *RNORM* is larger than the tolerance, keep using the multigrid algorithm recursively. A section of pseudo-code of the outer loop might look like:

$\tilde{\phi}$ = *[first guess]*

$RNORM^l = \left\| \Delta^h \phi - \rho \right\|$

while (RNORM $> \varepsilon \| \rho \|$*)*

$l = l + 1$

ϕ = *MG_relax(*ρ, *h, l,* $\tilde{\phi}$*)*

$RNORM^l = \left\| \Delta^h \phi - \rho \right\|$

end while

Next we want to examine issues such as convergence properties, cost and accuracy.

(i) Convergence properties of outer loop

A remarkable fact which can even be proven analytically for some cases, as well as being actually observed in numerical simulations is that:

$$RNORM^{l+1} \leq C \cdot RNORM^l \tag{67}$$

where C is a constant less than one and is a function of the mesh spacing h. For the five-point discretization of the Laplacian that we have been using, $C \cong 0.2$. Other common discretizations yield values in the ballpark of 0.2 - 0.4.

(ii) Cost of computation for a fixed tolerance

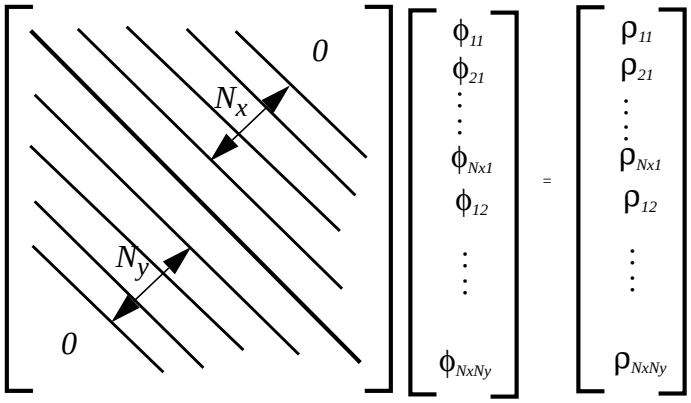
On each level, the number of floating point operations per iteration is on the order of the number of mesh points at that level. If K is denoted as the number of floating point operations and the number of mesh points in each direction is $N = 2^p$, then

$$K \sum_{q=0}^{p} 2^{2p} \leq K' N \, (log_2 N) \tag{68}$$

(iii) **Number of mesh points $N \neq 2^p$.**

When the number of grid points is not a power of two, represent the number of mesh points in each direction by the largest common power of 2 multiplied by an odd integer, N_x^B and N_y^B (i.e., $N_x = 2^p N_x^B$ and $N_y = 2^p N_y^B$). Let's assume that $N_x > N_y$. The next step is to call MG_relax(ρ, h, l, N_x, N_y, $\tilde{\phi}$). As long as $N_x > N_x^B$ and $N_y > N_y^B$, it makes sense to coarsen the grid and apply the usual multigrid algorithm. When we finally arrive at the coarsest level of N_x^B and N_y^B, then it's time to do a direct solve, using, e.g., Gaussian elimination. The matrix equation can be ordered thusly:

(69)



The matrix A is banded. Upon LU decomposition, the lower diagonal matrix L and the corresponding upper diagonal matrix U will both have a bandwidth of about N_x. The storage for these matrices is the product of the number of unknowns multiplied by the bandwidth, so that the storage should be of $O(N_x^2 N_y)$. In terms of floating-point work, the most expensive

part will be in performing the *LU* decomposition, requiring $O(N_x^3 N_y)$ floating-point operations. However, this only needs to be done once. The backsolve requires about $O(N_x^2 N_y)$ floating-point operations.

(iv) **Accuracy**.

Let's develop some more notation. If the exact continuous solution and source terms are denoted by the subscript *E*, then they are related by the differential equation:

$$\Delta\phi_E = \rho_E \tag{70}$$

We will call the exact continuous solution evaluated at discrete grid points $\phi^D_{E_{i,j}}$. We would like to find the truncation error, i.e., the difference between the discrete operator acting on the exact solution and the discrete RHS. of equation (-1). For central differencing

$$(\Delta^h \phi^D_E)_{i\Delta x, j\Delta y} = (\Delta\phi_E)_{i\Delta x, j\Delta y} + O(h^2) \tag{71}$$

But

$$\begin{aligned}(\Delta\phi_E)_{i\Delta x, j\Delta y} &= (\rho_E)_{i\Delta x, j\Delta y} \\ &= \rho \\ &= \Delta^h \phi\end{aligned}$$

so that

$$\Delta^h \phi^D_E = \Delta^h \phi + O(h^2)$$

Then, by linearity

$$\Delta^h (\phi^D_E - \phi) = O(h^2) \tag{72}$$

The Laplacian operator is bounded and independent of *h* for Dirichlet boundary conditions. For this case, if we can show that

$$\|\Delta^h \phi\| \geq C \|\phi\| \tag{73}$$

where *C* is positive and independent of the mesh spacing, we can conclude that

$$\phi^D_E - \phi = O(h^2) \tag{74}$$

6.2.4 Time step considerations

Suppose that the temperature $T(\vec{x}, t)$ is governed by the unsteady heat equation:

$$\frac{\partial T}{\partial t} = \kappa \Delta T + f(\vec{x}) \tag{75}$$

where κ is the heat diffusion coefficient. To approach this numerically, we must replace the Laplacian by its discrete representation, Δ^h, and decide at what time to evaluate it, as well as figure out how to properly discretize the partial derivative w.r.t. *t*. A number of choices present themselves.

6.2.4.1 Forward Euler

The discretized equation takes the form:

$$\frac{T^{n+1} - T^n}{\Delta t} = \kappa \Delta^h T^n + f \tag{76}$$

or equivalently

$$T^{n+1} = (I + \kappa(\Delta t)\,\Delta^h)\,T^n + (\Delta t)\,f \tag{77}$$

where *I* is the identity matrix. If the steady-state solution is defined by:

$$\Delta^h T_E^n = -f \tag{78}$$

and the truncation error by:

$$\delta T^n = T^n - T_E^n \tag{79}$$

then, for the solution to converge, i.e., $\|\delta T^{n+1}\| \le \|\delta T^n\|$, it is a sufficient and necessary condition that

$$\frac{\kappa \Delta t}{h^2} \le \frac{1}{2n} \tag{80}$$

where Δt is the time step, *h* is the mesh spacing, and *n* is the number of spatial dimensions in the problem. Catastrophic failure results from violating this condition, but we are reluctant to accept this overly restrictive time step. Recall that the grand plan is to couple this part of the Navier-Stokes equation to the hyperbolic part. We saw that Lax-Wendroff, Fromm and upwinding performed well with $\Delta t = O(\Delta x)$. In addition, we can see trouble from the outset since the errors are unbalanced. Consideration of global error for the hyperbolic case shows that we worked hard to obtain error of $O(h^2, \Delta t^2)$, indicating that *h* is proportional to Δt, whereas for the parabolic case just considered, the error is of $O(h^2, \Delta t)$.

We can get around the stability problem by making the temporal differencing implicit, e.g., using backward Euler.

6.2.4.2 Backward Euler

This scheme is also first order in time, but it is implicit, which leads to unconditional stability. The discretized form of the governing equation is:

$$\frac{T^{n+1} - T^n}{\Delta t} = \kappa \Delta^h T^{n+1} + f \tag{81}$$

or

$$(I - \kappa (\Delta t) \Delta^h) T^{n+1} = T^n + (\Delta t) f \tag{82}$$

where I is the identity matrix.

If e perform a Fourier analysis of the five-point Laplacian discretization similar to the one we madefor hyprbolic problems in Sections 1.2 and 1.5, we can show that the amplitude of the k_xth/k_yth Fourier mode behaves in the following fashion,

$$a^{n+1}_{k_x, k_y} \{ 1 - \Delta t \kappa [-4 + 2\cos\beta_x + 2\cos\beta_y] \} \frac{1}{h^2} = a^n_{k_x, k_y}$$

or

$$a^{n+1}_{k_x, k_y} = \frac{a^n_{k_x, k_y}}{1 + \Delta t \kappa [4 - 2\cos\beta_x - 2\cos\beta_y]} \tag{83}$$

Since the denominator is less than unity, the conclusion is that the amplitudes of Fourier modes decay.

The implicit nature of the algorithm is a mixed blessing. Unlike forward Euler, stability does not restrict the size of the time step although accuracy considerations will effectively limit its size. On the other hand, it means that for almost all practical problems, we now have a linear system to solve. As before, we can expect that careful organization of the system will allow us to have a banded matrix. Therefore, we will probably want to resort to multigrid for an efficient algorithm if we use this discretization. One important consideration remains: we still have a mismatch in the temporal and spatial errors. As is the case for forward Euler, $\Delta t = O(h^2)$. Since we are stuck with an implicit method anyways, with just a little extra work, we can fix the error mismatch by using Crank-Nicholson, as shown in the next section.

6.2.4.3 Crank-Nicholson

For this case, the discretization is:

$$T^{n+1} = T^n + \frac{\Delta t}{2}\kappa\Delta^h (T^{n+1} + T^n) + (\Delta t) f \tag{84}$$

or

$$(I - \frac{\Delta t}{2}\kappa\Delta^h) T^{n+1} = (I + \frac{\Delta t}{2}\kappa\Delta^h) T^n + (\Delta t) f \tag{85}$$

This method costs no more to implement than backward Euler and is similarly unconditionally stable. We have also gained the bonus of a scheme which has second-order accuracy in time. Performing a Fourier analysis gives:

$$a^{n+1}_{k_x, k_y} \{ 1 - \frac{\Delta t}{2}\kappa [-4 + 2\cos\beta_x + 2\cos\beta_y] \frac{1}{h^2} \} =$$

$$= a^n_{k_x, k_y} \{ 1 + \frac{\Delta t}{2}\kappa [-4 + 2\cos\beta_x + 2\cos\beta_y] \frac{1}{h^2} \}$$

To simplify, if we define

$$\mu = \frac{\Delta t}{2h^2}\kappa [-4 + 2\cos\beta_x + 2\cos\beta_y] \tag{86}$$

then we can write

$$a^{n+1}_{k_x, k_y} = a^n_{k_x, k_y} \left[\frac{1-\mu}{1+\mu} \right] \tag{87}$$

We can see that the upper and lower bounds on the growth of the Fourier amplitudes are:

$$-1 \le \frac{1-\mu}{1+\mu} \le 1$$

This scheme is unconditionally stable, although we still must worry about solving a linear system. What's the best approach for solving this? The cost of a direct solver is prohibitive, just as it was in the case of the Poisson equation. We would like to be able to use the multigrid methodology developed in the previous section. Let's denote the temperature at time level *n+1* as $\tilde{T}$. We note that the form of the equation (85) above can be written as a linear operator on $\tilde{T}$ being equal to a right-hand side:

$$L\tilde{T} = \rho \tag{88}$$

if the linear operator is:

$$L = I - \frac{\Delta t}{2}\kappa\Delta^h \tag{89}$$

For example, using the standard five-point Laplacian, we can write $L\tilde{T}$ at any point *(i, j)* as:

$$(L\tilde{T})_{i,j} = (1 + \frac{4\Delta t\kappa}{2h^2})\,\tilde{T}_{i,j} - \frac{\Delta t\kappa}{2h^2}\,[\tilde{T}_{i+1,j} + \tilde{T}_{i-1,j} + \tilde{T}_{i,j+1} + \tilde{T}_{i,j-1}] \tag{90}$$

The RHS of equation (-3) is made up of values of temperature at time level *n* as well as the original source term *f*:

$$\rho = (I + \frac{\Delta t}{2}\kappa\Delta^h)\,T^n + (\Delta t)\,f \tag{91}$$

Is this kosher? What did we need for multigrid to work? A number of considerations present themselves:

(i) **Linearity.** Although not strictly essential, the linearity of the system makes our task simpler. Recall that, in the previous section, we were able to develop a scheme based on the variation between the guess of the solution and the updated value of the solution. For $L\phi = \rho$. we defined the residual to be $R = \rho - L\tilde{\phi}$ where $\tilde{\phi}$ was the initial guess of the solution ϕ; we defined the remainder to be $\delta = \phi - \tilde{\phi}$; and then performed multigrid recursively on the system $L\delta = R$ with homogeneous Dirichlet boundary conditions; and updated the solution by $\phi = \tilde{\phi} + \delta$. The size of the residual gave a direct measure of how close we were in coming to the solution.

(ii) **Damping of short wavelengths by the local relaxation scheme.** This meant that, after a relaxation step, we could accurately represent the solution on a coarser grid. We can transfer the high wavenumber error to low wavenumber if we choose λ to be:

$$\lambda = -\left[1 + \frac{4\Delta t\kappa}{2h^2}\right]^{-1} \tag{92}$$

Although $\lambda=\lambda(h,\Delta t)$ now, rather than being directly related to the time step alone, this choice of λ does allow us to squelch the high wavenumber error.

(iii) **Implicit assumption of ellipticity.** For elliptic problems, if we know the degree to which ρ is smooth, then we also know how smooth ϕ is; and vice versa. Why is this important? We know that if we eliminate the short wavelengths in the error, then we also get rid of the short wavelengths in the RHS. as well. (Technically, the smoothness of $L\phi$ at a point $\vec{x}$ is determined by the smoothness of ρ in the neighborhood of $\vec{x}$.) For transonic flows, there are both hyperbolic regions (in the supersonic portion of the flow), as well as elliptic regions (subsonic flow). Although multigrid is used in transonic flow problems, caution must be exercised in the regions of supersonic flow.

7. The prototype advection equation.

In two dimensions, our model advection equation is

$$\frac{\partial \rho}{\partial t} + (\underset{\sim}{u} \cdot \nabla)\,\rho = 0 \tag{1}$$

If the velocity field is divergence free,

$$\nabla \cdot \vec{u} = 0 \tag{2}$$

then (1) can be rewritten as

$$\frac{\partial s}{\partial t} + u\frac{\partial s}{\partial x} + v\frac{\partial s}{\partial y} = 0 \tag{3}$$

What are our requirements for a high quality method? We want a good first-order method that satisfies the maximum principle and has reasonable stability properties in terms of CFL number. We will then add to this method the minimum number of terms required to make the scheme second order in smooth regions, but must be can be taken out at discontinuities to avoid Gibb's phenomenon.

For the purposes of the following discussion let's assume that the components of the velcoity $\underset{\sim}{u} = (u, v)$ are positive constants, $u > 0$ and $v > 0$. As usual, we assume that the mesh spacing is uniform in both dimensions. Now think of ρ_{ij}^n as the integrated average of the advected scalar, e.g., mass, in cell *(i, j)* at time *n*,

$$\rho_{ij}^n \approx \frac{1}{\Delta x \Delta y}\int \rho\,(x, y, t^n)\,dxdy \tag{4}$$

What happens as we move forward in time? The mass in cell *(i, j)* at time *n+1* results from advection directly from the cell itself at time *n*, as well as contributions from its nearest neighbors. Recalling that *u* and *v* are positive, we can express the mass in cell *(i, j)* at the next time level using piecewise constant interpolation:

$$\rho_{ij}^{n+1} = \frac{1}{\Delta x \Delta y}\left[A_1\rho_{ij}^n + A_2\rho_{i-1,j}^n + A_3\rho_{i,j-1}^n + A_4\rho_{i-1,j-1}^n\right] \tag{5}$$

The areas A_1 to A_4 are all positive and at most equivalent to the cell area $\Delta x \Delta y$ (for the case of purely one-dimensional flow in the *x* direction). Since we are using the area-weighted average of the values at the old time, we will introduce no new maxima or minima by using this evolution equation. Figure +1 below depicts this process graphically. In this figure, the large region represents the domain at time level *n+1*, with cell *(i, j)* being at the center (cross-hatched region). There are four contributions coming from the cell itself

and the adjacent cells at time *n*, shown by the shaded region. All of the contributions are advected at the speed $\underset{\sim}{u}\Delta t$.

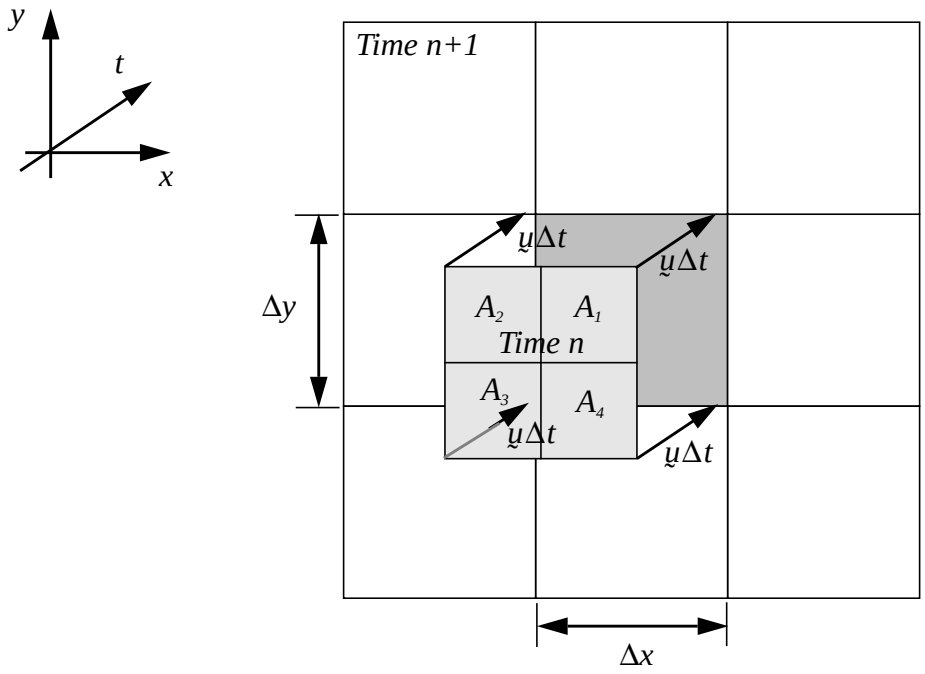


Figure 45Areas through which mass is transported to cell *(i, j)* at time *n+1*.

Next, we'd like to figure out just what the flux of mass through the cell edges are, e.g., the right cell edge $i + \frac{1}{2}$. We note that the velocity components will indicate how much flux is moving through in either the *x* or *y* direction. Let's make up the following notation to denote the areas through which relevant flux traverses:

$$B_o = (u\Delta t)\,\Delta y \tag{6}$$

$$B_B = \frac{1}{2}(u\Delta t)\ (v\Delta t) \tag{7}$$

$$B_T = \frac{1}{2}(u\Delta t)\ (v\Delta t) \tag{8}$$

Graphically,

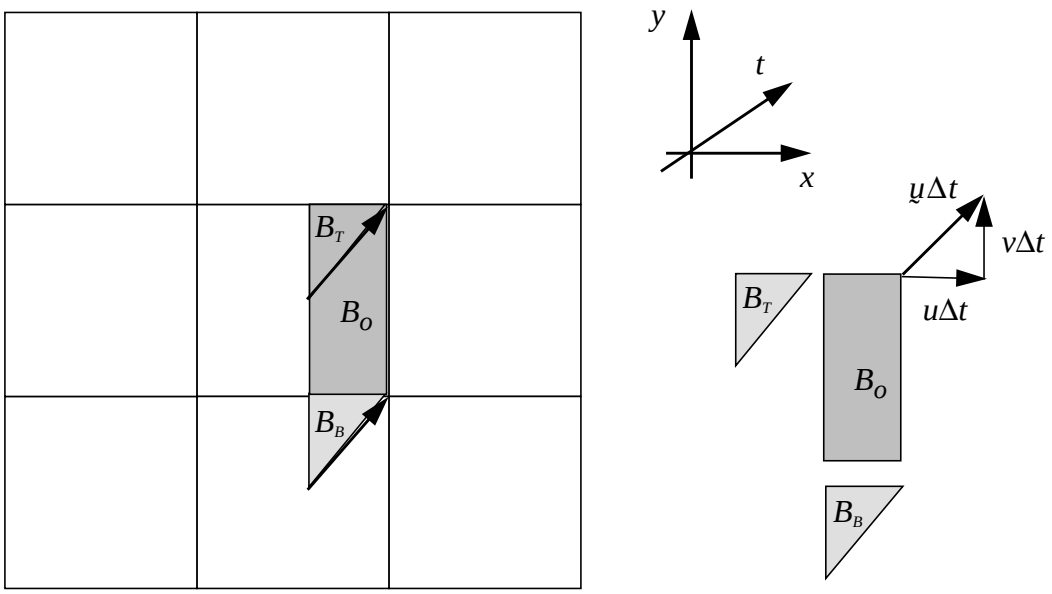


Figure 46. Flux which enters or leaves through right cell edge between time *n* and *n+1*

We can then express the mass flux out of edge $i+\frac{1}{2}$, $F_{i+1/2}$, as:

$$F_{i+\frac{1}{2}} = B_o\rho^n_{ij} + B_B\rho^n_{i,j-1} - B_T\rho^n_{ij}$$

$$= (u\Delta t)\,\Delta y\rho^n_{ij} + \frac{1}{2}(u\Delta t)\,(v\Delta t)\,(\rho^n_{i,j-1} - \rho^n_{ij})$$

and introducing the definition for $\rho_{i+1/2,j}$:

$$= (u\Delta t)\,\Delta y\rho_{i+1/2,j}$$

so that

$$\rho_{i+1/2,j} = \rho^n_{ij} + \frac{v\Delta t}{2\Delta y}(\rho^n_{i,j-1} - \rho^n_{ij}) \tag{9}$$

If we perform a similar analysis for the upper edge, we obtain the expression,

$$\rho_{i,j+1/2} = \rho^n_{ij} + \frac{u\Delta t}{2\Delta x}(\rho^n_{i-1,j} - \rho^n_{ij}) \tag{10}$$

ultimately arriving at the evolution equation,

$$\rho^{n+1}_{ij} = \rho^n_{ij} + \frac{u\Delta t}{2\Delta x}(\rho^n_{i-1/2,j} - \rho^n_{i+1/2,j}) + \frac{v\Delta t}{2\Delta y}(\rho^n_{i,j-1/2} - \rho^n_{i,j+1/2}) \tag{11}$$

This evolution equation is only first-order accurate. Is there something we can do about that? Equation (11) above is an approximation to

$$\rho^{n+1/2}_{i,j+1/2} = \rho^n_{ij} + \frac{\Delta x}{2}\frac{\partial \rho}{\partial x} + \frac{\Delta t}{2}\frac{\partial \rho}{\partial t} \tag{12}$$

Using in (12)

$$\begin{aligned}\rho^{n+1/2}_{i,j+1/2} &= \rho^n_{ij} + \frac{\Delta x}{2}\frac{\partial \rho}{\partial x} - \frac{u\Delta t}{2}\frac{\partial \rho}{\partial x} - \frac{v\Delta t}{2}\frac{\partial \rho}{\partial y} \\ &= \rho^n_{ij} + \left(\frac{\Delta x}{2} - \frac{u\Delta t}{2}\right)\frac{\partial \rho}{\partial x} - \frac{v\Delta t}{2}\frac{\partial \rho}{\partial y}\end{aligned}$$

As before, we'd like to apply a limiter to the term multiplied by $\partial\rho/\partial x$ and in addition, use upwind differencing on the term multiplied by $\partial\rho/\partial y$. Let's give it the old van Leer heave ho:

$$\rho^{n+1/2}_{i,j+1/2} = \rho^n_{ij} + \frac{1}{2}\left(1 - \frac{u\Delta t}{2\Delta x}\right)\Delta_x \rho_{ij} - \frac{\Delta t}{2\Delta y} v\left(\rho^n_{ij} - \rho^n_{i,j-1}\right) \tag{13}$$

where

$$\begin{aligned}\Delta_x \rho_{ij} = sign\left(\rho_{i+1,j} - \rho_{i-1,j}\right) \cdot \\ min\left(2\left|\rho_{i,j} - \rho_{i-1,j}\right|, \frac{1}{2}\left|\rho_{i+1,j} - \rho_{i-1,j}\right|, 2\left|\rho_{i,j} - \rho_{i+1,j}\right|\right)\end{aligned} \tag{14}$$

if $(\rho_{i-1,j} - \rho_{ij}) \cdot (\rho_{ij} - \rho_{i-1,j}) > 0$; or is set to zero otherwise.

This scheme is not monotonicity-preserving. While this is not a major drawback from the standpoint of linear systems, it is a problem for nonlinear advection. Two classes of fixes have been proposed: augment the van Leer limiter using B limiters (J. Saltzman); or do the interpolation (advection) in a more appropriate fashion (van Leer, 1984; Bell, Dawson and Shubin, 1989).

8. The Navier-Stokes equations.

Recall that our goal was to solve the incompressible momentum equation. For our purposes, we can think of momentum as a scalar which is being advected by an incompressible velocity field. The velocity field is divergence-free so that:

$$\nabla \cdot \underset{\sim}{u} = 0 \tag{1}$$

At the moment, we can actually handle the following model for the Navier-Stokes equation:

$$\frac{\partial s}{\partial t} + (\underset{\sim}{u} \cdot \nabla) s = \varepsilon \Delta s \tag{2}$$

where $\underset{\sim}{u} = \underset{\sim}{u}(\underset{\sim}{x})$ and ε is some diffusion coefficient. We know how to handle the advection and diffusion pieces separately; what remains is to explore hybrid approaches which allow us to combine these skills to solve equation (2) above. The basic difference approximation is:

$$s_{ij}^{n+1} = s_{ij}^{n} - \Delta t \, (\underset{\sim}{u} \cdot \nabla s)_{ij}^{n+1/2} + \frac{\Delta t}{2} \varepsilon \Delta^h \left(s^n + s^{n+1}\right)_{ij}$$

or, rewriting:

$$(I - \frac{\Delta t}{2} \varepsilon \Delta^h) \, s_{ij}^{n+1} = s_{ij}^{n} - \Delta t \, (\underset{\sim}{u} \cdot \nabla s)_{ij}^{n+1/2} + \frac{\Delta t}{2} \varepsilon \Delta^h s_{ij}^{n} \tag{3}$$

What are we doing here? We want to use Crank-Nicholson temporal differencing on the diffusion part of the equation since this scheme has desirable stability properties and can avoid gross mismatch of the spatial and temporal errors. We must require the advection scheme to have a compatible time step. We also want the advection part to be explicit to avoid its dependency on the (n+1)st time level. Furthermore, if the diffusion coefficient is zero, we want the hybrid scheme to reduce to second-order upwinding. Using the procedures which we by now have in place, we can do this. We have obtained a well-behaved system.

8.1 The treatment of the nonlinear advection terms

In computing the advection terms, what we will actually compute is $\nabla \cdot (\underset{\sim}{u} s)^{n+1/2}$. This is justified because the velocity field is divergence-free. In the discrete formulation, we will assume that the scalar s lives at the cell centers. On the other hand, the velocity $\underset{\sim}{u}$ has

components *(u, v)* which escort the scalar around the flowfield and will be defined at the edges of the cell as shown in Figure 46:

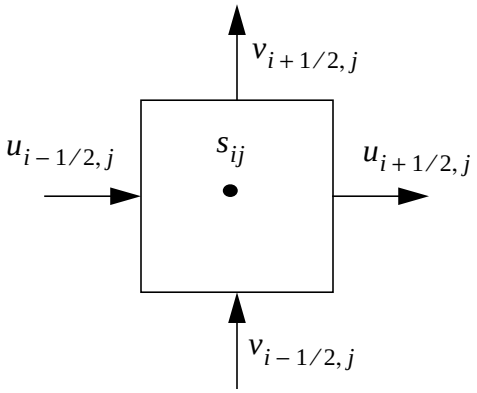


Figure 47. A representative cell

The divergence-free condition gives the following first-order approximation:

$$\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} = 0 \tag{4}$$

If the function $\underset{\sim}{u}(\underset{\sim}{x})$ is known analytically, then we can think of the velocity at the right edge as an average integrated value across the cell edge, i.e.,

$$u_{i+1/2,j} = \frac{1}{\Delta y} \int_{(j-\frac{1}{2})\Delta y}^{(j+\frac{1}{2})\Delta y} u\left((i+1/2)\Delta x, y\right) dy \tag{5}$$

This satisfies the divergence-free condition by the divergence theorem. Note that the RHS of equation (4) is one of four terms of the RHS of:

$$\int_V (\nabla \cdot \underset{\sim}{u})\, dV = \oint_S (\underset{\sim}{u} \cdot \hat{n})\, dA \tag{6}$$

We can now write a finite-difference approximation for $\nabla \cdot (\underset{\sim}{u}s)^{n+1/2}$. Recalling that we have specified that $\underset{\sim}{u} = \underset{\sim}{u}(\underset{\sim}{x})$ alone:

$$\nabla \cdot (\underset{\sim}{u}s) = \frac{1}{\Delta x}\left(u_{i+1/2,j} s^{n+1/2}_{i+1/2,j} - u_{i-1/2,j} s^{n+1/2}_{i-1/2,j}\right) +$$

$$\frac{1}{\Delta y}\left(v_{i,j+1/2} s^{n+1/2}_{i,j+1/2} - v_{i,j-1/2} s^{n+1/2}_{i,j-1/2}\right) \tag{7}$$

Now, recall that the discretized version of *s* exists at cell centers. To extrapolate to the cell edges, perform a Taylor expansion:

$$s^{n+1/2}_{i+1/2,j} \approx s^n_{ij} + \frac{\Delta x}{2}\frac{\partial s}{\partial x} + \frac{\Delta t}{2}\frac{\partial s}{\partial t} \tag{8}$$

We were able to write the PDE as

$$\frac{\partial s}{\partial t} = \nabla \cdot (\underline{u} s) + \varepsilon \Delta s$$

or, for our case of two-dimensional Cartesian coordinates:

$$\frac{\partial s}{\partial t} = \frac{\partial (us)}{\partial x} + \frac{\partial (vs)}{\partial y} + \varepsilon \Delta s \tag{9}$$

Putting equation (9) into equation (8), we obtain:

$$s^{n+1/2}_{i+1/2,j} = s^n_{ij} + \frac{\Delta x}{2}\frac{\partial s}{\partial x} - \frac{u\Delta t}{2}\frac{\partial s}{\partial x} - \frac{\Delta t}{2} s \frac{\partial u}{\partial x} - \frac{\partial (vs)}{\partial y} - \varepsilon \frac{\Delta t}{2}\Delta s$$

or

$$s^{n+1/2}_{i+1/2,j} = s^n_{ij} + \left(1 - \frac{u\Delta t}{\Delta x}\right)\frac{\Delta x}{2}\frac{\partial s}{\partial x} - \frac{\Delta t}{2} s \frac{\partial u}{\partial x} - \frac{\Delta t}{2}\frac{\partial (vs)}{\partial y} - \varepsilon \frac{\Delta t}{2}\Delta s \tag{10}$$

The fully discretized form of this equation is:

$$s^{n+1/2}_{i+1/2,j} = s^n_{ij} + \frac{1}{2}\left(1 - \frac{u_{i+1/2,j}\Delta t}{\Delta x}\right)(\Delta_x s)_{ij} - \frac{\Delta t}{2\Delta x} s^n_{ij}\left(u_{i+1/2,j} - u_{i-1/2,j}\right) -$$
$$\frac{\Delta t}{2\Delta y}\left(v_{i,j+1/2} s^{up}_{i,j+1/2} - v_{i,j-1/2} s^{up}_{i,j-1/2}\right) - \varepsilon \frac{\Delta t}{2}\left(\Delta^h s^n\right)_{ij} \tag{11}$$

where the upwind value of s was denoted by s^{up}, i.e.

$$s^{up}_{i,j+1/2} = \begin{cases} s^n_{ij} \ldots if \ldots (v_{i,j+1/2} > 0) \\ s^n_{i,j+1} \ldots if \ldots (v_{i,j+1/2} < 0) \end{cases} \tag{12}$$

There are several comments to be made regarding this approach.

(a) Should we be disturbed by putting in an explicit estimate of $(\varepsilon \Delta t \Delta s)/2$? The answer is no, since the Crank-Nicholson scheme gives the requisite stability, as is easily verified by computing the symbol of the scheme.

(b) What happens if $s_{ij} = s_0$, a constant, in the neighborhood of s_{ij}? In this case, in the computation of $s^{n+1/2}_{i+1/2,j}$, the term yielding $\partial v/\partial y \neq 0$, which looks problematic. However, the terms

$$\frac{\Delta t}{2}\frac{\partial (vs)}{\partial y} + \frac{\Delta t}{2} s \frac{\partial u}{\partial x}$$

cancel each other out because of the divergence-free condition. So, in this case is not a problem. This property of the finite difference scheme is called *freestream-preserving*.

(c) What about stability? The CFL condition for the advective part is

$$\left| \frac{\Delta t}{\Delta x} u_{i+1/2,j} \right|, \left| \frac{\Delta t}{\Delta x} v_{i,j+1/2} \right| \leq 1 \tag{13}$$

and recall that $\Delta t = O(\Delta x, \Delta y)$. To get accuracy out of the advection part, we want to limit the CFL numbers between braces above to the range of 0.5 to 1.0. Crank-Nicholson allows us to take a time step that is limited only by the advective CFL condition.

8.2 The incompressible Navier-Stokes equations

We would like to solve for the two-dimensional velocity field $\underline{u}$ on the bounded domain Ω with fixed, impermeable solid-wall boundaries. For now. The governing equations are continuity and momentum, which can be written:

$$\nabla \cdot \underline{u} = 0 \tag{14}$$

$$\frac{\partial \underline{u}}{\partial t} + (\underline{u} \cdot \nabla)\,\underline{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \underline{u} \tag{15}$$

The boundary conditions are either no-slip:

$$\underline{u} = 0 \ldots on \ldots \partial D \tag{16}$$

or, for inviscid walls:

$$\underline{u} \cdot \hat{n} = 0 \ldots on \ldots \partial D \tag{17}$$

To simplify notation, let's take our units so that ρ = constant. We realize that p is not a separately specified dependent variable; if we know $\underline{u}(\underline{x}, t)$, we can compute p: formally, the divergence of the momentum equation yields a Poisson equation for the pressure.Taking this formal observation to seriously as a perscription for solving the problem can lead to some serious conceptual and technical difficulties; to avoid those difficulties, we take a slightly indirect approach.

8.2.1 The divergence, gradient and inner product

We begin with some definitions. For any two-dimensional vector field, the divergence of $\underline{u} = (u, v)$ is given by:

$$div\,(\underline{u}) \;=\; \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \tag{18}$$

For any scalar field in two dimensions, the gradient is given by:

$$grad\,(\phi) \;=\; (\phi_x, \phi_y) \;=\; (\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}) \tag{19}$$

We observe, as a consequence of the divergence theorem,

$$\begin{aligned}\int_D \phi\, div\,(\underline{u})\, dV &= \int_D div\,(\phi \underline{u})\, dV - \int_D grad\,(\phi) \cdot \underline{u} dV \\ &= \oint_{\partial D} \phi\,(\underline{u} \cdot \hat{n})\, dA - \int_D grad\,(\phi) \cdot \underline{u} dV \end{aligned} \tag{20}$$

If the velocity field satisfies $\underline{u} \cdot \hat{n} \;=\; 0$ on ∂D, then we find that,

$$\int_D \phi\, div\,(\underline{u})\, dV \;=\; -\int_D grad\,(\phi) \cdot \underline{u} dV \tag{21}$$

The inner product of two scalars is defined by,

$$\langle \phi, \psi \rangle_S \;=\; \int_D \phi\,(\underline{x})\, \psi\,(\underline{x})\, dV \tag{22}$$

while the inner product oftwo vectors is defined by,

$$\langle \underline{v}, \underline{w} \rangle_V \;=\; \int_D \underline{v}\,(\underline{x}) \cdot \underline{w}\,(\underline{x})\, dV \tag{23}$$

Thus, when the vector field satisfies $\underline{u} \cdot \hat{n} \;=\; 0$ on ∂D, we obtain,

$$\langle div\,(\underline{u})\,, \psi \rangle_S \;=\; -\langle \underline{u}, grad\,(\phi) \rangle_V \tag{24}$$

8.2.2 The Hodge decomposition

Given the vector field, $\underline{w}$, on a simply connected domain, *D,* which can be described by:

$$\underline{w} \;=\; \underline{w}\,(\underline{x}) \ldots in \ldots D \tag{25}$$

$$\underline{w} \cdot \hat{n} \;=\; 0 \ldots on \ldots \partial D \tag{26}$$

there is a unique decomposition of $\underline{w}$ into a divergence-free part, $\underline{w}_d$, and the gradient of some scalar function, ϕ:

$$\underline{w} \;=\; \underline{w}_d + grad\,(\phi) \tag{27}$$

To get a handle on what this $\underline{w}_d$ is, take the divergence of equation (27). We realize that by definition $div\,(\underline{w}_d) \;=\; 0$. This means that in order to solve

$$div(\underset{\sim}{w}) = div(grad(\phi)) \tag{28}$$

we can find $\underset{\sim}{w}_d$ by:

$$\underset{\sim}{w}_d = \underset{\sim}{w} - grad(\phi) \tag{29}$$

If we take the divergence of equation (29) above, we see that the divergence of $\underset{\sim}{w}_d$ vanishes:

$$div(\underset{\sim}{w}_d) = div(\underset{\sim}{w}) - div(grad(\phi))$$

The terms on the RHS cancel each other out, as shown in equation (29) above, leaving a big fat goose egg.

We can also see that the components of $\underset{\sim}{w}$ are orthogonal, i.e.:

$$\langle \underset{\sim}{w}_d, grad(\phi) \rangle_V = \int_D \underset{\sim}{w}_d \cdot grad(\phi)\, dV$$

Now, using the no-flow boundary condition $\underset{\sim}{u} \cdot \hat{n} = 0$

$$\langle \underset{\sim}{w}_d, grad(\phi) \rangle_V = -\int_D div \underset{\sim}{w}_d \phi dV$$
$$= 0$$

The orthogonality is a consequence of the fact that divergence and gradient operators are adjoints.

8.2.3 The projection operator

One of the difficulties of working with the incompressible Navier-Stokes equation is that we must both satisfy the divergence-free condition imposed by continuity and find the pressure gradient. We can in a sense blend these constraints by using a projection operator which forces the velocity field to be divergence-free; whatever remains from the right-hand side of the momentum equation is absorbed intothe pressure gradient. Another advantage of the projection formulation is that it leads to pure initial value formulation for $\underset{\sim}{u}$.

We can define a projection operator, *P*, as something which takes a vector field and extracts its divergence-free part. So, we can hit the vector field $\underset{\sim}{w}$ with the projection operator to obtain $\underset{\sim}{w}_d$. If $\underset{\sim}{w}$ is sufficiently smooth, from equations (27) and (29):

$$\underset{\sim}{w}_d = P\underset{\sim}{w} = (I - grad(divgrad)^{-1} div)\, \underset{\sim}{w} \tag{30}$$

P can be extended on $L^2(D)$. It is a way of extracting the divergence-free part, even if we don't have enough derivatives for the individual operations in (27) and (29) to be well-defined.

Some properties of the projection operator are that it is symmetric;

$$\langle \underset{\sim}{w}, P\underset{\sim}{w} \rangle = \langle P\underset{\sim}{w}, \underset{\sim}{w} \rangle \tag{31}$$

$$P = P^T \tag{32}$$

and bounded:

$$\| P \| = 1 \tag{33}$$

$$\| P\underset{\sim}{w} \| \leq \| \underset{\sim}{w} \| \tag{34}$$

Also, since the decomposition is unique, P is idempotent.

$$P^2 = P \tag{35}$$

What do projection operators look like in space? If we set the domain to be $\Re^3$ and $\underset{\sim}{w}$ is of compact support, then we can write $\underset{\sim}{w}_d$ as:

$$\underset{\sim}{w}_d = \int_{\Re^3} K(\underset{\sim}{x} - \underset{\sim}{x}') \, \underset{\sim}{w}(\underset{\sim}{x}) \, d\underset{\sim}{x} \tag{36}$$

where

$$K(\underset{\sim}{x}) = \delta(\underset{\sim}{x}) I - \frac{1}{4\pi} \left(\frac{I}{|\underset{\sim}{x}|^3} - \frac{3\underset{\sim}{x}\underset{\sim}{x}^T}{|\underset{\sim}{x}|^3} \right) \tag{37}$$

For bounded domains, we want a projection formulation of the Navier-Stokes equation,

$$\frac{\partial \underset{\sim}{u}}{\partial t} + (\underset{\sim}{u} \cdot \nabla) \underset{\sim}{u} = -\nabla p + \nu \Delta \underset{\sim}{u} \tag{38}$$

subject to the no-slip boundary condition,

$$\underset{\sim}{u} = 0 \ldots on \ldots \partial D \tag{39}$$

or, alternatively, if $\nu = 0$, we will impose the no-flow condition,

$$\underset{\sim}{u} \cdot \hat{n} = 0 \ldots on \ldots \partial D \tag{40}$$

First, apply the projection operator to equation (30),

$$P\left(\frac{\partial \underset{\sim}{u}}{\partial t}\right) + P\left((\underset{\sim}{u} \cdot \nabla)\underset{\sim}{u}\right) = P(-\nabla p) + P(\nu \Delta \underset{\sim}{u})$$

The projection of a pure gradient is zero, so that $P(-\nabla p) = 0$. Rewriting,

$$\frac{\partial \underset{\sim}{u}}{\partial t} + P\left((\underset{\sim}{u} \cdot \nabla)\underset{\sim}{u} - \nu \Delta \underset{\sim}{u}\right) = 0 \tag{41}$$

with continuity automatically satisfied ($\nabla \cdot \underset{\sim}{u} = 0$). If we compare equation (41) to the original equation (36), we arrive at an expression for the pressure gradient,

$$\nabla p = (I - P)\,(-(\underline{u}\cdot\nabla)\,\underline{u} + \nu\Delta\underline{u}) \tag{42}$$

We see from equation (38) above that the pressure gradient absorbs any divergence that results from the piece of $\frac{\partial u}{\partial t}$ which is not divergence-free.

We might also be tempted eliminate the viscous terms from (42) by using

$$P\nu\Delta\underline{u} = \nu\Delta P\underline{u}$$

Since we would expect that the divergence and gradient operators commute with Δ. This is valid in infinite space, but is not correct in the presence of no-slip boundaries. In the latter case, the effect of the viscous terms, combined with the boundary conditions, is to cause the velocity field to move off of the subspace divergence-free velocity fields.

8.3 The projection method

Let's examine the discretization of the projection formulation in two dimensions. Assume that we have discretizations for the advective and diffusive parts which are stable, satisfy the boundary conditions, and are denoted by the following:

$$A_{ij} \approx (\underline{u}\cdot\nabla)\,\underline{u} \tag{43}$$

$$(\Delta^h\underline{u})_{ij} \approx \Delta\underline{u} \tag{44}$$

The discrete evolution equation can be written:

$$u_{ij}^{n+1} = u_{ij}^n + \Delta t\,(P\,(-A + \nu\Delta^h\underline{u}))_{ij} \tag{45}$$

where P is a discretization of a continuous projection operator. Typically, we will use a fractional-step method:

$$\underline{u}_{ij}^{\,*} = \underline{u}^n - \Delta t A_{ij} + \Delta t\nu\,(\Delta^h\underline{u})_{ij} \tag{46}$$

$$\underline{u}_{ij}^{n+1} = P\underline{u}_{ij}^{\,*}$$

But what exactly is the projection operator? Let's assume we're operating in a discrete domain $\Omega = [1\ldots N]\,x\,[1\ldots N]$ with either solid walls, for which:

$$\underline{u}\cdot\hat{n} = 0\ldots on\ldots\partial\Omega \tag{47}$$

or doubly periodic boundaries. Our strategy will be to:

(a) Define the discrete divergence D using finite differences; and

(b) Define the discrete gradient $G = -D^T$ with respect to some pair of inner products

We define the discrete inner product for scalar fields by,

$$\langle \phi, \psi \rangle_S = h^2 \sum_{ij} \phi_{ij} \cdot \psi_{ij} \tag{48}$$

and for vector fields by,

$$\langle \underline{v}, \underline{w} \rangle_V = h^2 \sum_{ij} \underline{v}_{ij} \cdot \underline{w}_{ij} \tag{49}$$

To begin, we will use centered difference approximations to define the discrete analogs of the divergence, gradient and projection operators. Ultimately, though, these will prove unsatisfactory for a variety of reasons, and we will replace them with more appropriate discretizations.

8.3.1 The discrete divergence operator

If the components of $\underline{u} = (u, v)$, for the cell *(i,j)* which is sufficiently removed from the boundaries, we can define the discrete divergence by:

$$(D\underline{u})_{ij} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{h} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{h} \tag{50}$$

where the cell-edge velocities are assumed to be the average of the two adjacent cell centers:

$$u_{i+1/2,j} = \frac{u_{ij} + u_{i+1,j}}{2} \tag{51}$$

and

$$v_{i,j+1/2} = \frac{v_{ij} + v_{i,j+1}}{2} \tag{52}$$

In the interior, this discretization reduces to centered differences:

$$(D\underline{u})_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + \frac{v_{i,j+1} - v_{i,j-1}}{2h} \tag{53}$$

For the boundary cells with solid walls, e.g.,

$$(D\underline{u})_{1,j} = \frac{u_{2,j} + u_{1,j}}{2h} + \frac{v_{1,j+1/2} - v_{1,j-1/2}}{h} \tag{54}$$

or for periodic boundary conditions:

$$(D\underline{u})_{1,j} = \frac{u_{2,j} - u_{N,j}}{2h} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{h} \tag{55}$$

8.3.2 The discrete gradient operator

Next, we want to find the discrete gradient. For the continuous case, we know that the boundary terms vanish ($\underline{u} \cdot \hat{n} = 0$) and that the velocity field is divergence-free ($\nabla \cdot \underline{u} = 0$). The relationship between the vector fields, the adjoint condition and the scalar inner product will define the discrete gradient.

Recall that in the continuous case,

$$\int_D \phi div(\underline{u})\, dV = -\int_D grad(\phi) \cdot \underline{u} dV \tag{56}$$

We would also like this to hold for the discrete case so that,

$$\langle D\underline{u}, \phi \rangle_S = -\langle \underline{u}, G\phi \rangle_V \tag{57}$$

or

$$G = -D^T \tag{58}$$

How do we compute the discrete gradient in the *x*-direction, $(G\phi)^x_{ij}$? We will need to use the adjoint relation. Let $\hat{e}^x_{kl}$ be a vector field defined on Ω by

$$\hat{e}^x_{kl;ij} = \begin{cases} (1,0) \ldots if \ldots (i,j) = (k,l) \\ (0,0) \ldots otherwise \end{cases} \tag{59}$$

The function of $\hat{e}^x_{kl}$ is to pick out the *x* component of the gradient at *(k, l)*. From the adjoint relation, we know that:

$$\begin{aligned} \langle D\hat{e}^x_{kl}, \phi \rangle_S &= -\langle \hat{e}^x_{kl}, G\phi \rangle_V \\ &= -h^2 \sum_{ij} \hat{e}^x_{kl;ij} \cdot (G\phi)_{ij} \\ &= -h^2 (1,0) (G\phi)_{kl} \\ &= -h^2 (G\phi)^x_{kl} \end{aligned}$$

so that we can find the gradient by:

$$(G\phi)^x_{ij} = -\frac{1}{h^2} \langle D\hat{e}^x_{ij}, \phi \rangle_S \tag{60}$$

For the *y* direction, we find that:

$$(G\phi)^y_{kl} = -\frac{1}{h^2} \langle D\hat{e}^y_{kl}, \phi \rangle_S \tag{61}$$

where

$$\hat{e}^{y}_{kl;ij} = \begin{cases} (0,1) \ldots if \ldots (i,j) = (k,l) \\ (0,0) \ldots otherwise \end{cases} \tag{62}$$

8.3.3 The discrete projection operator

In order to define the discrete projection operator, apply the discrete divergence to the definition of the Hodge decomposition of the vector field $\underset{\sim}{w}$:

$$D\underset{\sim}{w} = D(\underset{\sim}{w}_d + G\phi)$$

We know that $D\underset{\sim}{w}_d = 0$ since this component is divergence free. Then we can say that,

$$D\underset{\sim}{w} = DG\phi \tag{63}$$

If we can solve equation (63) for $G\phi$, then we can solve for $\underset{\sim}{w}_d$ by,

$$\underset{\sim}{w}_d = \underset{\sim}{w} - G\phi \tag{64}$$

We note that,

$$\begin{aligned} \langle \underset{\sim}{w}_d, G\phi \rangle_V &= -\langle D\underset{\sim}{w}_d, \phi \rangle_S \\ &= 0 \end{aligned}$$

since $D\underset{\sim}{w}_d$ is by definition zero. This indicates that $\underset{\sim}{w}_d$ is orthogonal to $G\phi$.

The definition of the discrete projection operator is,

$$P = (I - G(DG)^{-1}D) \tag{65}$$

Just as for the continuous projection operator, the discrete projection is symmetric, bounded, and idempotent.

. Can we be sure that equation (63) is always solvable? We must examine this question through consideration of solvability conditions and the Fredholm alternative. The Fredholm alternative tells us that $L\phi = \rho$ is solvable if the statement that $L^T\pi = 0$ implies that π and ρ are orthogonal, i.e., that ρ is normal to the kernel of L. For our particular case, we have shown that $D = -G^T$. Let's modify the notation of equation (63) to emphasize this result. If we denote the discretized divergence matrix as the rectangular matrix A, then the discretized gradient matrix is $-A^T$. Then, equation (63) has the form:

$$-AA^T\Phi = Af \tag{66}$$

We also know that in our case, A is symmetric so that $(-AA^T)^T = -AA^T$. The Fredholm alternative says that if $AA^T\Phi = 0$, then $\langle \Phi, Af \rangle = 0$. Let's use as a starting point our statement that $AA^T\Phi = 0$:

$$\begin{aligned} 0 &= \langle AA^T\Phi, \Phi\rangle \\ &= \langle A^T\Phi, A^T\Phi\rangle \\ &= \left\| A^T\Phi \right\|^2 \end{aligned}$$

This means that $A^T\Phi$ is itself zero. (This is equivalent to saying that if $DG\phi = 0$, then $G\phi = 0$.) So, we have shown that

$$\begin{aligned} \langle \Phi, Af\rangle &= \langle A^T\Phi, f\rangle \\ &= \langle 0, f\rangle \\ &= 0 \end{aligned}$$

Thus, $D\underset{\sim}{w} = DG\phi$ is always solvable because of the adjoint condition $D = -G^T$. In particular, iterative methods have a chance of working, even though the operator is not invertible.

8.3.4 Chorin's projection method.

In Chorin's original version of the projection method, we are given $\underset{\sim}{u}$ as initial data. We will solve the discretized equation

$$D\underset{\sim}{u} = DG\phi \tag{67}$$

for ϕ. We know that this equation is solvable because D and G are adjoints. Next, we calculate the divergence-free velocity by:

$$\underset{\sim}{u}_d = \underset{\sim}{u} - G\phi \tag{68}$$

The evolution equation is:

$$u_{ij}^{*} = \underset{\sim}{u}^n - \Delta t A_{ij} + \Delta t \nu \left(\Delta^h \underset{\sim}{u}\right)_{ij} \tag{69}$$

where $A = A(\underset{\sim}{u})$ and is the discrete analog of $(\underset{\sim}{u} \cdot \nabla)\,\underset{\sim}{u}$. Then

$$\underset{\sim}{u}^{n+1} = P\underset{\sim}{u}^{*} \tag{70}$$

where P is the discrete projection operator,

Between two time levels, we can define the gradient of pressure as,

$$(Gp)^{n,\,n+1} \approx \frac{\underset{\sim}{u}^{*} - \underset{\sim}{u}^{n+1}}{\Delta t} = (I - P)\left(\frac{\underset{\sim}{u}^{*} - \underset{\sim}{u}^{n}}{\Delta t}\right) \tag{71}$$

This is an intrinsically first-order method.

8.3.5 A second-order projection method

We would like to increase the accuracy of the projection to $O(\Delta t^2)$. By analogy with the advection/diffusion equation, we can again use some sort of hybrid scheme which handles different pieces of the discretized equation in different ways. We had some success in using Crank-Nicholson previously, so let's examine what good this method would do here. For this case, the evolution equation is:

$$\underline{u}^{n+1} = \underline{u}^n + \Delta t P\left(-A(\underline{u})^{n+1/2} + \frac{\nu}{2}\Delta^h(\underline{u}^n + \underline{u}^{n+1})\right) \tag{72}$$

The first term in brackets represents the discretization of the $(\underline{u}\cdot\nabla)\underline{u}$ in the Navier-Stokes equation. Our experience obtained from studying the scalar advection - diffusion equation indicates that we should use an explicit hyperbolic predictor/corrector method on this piece of the equation. The second term in brackets is the diffusive part. We note immediately that there is a spot of trouble here: grouping together terms which are dependent on the new time level leads to having to solve a problem of the form

$$(I - \frac{\nu\Delta t}{2}P\Delta^h)\,\underline{u}^{n+1} = RHS.$$

The presence of the projection operator here is a serious problem. We could solve for this iteratively, but with every iteration, we would have to do a lot of work in solving linear systems ($D\underline{u} = DG\phi$). In addition, applying the projection operator to Δ^h is rather ugly and can lead to conditioning problems. The fix is to assume at the beginning of the time step that we know both $\underline{u}^n$ and a time-centered value: $(Gp)^{n-1/2}$. The basic idea is to first compute in step 1:

Step 1

$$\underline{u}^* = \underline{u}^n + \Delta t\left[-A(\underline{u})^{n+1/2} + \frac{\nu}{2}\Delta^h(\underline{u}^n + \underline{u}^*) - (Gp)^{n-1/2}\right] \tag{73}$$

followed by step 2:

Step 2

$$\underline{u}^{n+1} = P\underline{u}^* \tag{74}$$

$$(Gp)^{n+1/2} = (Gp)^{n-1/2} + \frac{\underline{u}^* - \underline{u}^{n+1}}{\Delta t} \tag{75}$$

$(Gp)^{n+1/2}$ can also be written as:

$$(Gp)^{n+1/2} = -(I-P)\left[-A(\underline{u})^{n+1/2} + \frac{\nu}{2}\Delta^h(\underline{u}^n + \underline{u}^*)\right] \tag{76}$$

To compute $\underset{\sim}{u}^*$ in step 1, we assumed that we knew $(Gp)^{n-1/2}$, and also that $A(\underset{\sim}{u})^{n+1/2}$ was explicit. These two assumptions are easy to satisfy. Also, to solve for the incremental value $\underset{\sim}{u}^*$, we had to solve a system of the form:

$$(I - \frac{\nu \Delta t}{2} \Delta^h)\, \underset{\sim}{u}^* = RHS \tag{77}$$

This is not a problem either, and in fact converges faster than the Poisson equation .

What about step 2? For the projection, we needed to solve for the discrete divergence, *D*, which is given by:

$$(D\underset{\sim}{u})_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + \frac{v_{i,j+1} - v_{i,j-1}}{2h} \tag{78}$$

and the discrete gradient $G = -D^T$. Mathematically, $DG\phi$ looks like:

$$DG\phi = \frac{1}{4h^2} [\phi_{i+2,j} + \phi_{i,j+2} + \phi_{i-2,j} + \phi_{i,j-2} - 4\phi_{ij}] \tag{79}$$

The stencil for $DG\phi$ looks like:

$$\left(\frac{1}{2h^2}\right) \cdot$$

			1			
	1		-4		1	
			1			

Figure 48. The stencil for DGϕ

This looks somewhat like an extended five-point Laplacian, although it will be a little more tricky to implement multigrid and will require additional care at the boundaries. However, it is now a matter of applying known techniques, ultimately boiling down to easy linear algebra.

To verify the second-order accuracy of this method, let's denote the time level by *n* and the iteration level by *l*. Recall that we needed a guess for pressure gradient which lags the velocity field by half a time step. We set the first guess for gradient of pressure:

$$(Gp)^{n+1/2,\,0} = (Gp)^{n-1/2} \tag{80}$$

Then solve for the velocity by the fractional step method:

$$\underline{u}^{*\,l} = \underline{u}^n + \Delta t\left[-A(\underline{u})^{n+1/2} + \frac{\nu}{2}\Delta^h(\underline{u}^n + \underline{u}^*)^l - (Gp)^{n-1/2,\,l}\right] \tag{81}$$

and in the second step:

$$\underline{u}^{n+1} = P\underline{u}^{*\,l} \tag{82}$$

$$(Gp)^{n+1/2,\,l+1} = (Gp)^{n-1/2,\,l} + \frac{1}{\Delta t}(I-P)(\underline{u}^{*\,l} - \underline{u}^{n+1,\,l+1}) \tag{83}$$

Let's contrast the Bell-Colella-Glaz formulation above with the Crank-Nicholson approach. When starting at the same time, Crank-Nicholson evolves as follows,

$$\underline{u}^{CN} = \underline{u}^n + \Delta t\left[-A(\underline{u})^{n+1/2} + \frac{\nu}{2}\Delta^h(\underline{u}^n + \underline{u}^{CN}) - (Gp)^{CN}\right] \tag{84}$$

Comparing equations (84) and(81),

$$\underline{u}^{CN} - \underline{u}^* = \frac{\nu}{2}\Delta^h(\underline{u}^{CN} - \underline{u}^*) - (Gp)^{CN} + (Gp)^{n-1/2}$$

Rewriting

$$(I - \frac{\nu\Delta t}{2}\Delta^h)(\underline{u}^{CN} - \underline{u}^*) = -\Delta t\,((Gp)^{CN} - (Gp)^{n-1/2})$$

$$= O(\Delta t^2)$$

since the pressure gradient can only accumulate error of $O(\Delta t)$ in one time step. But what we'd really like to compare is $\underline{u}^{CN} - P\underline{u}^*$:

$$\underline{u}^{CN} - \underline{u}^{n+1} = P(\frac{\nu\Delta t}{2}\Delta^h(\underline{u}^{CN} - \underline{u}^*)) \tag{85}$$

and we have just shown above that the quantity of $\underline{u}^{CN} - \underline{u}^* = O(\Delta t^2)$. Numerical simulations also indicate that this scheme is of $O(\Delta t^2)$.

The tricky thing to compute in this technique is the advection term, $A^{n+1/2}$. We will use Godunov methodology, that is, extrapolate for u at the cell edge from the adjacent cell centers and find the upwind state. So, for the predictor step, we will extrapolate for velocity at, e.g., the right cell edge:

$$\underline{u}^{n+1/2}_{i+1/2,\,j} = \underline{u}^n_{ij} + \frac{\Delta x}{2}\frac{\partial \underline{u}}{\partial x} - \frac{\Delta t}{2}u\frac{\partial \underline{u}}{\partial x} - \frac{\Delta t}{2}v\frac{\partial \underline{u}}{\partial y}$$

$$+ \frac{\nu\Delta t}{2}\Delta\underline{u} - \frac{\Delta t}{2}\nabla p + O(\Delta x^2, \Delta t^2)$$

or:

$$\underset{\sim}{u}^{n+1/2}_{i+1/2,j} = \underset{\sim}{u}^{n}_{ij} + \frac{1}{2}\left[1 - \underset{\sim}{u}^{n}_{ij}\frac{\Delta t}{\Delta x}\right]\left.\frac{\partial \underset{\sim}{u}}{\partial x}\right|_{ij}\Delta x - \frac{\Delta t}{2} v_{ij}[u]^{y}_{upwind}$$
$$+ \left(\frac{\nu \Delta t}{2}(\Delta^{h}\underset{\sim}{u}^{n})_{ij} - \frac{\Delta t}{2}(Gp)^{n-1/2}\right) \tag{86}$$

where the second term on the RHS incorporates geometric limiting (cf. section xxx.xx.x); the third term uses the jump in the velocity component, *[u]*, in the *y* direction, choosing the upwind state (w.r.t. v_{ij}); and the fifth term approximates the pressure gradient term with accuracy up to $O(\Delta t^2)$.

There are a still a couple of problems with this approach which need to be resolved. For one, the use of the term $(Gp)^{n-1/2}$ in the predictor introduces a nonlinear instability, requiring a CFL condition on time step. We need,

$$\frac{|u|\Delta t}{\Delta x}, \frac{|v|\Delta t}{\Delta x} \le 0.5 \tag{87}$$

Defining the velocity unambiguously on the cell edges also creates some trouble because there is no reason to expect that $u_{i+1} - u_i = v_{j+1} - v_i$. As an example, advection of a scalar like density is given by:

$$\rho^{n+1}_{ij} = \rho^{n}_{ij} + \frac{\Delta t}{\Delta x}\left[u^{n+1/2}_{i-1/2,j}\rho^{n+1/2}_{i-1/2,j} - u^{n+1/2}_{i+1/2,j}\rho^{n+1/2}_{i+1/2,j}\right]$$
$$+ \frac{\Delta t}{\Delta y}\left[v^{n+1/2}_{i,j-1/2}\rho^{n+1/2}_{i,j-1/2} - v^{n+1/2}_{i,j+1/2}\rho^{n+1/2}_{i,j+1/2}\right] \tag{88}$$

But if the density is constant everywhere, i.e., $\rho^{n} \equiv \rho_0$, then we would like it to remain constant in time so that $\rho^{n+1} \equiv \rho_0$. However, this is not going to happen unless:

$$\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} = 0 \tag{89}$$

This version of the BCG technique does not guarantee this condition; thus, continuity may be violated. There are other implications for the momentum equation.

The next task is to find a way to eliminate the CFL condition imposed by the nonlinearity introduced by using $(Gp)^{n-1/2}$ and to fix the velocity flux. In the previous section we found that a basic feature of the BCG approach was to compute advective terms using time-centered cell edge velocities and then using a Riemann solver to find the upwind state. However, we ran into difficulties in maintaining a divergence-free velocity field and in handling the nonlinearities introduced by a lagging pressure gradient. The fix is apply the projection operator to the edge velocities; this will ensure that they will be divergence-free. In keeping with the spirit of terminology proliferation, we will denote this projection

of edge velocities as the *mac projection* (mesh-and-cell projection). In addition, we get another piece out of the projection for free, namely the gradient of a scalar. This will be used to solve the pressure gradient problem. The steps are as follows:

(a) Perform the inviscid predictor calculation, ignoring the ∇p term altogether. So, we extrapolate the velocity field to the cell edges and choose the upwind state to find the fractional-step values $\underset{\sim}{u}^*_{i+1/2,j}$ and $\underset{\sim}{u}^*_{i,j+1/2}$.

(b) Compute the MAC divergence of $\underset{\sim}{u}_{ij}$, as defined by edge velocities,

$$(D_{mac}\underset{\sim}{u}) = \frac{u^*_{i+1/2,j} - u^*_{i-1/2,j}}{\Delta x} + \frac{v^*_{i,j+1/2} - v^*_{i,j-1/2}}{\Delta y} \tag{90}$$

(c) Solve $\Delta^h \phi = D_{mac}\underset{\sim}{u}$ for ϕ, where Δ^h is the standard five-point Laplacian, subject to appropriate boundary conditions: if solid wall, then $\phi_{0,j} = -\phi_{1,j}$; if periodic, then $(\partial\phi)/(\partial\hat{n}) = 0$.

(d) Then solve for the time-centered velocity components at the cell edges,

$$u^{n+1/2}_{i+1/2,j} = u^*_{i+1/2,j} - \frac{\phi_{i+1,j} - \phi_{i,j}}{h} \tag{91}$$

$$v^{n+1/2}_{i,j+1/2} = v^*_{i,j+1/2} - \frac{\phi_{i,j+1} - \phi_{i,j}}{h} \tag{92}$$

where h is the mesh spacing. Now u and v satisfy $D_{mac}\underset{\sim}{u}^{n+1/2} = 0$

This takes care of the normal velocities at the cell edges. Now, what about the tangential ones? The simplest approach is, e.g., for the case of u at the top edge:

$$u^{n+1/2}_{i,j+1/2} = u^*_{ij} - \frac{1}{4h}\left[\phi_{i+1,j} + \phi_{i+1,j} + \phi_{i-1,j} + \phi_{i-1,j+1}\right] \tag{93}$$

The term in brackets approximates $\left.\frac{\Delta t}{2}\frac{\partial p}{\partial x}\right|_{i,j+1/2}$ and makes the approximation second order.

(e) Now finish off the extrapolation at the cell edges by examining what is coming in from either side of each edge. Coming into the right edge of cell *(i, j)* from the left is:

$$\underset{\sim}{u}^{n+1/2}_{i,j+1/2_L} = \underset{\sim}{u}^n_{ij} + \frac{1}{2}\left[1 - \frac{\Delta t}{\Delta x}\right]\Delta^x \underset{\sim}{u}_{ij} - \frac{v^n_{ij}\Delta t}{2\Delta y}\left[\underset{\sim}{u}\right]^2_{ij} + \frac{\nu\Delta t}{2}\left(\Delta^h \underset{\sim}{u}^n\right)_{ij} \tag{94}$$

where $\Delta^x \underset{\sim}{u}_{ij}$ are van Leer slopes and Δ^h is the five-point Laplacian. The term $[\underset{\sim}{u}]$ denotes the jump in $\underset{\sim}{u}$ and is calculated by:

$$v_{ij}^n [\underset{\sim}{u}]_{ij}^2 = \begin{cases} v_{ij}^n [\underset{\sim}{u}_{ij}^n - \underset{\sim}{u}_{i,j-1}^n] \ldots if \ldots (v_{ij}^n > 0) \\ v_{ij}^n [\underset{\sim}{u}_{i,j+1}^n - \underset{\sim}{u}_{ij}^n] \ldots if \ldots (v_{ij}^n < 0) \end{cases} \tag{95}$$

We obtain similar values coming from the right:

$$\underset{\sim}{u}_{i,j+1/2_R}^{n+1/2} = \underset{\sim}{u}_{i+1,j}^n + \frac{1}{2}\left[-1 - \frac{\Delta t}{\Delta x}\right]\Delta^x \underset{\sim}{u}_{i+1,j} - \frac{v_{ij}^n \Delta t}{2\Delta y}[\underset{\sim}{u}]_{i+1,j}^2 + \frac{\nu \Delta t}{2}(\Delta^h \underset{\sim}{u}^n)_{i+1,j} \tag{96}$$

(We have now accounted for everything properly in the Taylor expansion except for the $(\Delta t \nabla p)/2$ term. This will be taken care of in the second step.)

(f) Now, finally, choose the upwind state. Recall that we have gotten to this point by extrapolating to the cell edges from the right and left:

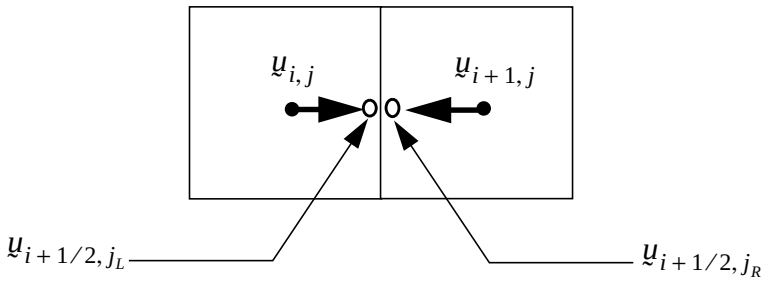


Figure 49. Extrapolation of cell-edge velocities from left and right

So, there are three possibilities:

$$\underset{\sim}{u}_{i+1/2,j} = \begin{cases} \underset{\sim}{u}_{i+1/2,j_L} \ldots if \ldots (\underset{\sim}{u}_{i+1/2,j_L}, \underset{\sim}{u}_{i+1/2,j_R} > 0) \\ \{ u_{i+1/2,j} = 0 \\ \quad v_{i+1/2,j} = v_{avg} \} \ldots if \ldots (\underset{\sim}{u}_{i+1/2,j_L} \cdot \underset{\sim}{u}_{i+1/2,j_R} < 0) \\ \underset{\sim}{u}_{i+1/2,j_R} \ldots if \ldots (\underset{\sim}{u}_{i+1/2,j_L}, \underset{\sim}{u}_{i+1/2,j_R} < 0) \end{cases} \tag{97}$$

where

$$v_{avg} = \frac{1}{2}\left[v_{i+1/2,j_L} + v_{i+1/2,j_R}\right] \tag{98}$$

8.3.6 Second-order cell-centered projection methods

What happens with cell-centered projections? Given an arbitrary discrete vector field, $\underset{\sim}{u}_{ij}$, defined at cell centers, we would like to apply the projection operator to obtain $\underset{\sim}{u}_{ij} = \underset{\sim}{u}_{d_{ij}} + (G\phi)_{ij}$. We will examine two cases: doubly periodic boundary conditions and solid wall boundaries (which means homogeneous Dirichlet BC on $\underset{\sim}{u}$ and homogeneous Neumann BC on ϕ). Assume that the number of mesh points in each direction, *nx* and *ny*, are both even. Assume further that the mesh spacing, *h*, is uniform in both directions. Recall that we obtain a Poisson-like equation which must be used to solve for ϕ:

$$DG\phi = D\underset{\sim}{u} \tag{99}$$

We define the gradient of ϕ in the *x* and *y* directions by:

$$(G\phi)^x_{ij} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h} \tag{100}$$

$$(G\phi)^y_{ij} = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h} \tag{101}$$

The discrete div grad ϕ can be written:

$$DG\phi = \frac{1}{4h^2}\left[\phi_{i+2,j} + \phi_{i,j+2} + \phi_{i-2,j} + \phi_{i,j-2} - 4\phi_{ij}\right] \tag{102}$$

For the case of doubly periodic boundary conditions, this is easy to extend periodically as required. However, the result is a set of four completely decoupled stencils:

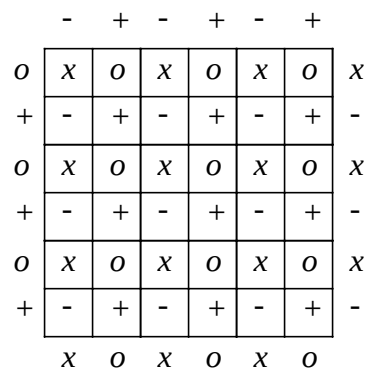


Figure 50. Completely decoupled stencils of DGϕ with periodic boundary conditions

The cells marked with *x* communicate only with other cells marked *x* at this grid level; likewise for cells marked with o, + and -. Well, what about multigrid which is supposed to eliminate short-wavelength components in the residual? Can we use this technique to our advantage? The answer is no, not right away. Let's focus on a grouping of four fine-mesh cells which is assumed to be repeated throughout the domain, shown in Figure 51:

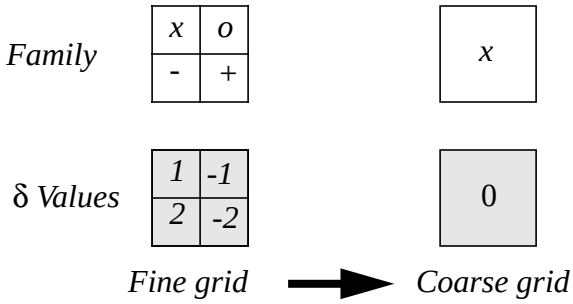


Figure 51. Detail of a decoupling example that is problematic for multigrid

We would like to relax on the fine grid to eliminate short-wavelength components in the quantities *R* and δ. However, note that for this example, $(DG\delta)_{ij} = 0$. Also, the average of the fine-grid cells onto the coarse grid would be zero.Therefore, in this case multigrid will not smooth the error.

The second boundary condition under consideration, i.e., that of solid walls, is equivalent to extending the solution in the following way:

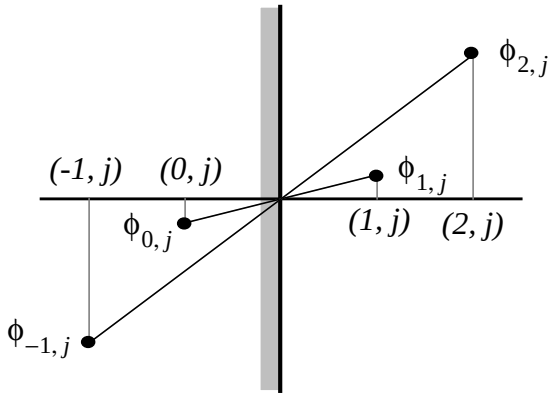


Figure 52Phantom nodes and relation to interior points for solid-wall boundary conditions

Linear interpolation is performed between nodes in the interior of the problem domain and phantom nodes which are inside the boundary to create a value of zero at the wall (i.e., $\phi_{-1,j} = -\phi_{2,j}$ and $\phi_{0,j} = -\phi_{1,j}$). The good news is that there are no more decoupled

meshes since everything couples in a global sense at the boundaries, as shown below in Figure 53. For example, we need to compute $DG\phi$ at the upper left corner of the domain (the dark circle of type *x*). Information from the interior of the domain comes only from the *x* family itself; however, the phantom nodes within the boundary have also taken marching orders from the "*o*" and "-" families and so can communicate their evil intent to family *x*.

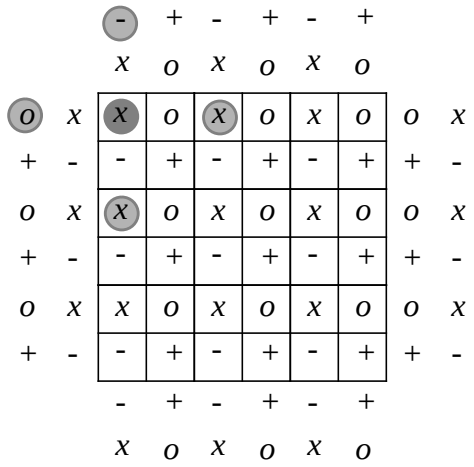


Figure 53. Four families of cells communicate at the borders with solid-wall boundary conditions

However, as can be seen from the above, relaxation is a local process. The updating of the solution of ϕ in the interior of the domain must wait for numerical "diffusion" to take place from the information at the boundaries. This also merits a big gong for multigrid.

We might next ask if there is some way around this coupling/decoupling problem. We will try to fix this deficiency by using local decoupling. Two approaches follow. First, we can try to mimic doubly periodic BC by using nonstandard averaging and interpolation operators which respect the coupling/decoupling. For example, in going from the fine grid to the coarse grid, we might simply use adjacent cells of the same family to compute the coarse-grid residual. In this case, the coarse-grid residual is given by,

$$R_o^c = \frac{1}{4}\sum_{i=1}^{4} R_{o_i}^f \tag{103}$$

where *f* and *c* denote fine- and coarse-grid values, respectively. The fine-grid values used to interpolate for the coarse-grid value in the dark circle are shown with arrows and fingers in Figure 53, below. The quantity $DG\phi$ calculated at the solid-wall boundaries still communicates with the other families and provides the appropriate coupling. Since we're

at the coarser grid level, the information “diffusion” from the boundary cells will infiltrate into the interior more readily than for the fine grid.

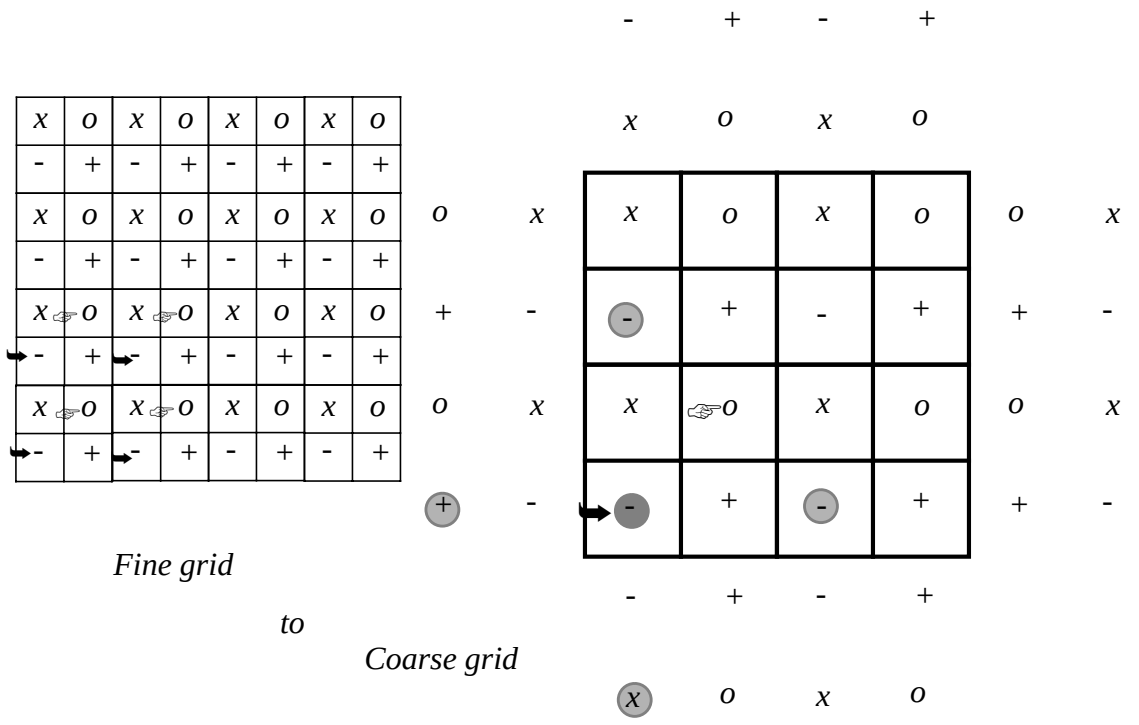


Figure 54. Local decoupling approach applied to multigrid

A second approach to eliminate decoupling is by finding an appropriate discrete divergence and gradient such that $DG\phi$ only has constant solutions for doubly periodic boundary conditions. For this case, the discrete divergence is defined by:

$$(D\underline{u})_{ij} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{h} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{h} \tag{104}$$

where the cell edge velocities may be defined by:

$$u_{i+1/2,j} = \frac{1}{2}(u_{ij} + u_{i+1,j}) - \alpha (D^x_+ \; D^x_- \; u)_{ij} \tag{105}$$

$$v_{i,j+1/2} = \frac{1}{2}(v_{ij} + v_{i,j+1}) - \alpha (D^y_+ \; D^y_- \; v)_{ij} \tag{106}$$

and

$$(D^x_+ \; D^x_- \; u)_{ij} = u_{i+1,j} - 2u_{ij} + u_{i-1,j} \tag{107}$$

Now why do we need this extra notation? The second term on the RHS of equations (106) and (107) above comes from carrying higher-order terms in the Taylor expansion performed in the discretization. The gradient in the *x* direction is:

$$\frac{\partial u}{\partial x} \approx (D_o u)_{ij} - \alpha h D_- \ (D^x_+ \ D^x_- \ u) \tag{108}$$

Thus, the second term on the RHS of equation (108) above is an approximation to

$$\alpha h^2 \frac{\partial^3 u}{\partial x^3}$$

Next, the discrete gradient is related to the divergence by the adjoint condition so that $G = -D^T$. To define *G:*

$$(G\phi)^x_{ij} = \frac{\phi_{i+1/2,j} - \phi_{i-1/2,j}}{h} \tag{109}$$

where

$$\phi_{i+1/2,j} = \frac{1}{2}(\phi_{ij} - \phi_{i+1,j}) - \alpha (D^x_+ \ D^x_- \ u)_{i+1,j} \tag{110}$$

Next we need to solve the equation:

$$DG\phi = D\underset{\sim}{w} \tag{111}$$

and

$$\underset{\sim}{w}_d = \underset{\sim}{w} - G\phi \tag{112}$$

When $DG\phi$ is zero, ϕ is a constant. This can be shown by using a discrete Fourier transform to diagonalize *DG*.

We expect that the short-wavelength modes would be damped by point relaxation using Gauss-Seidel with red/black ordering. For this case, in the first sweep the even cells are updated,

$$\phi^{l+1/2}_{ij} = \begin{cases} \phi^l_{ij} + \lambda (\rho - DG\phi^l)_{ij} \ldots if \ldots (i+j) = even \\ \phi^l_{ij} \ldots if \ldots (i+j) = odd \end{cases} \tag{113}$$

Then perform a second sweep to get from iteration level *l+1/2* to *l+1* by reversing even and odd in equation (113) to update the odd cells (see the discussion in section xxx.x.x.). If we choose the "time" step parameter λ so as to eliminate the diagonal terms in the matrix $DG\phi$, we don't completely eliminate the high wavenumber modes, but we do damp them. For $\lambda = -1/c$, we can write,

$$(DG\phi)_{ij} = c\phi_{ij} + \sum_s \phi_s \tag{114}$$

This is not the same as the $2\Delta x$ mode which we discussed in section xxx.xx.x previously.

Now, let's discuss the solid-wall boundary. The discrete divergence is defined as before for the doubly periodic case, but the discrete gradient loses accuracy near the boundaries. This manifests itself as a failure in multigrid. So, now comes the moment when we will relax the condition $G = -D^T$ at the boundary, but we must somehow reconcile with the solvability condition.

Let's backtrack for a moment. Recall that to find the divergence-free velocity field by hitting it with a projection, we wanted to solve $\underset{\sim}{w}_d = P\underset{\sim}{w}$. The first step was to solve $DG\phi = D\underset{\sim}{w}$ to obtain ϕ so that we could compute $\underset{\sim}{w}_d = \underset{\sim}{w} - G\phi$. Another way to write the projection operator is:

$$P\underset{\sim}{w} = (I - G(DG)^{-1}D)\,\underset{\sim}{w} \tag{115}$$

Recall that $P^2 = P$, but the conjugate transpose $P^* \neq P$. The Fredholm alternative tells us that $DG\phi = D\underset{\sim}{w}$ is solvable if, given that, $(DG)^T\phi = 0$ then $\langle D\underset{\sim}{w}, \phi\rangle = 0$. We can reason this out in the following way: if $(DG)^T\phi = 0$, then $DG\phi = 0$. This means that ϕ is a constant, say C. It also means that $\langle D\underset{\sim}{w}, \phi\rangle = 0$ if it is written in conservation form.

References

A. M. Abd-el-Fattah, L. F. Henderson and A. Lozzi (1976) “Precursor Shock Waves at a Slow-Fast Gas Interface, *J. Fluid Mech.* **76**, 157-176.

A. M. Abd-el-Fattah and L. F. Henderson (1978) “Shock Waves at a Fast-Slow Gas Interface”, *J. Fluid Mech.* **86**, 15-32.

A. M. Abd-el-Fattah and L. F. Henderson (1978) “Shock Waves at a Slow-Fast Gas Interface”, *J. Fluid Mech.* **89**, 79-95.

A. S. Almgren and J. B. Bell and P. Colella and L. H. Howell (1993) “An Adaptive Projection Method for the Incompressible Euler Equations”, Proceedings of the 11th AIAA Computational Fluid Dynamics Conference, Orlando, FL.

A. S. Almgren and J. B. Bell and P. Colella and T. Marthaler (1994) “A Cartesian Grid Projection Method for the Incompressible Euler Equations” (manuscript) Lawrence Livermore National Laboratory

J. B. Bell, C. N. Dawson and G. R. Shubin (1988) “An Unsplit Higher Order Godunov Method for Scalar Conservation Laws in Multiple Dimensions”, *J. Comp. Phys.* **74**, 1-24.

J. B. Bell, P. Colella, and H. M. Glaz (1989) “A Second-Order Projection Method for the Incompressible Navier-Stokes Equations”, *J. Comput. Phys.* **85**, 257-283.

J. B. Bell, P. Colella, and J. A. Tragenstein (1989) “Higher-Order Godunov Methods for General Systems of Hyperbolic Conservation Laws”, *J. Comput. Phys.* **82**, 362-397.

J. B. Bell, H. M. Glaz, J. M. Solomon, and W. G. Szymczak, “Application of a Second-Order Projection Method to the Study of Shear Layers”, in the *Proceedings of the 11th International Conference on Numerical Methods in Fluid Dynamics*, Williamsburg, VA, June 27-July 1, 1988.

J. B. Bell, and D. L. Marcus (1992) “A Second-Order Projection Method for Variable Density Flows”, *J. Comput. Phys.* **101**, 334-348.

J. B. Bell, and D. L. Marcus (1992) “Vorticity intensification and transition to turbulence in the three-dimensional Euler equations”, *Comm. Math Phys.* **147**, 371-394.

J. B. Bell, M. J. Berger, J. Saltzman, and M. Welcome (1994) “Three Dimensional Adaptive Mesh Refinement for Hyperbolic Conservation Laws”, *SIAM J. Sci. Comput.* **15**, 127-138.

M. J. Berger and P. Colella (1989) “Local Adaptive Mesh Refinement for Shock Hydrodynamics”, *J. Comput. Phys.*, **82**, 64-84.

J. U. Brackbill, D. B. Kothe and C. Zemach (1992) “A Continuum Method for Modeling Surface Tension”, *J. Comput. Phys.* **100**, 335-354.

D. L. Brown (1993) “An Unsplit Godunov Method for Systems of Conservation Laws on Curvilinear Overlapping Grids”, **LA-UR-93-2868**, Los Alamos National Laboratory.

D. L. Brown and G. Chesshire and W. D. Henshaw (1990) “Getting Started with CMPGRD, Introductory User's Guide and Reference Manual”, **LA-UR-90-3729**, Los Alamos National Laboratory.

G. Chesshire and W. D. Henshaw (1990) “Composite Overlapping Meshes for the Solution of Partial Differential Equations”, *J. Comput. Phys.*, **90**(1), 1-64

G. Chesshire and W. D. Henshaw (1994) “A Scheme for Conservative Interpolation on Overlapping Grids”, *SIAM J. Sci. Comput.* **15**(4), 819-845

A. J. Chorin, *Numerical Solution of the Navier-Stokes Equations,* Math. Comput. **22** (1968), 745-762

A. J. Chorin, *On the Convergence of Discrete Approximations to the Navier-Stokes Equations,* Math. Comput. **23** (1969), 341-353

A. J. Chorin, *Flame Advection and Propagation Algorithms* J. Comput. Phys. **35** (1980), 1-11

A. J. Chorin, *Curvature and Solidification,* J. Comput. Phys. **57** (1985), 472-490

A. J. Chorin and J. E. Marsden, “A Mathematical Introduction to Fluid Mechanics”, (3rd ed) Springer-Verlag (1992)

P. Colella, *A Direct Eulerian MUSCL Scheme for Gas Dynamics* (1985) SIAM J. Sci. Stat. Comput. **6**), 104-117

P. Colella, and P. Woodward (1984) *The Piecewise Parabolic Method (PPM) for Gas Dynamical Simulations,* J. Comput. Phys. **54**, 174-201

P. Colella (1990) *Multidimensional Upwind Methods for Hyperbolic Conservation Laws,* J. Comput. Phys. **87**, 171-200

P. Colella, L. F. Henderson, and E. G. Puckett (1989) *A Numerical Study of Shock Wave Refraction at a Gas Interface,* Proceedings of the AIAA 9th Computational Fluid Dynamics Conference, Buffalo, New York, 426-439

D. S. Dandy and L. G. Leal (1989) “Buoyancy-driven motion of a deformable drop through a quiescent liquid at intermediate Reynolds numbers”, *J. Fluid Mech.* **208**, 161.

D. S. Dandy and H. A. Dwyer (1990) “A Sphere in Shear Flow at Finite Reynolds Numbers: Effect of Shear on Particle Lift, Drag, and Heat Transfer”, *J. Fluid. Mech.*, **216**,381-410.

R. DeBar (1974) “A Method in Two-D Eulerian Hydrodynamics”, **UCID-19683**, Lawrence Livermore National Laboratory.

N. V. Deshpande (1989) “Fluid Mechanics of Bubble Growth and Collapse in a Thermal Ink-Jet Printhead”, SPSE/SPIES Electronic Imaging Devices and Systems Symposium, January 1989.

H. A. Dwyer (1989) “Calculation of Droplet Dynamics in High Temperature Environments”, *Prog.*

Energy Combust. Sci., **15,** 131-158.

S. A. Elrod, B. Hadimioglu, B. T. Khuri-Yakub, E. G. Rawson, E. Richley, C. F. Quate, N. N. Mansour and T. S. Lundgren (1989) Nozzleless Droplet Formation with Focused Acoustic Beams, *J. Appl. Phys.* **65**(9), 3341-3347.

A. F. Ghoniem, A. J. Chorin and A. K. Oppenheim (1982) *Numerical Modeling of Turbulent Flow in a Combustion Tunnel,* Phil. Trans. R. Soc. Lond. **A 304**, 303-325.

L. F. Henderson (1966) *The Refraction of a Plane Shock Wave at a Gas Interface,* J. Fluid Mech. **26**, 607-637.

L. F. Henderson (1989) *On the Refraction of Shock Waves,* J. Fluid Mech. **198**, 365-386.

L. F. Henderson, E. G. Puckett, and and P. Colella, *On the Anomalous Refraction of Shock Waves,* Proceedings of the Second Japan-Soviet Union Symposium on Computational Fluid Dynamics, Tsukuba, Japan, August 27-31, 1990

L. F. Henderson, P. Colella, and E. G. Puckett (1991) "On the Refraction of Shock Waves at a Slow-Fast Gas Interface", *J. Fluid Mech.* **224**, 1-27.

L. F. Henderson, E. G. Puckett and P. Colella (1992) Anomalous Refraction of Shock Waves, In *Shock Waves,* K. Takayama (ed.), Springer Verlag, 283-286.

[Hi] C. W. Hirt and B. D. Nichols (1981) "Volume-of-Fluid (VOF) Method for the Dynamics of Free Boundaries, *J. Comput. Phys.* **39**, 201-225.

R. S. Hotchkiss (1979) "Simulation of Tank Draining Phenomena with the SOLA-VOF Code", **LA-8163-MS**, Los Alamos National Laboratory.

K. S. Holian, S. J. Mosso, D. A. Mandell and R. Henninger (1991) "MESA: A 3-D Computer Code for Armor/Anti-Armor Applications", **LA--UR--91-569**.

J. M. Hyman (1984) *Numerical Methods for Tracking Interfaces*, Physica **12D**, 396-407.

J. Kim and P. Moin (1985) *Application of a Fractional-Step Method to the Incompressible Navier-Stokes Equations,* J. Comput. Phys. **59**, 308-323

D. B. Kothe, R. C. Mjolsness and M. D. Torrey (1991) "RIPPLE: A Computer Program for Incompressible Flows with Free Surfaces", **LA-12007-MS**, Los Alamos National Laboratory.

D. B. Kothe, J. R. Baumgardner, S. T. Bennion, J. H. Cerutti, B. J. Daly, K. S. Holian, E. M. Kober, S. J. Mosso, J. W. Painter, R. D. Smith and M. D. Torrey, (1992) "PAGOSA: A Massively-Parallel, Multi-Material Hydro-Dynamics Model for Three-Dimensional High-Speed Flow and High-Rate Deformation", **LA-UR-92-4306**.

R. J. LeVeque "Numerical Methods for Conservation Laws", Birkhauser (1992)

D. L. Marcus, and J. B. Bell (1992) "The Structure and Evolution of the Vorticity and Temperature

Fields in Thermals", *Theoretical and Comput. Fluid Dynamics* **3**, 327-344

D. L. Marcus, and J. B. Bell (1994) "Numerical Simulation of a Viscous Vortex Ring Interaction with a Density Interface", *Physics of Fluids A* (in press)

D. L. Marcus, E. G. Puckett, J. B. Bell, and J. Saltzman (1991) "Numerical Simulation of Accelerated Interfaces', In *Proc. 3rd International Workshop on the Physics of Compressible Turbulent Mixing*, R. Dautray (ed.), Royaumont, France, 63-81.

G. H. Miller and E. G. Puckett (1994) Edge Effects in Molybdenum-Encapsulated Molten Silicate Shock Wave Targets, *J. Appl. Phys.* **75**(3), 1426-1434.

W. Mulder, S. Osher, and J. A. Sethian, *Computing Interface Motion in Compressible Gas Dynamics,* submitted to J. Comput. Phys.

B. D. Nichols, C. W. Hirt and R. S. Hotchkiss (1980) "SOLA-VOF: A Solution Algorithm for Transient Fluid Flow with Multiple Free Boundaries", **LA-8355** Los Alamos National Laboratory.

W. F. Noh and P. R. Woodward, *SLIC (Simple Line Interface Method),* in Lecture Notes in Physics **59**, A. I. van de Vooren and P. J. Zandbergen (ed.), Springer Verlag, Berlin (1976).

S. Osher and J. A. Sethian (1988) *Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations,* J. Comput. Phys. **79**, 12-49.

B. J. Parker and D. L. Youngs (1992) "Two and Three Dimensional Eulerian Simulation of Fluid Flow with Material Interfaces", AWE Preprint **01/92** UK Atomic Weapons Establishment.

N. A. Petersson and J. F. Malmliden (1993) "Computing the flow around a submerged body using composite grids", *J. Comput. Phys.*, **105**, 47-57.

J. E. Pilliod (1992) "An Analysis of Piecewise Linear Interface Reconstruction Algorithms for Volume-Of-Fluid Methods" *Masters Thesis,* U. C. Davis, September 1992.

J. E. Pilliod and E. G. Puckett "Second-Order Volume of Fluid Algorithms for Tracking Material Interfaces" In preparation for submittal to *J. of Comput. Physics*.

E. G. Puckett, L. F. Henderson, and P. Colella (1989) *Computations of the Refraction of a Plane Shock Wave at a Slow-Fast Gas Interface,* Proceedings of the 17th International Symposium on Shock Waves and Shock Tubes, Lehigh University, Bethlehem, PA, July 17-21, 1989

E. G. Puckett (1991) "A Volume of Fluid Interface Tracking Algorithm with Applications to Computung Shock Wave Refraction", in *Proc. 4th Int. Sym. on Comp. Fluid Dynamics,* H. A. Dwyer (ed.), Davis, California.

E. G. Puckett (1991) "A Numerical Study of Shock Wave Refraction at a CO_2/CH_4 Interface", In *Multidimensional Hyperbolic Problems and Computations*, Volumes in Mathematics and Its Applications **29**, J. Glimm and A. J. Majda (ed.), Springer Verlag, New York, 261-280.

E. G. Puckett and J. Saltzman (1992) "A 3-D Adaptive Mesh Refinement Algorithm for Multi-Ma-

terial Gas Dynamics Mixing", *Physica D* **60**, 84-93.

E. G. Puckett, L. F. Henderson, and P. Colella (1994) "A General Theory of Anomalous Refraction", in *Proc. 19th International Symposium on Shock Waves*, R. Brun (ed.) (to appear).

E. G. Puckett, D. L. Marcus, J. B. Bell, and A. S. Almgren (1994) "A Projection Method for Multi-Fluid Flows with Interface Tracking", (manuscript) Lawrence Livermore National Laboratory.

W. J. Rider (1994) "Approximate Projection Methods for Incompressible Flow: Implementation, Variants and Robustness" (manuscript) Los Alamos National Laboratory

J. A. Sethian (1982) *An Analysis of Flame Propagation* PhD Thesis, U. C. Berkeley.

J. A. Sethian (1984) *Turbulent Combustion in Open and Closed Vessels*, J. Comput. Phys. **54**, 425-456.

J. A. Sethian (1985) *Curvature and the Evolution of Fronts*, Comm. Math. Phys. **101**, 487-499.

J. A. Sethian (1990) *Numerical Algorithms for Propagating Interfaces: Hamilton-Jacobi Equations and Conservation Laws*, J. Differential Geometry **31**, 131-161

G. I. Taylor (1934) *Proc. Roy. Soc.* London A **146** 501.

P. A. Torpey (1988) "Prevention of Air Ingestion in a Thernmal Ink-Jet Device", 4th International Congress on Advances in Non-Impact Print Technologies, March 1988.

M. D. Torrey, R. C. Mjolsness and L. R. Stein (1987) "NASA-VOF3D: A Three-Dimensonal Computer Program for Incompressible Flows with Free Surfaces", **LA-11009-MS**, Los Alamos National Laboratory.

S. O. Unverdi and G. Trygvasson (1992) "A Front Tracking Method for Viscous Incompressible Flows", *J. Comput. Physics*, **100**, 25-37.

B. van Leer (1979) "Towards the Ultimate Conservative Difference Scheme, A Second Order Sequel to Godunov's Method", *J. Comput. Phys*. **32**, 101-136.

B. van Leer (1984) "On the Relation Between the Upwind Schemes of Godunov, Enquist-Osher and Roe", *JIAM J. Sci. Stat. Comput.* **5**, 1-20.

D. B. Wallace (1989) "A Method of Characteristics Model of a Drop on-Demand Ink-Jet Device Using an Integral Method Drop Formation Model", **89-WA/FE-4**, ASME Annual Winter Meeting, December 10-15, 1989.

P. R. Woodward and P. Colella (1984) "The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks", *J. Comput. Phys.* **54**, 115-173.

H. C. Yee (1986) "Upwind and Symmetric Shock-Capturing Schemes" in *Proceedings of the Seminar on Computational Aerodynamics*, M. Hafez (ed.), U. C. Davis, Davis, CA

D. L. Youngs (1982) "Time-Dependent Multi-Material Flow with Large Fluid Distortion", in *Nu-*

merical Methods for Fluid Dynamics, K. W. Morton and M. J. Baines (ed.), Academic Press, London, 273-285

D. L. Youngs (1984) “Numerical Simulation of Turbulent Mixing by Rayleigh-Taylor Instability”, *Physica* **12D**, 32-44

S. T. Zalesak (1989) “Fully Multidimensional Flux-Corrected Transport Algorithms for Fluid”, *J. Comput. Phys.* **31**, 335-362