## 2.3.5. Stability Analysis

Reading: Tannehill et al. Section 3.6.

**Stability** – For a stable numerical scheme, the errors in the initial condition will not grow unboundedly with time.

In this section, we discuss the methods for determining the stability of F.D. schemes. This is very important when designing a F.D. scheme and for understanding it behavior.

There are several methods:

- Energy method
- von Neumann method
- Matrix method (for systems of equations)
- Discrete perturbation method (will not discuss)

Note: Stability refers to the F.D.
Does not involve B.C. or I.C.
Refers to time-matching problems only

### The energy method

Read Durran Section 2.2 (handout, see web link)

This method is used much less often than the von Neumann method.

It's attractive because it works for nonlinear problem and problems without period B.C.

The key is to show that a positive definite quantity like $\sum_i (u_i^n)^2$ is bounded for all n.

We illustrate this method using the upstream-forward scheme for wave (advection) equation $\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0$$

Let $\mu = c\Delta t/\Delta x$ →

$$u_i^{n+1} = (1-\mu)u_i^n + \mu u_{i-1}^n$$

Squaring both sides and summing over all grid points:

$$\sum_i (u_i^{n+1})^2 = \sum_i [(1-\mu)^2 (u_i^n)^2 + 2(1-\mu)\mu u_i^n u_{i-1}^n + \mu^2 (u_{i-1}^n)^2] \qquad (1)$$

Assuming periodic B.C. →

$$\sum_i (u_i^n)^2 = \sum_i (u_{i-1}^n)^2$$

and using the Schwarz inequality (which says that for two vectors U and V, $|U \cdot V| \le |U| \cdot |V|$ ),

$$\sum_i u_i^n u_{i-1}^n \le \sqrt{\sum_i (u_i^n)^2}\ \sqrt{\sum_i (u_{i-1}^n)^2} = \sum_i (u_i^n)^2 .$$

If $\mu(1-\mu) \ge 0$, all coefficients of RHS terms in (1) are positive, and we have

$$\sum_i (u_i^{n+1})^2 \le [(1-\mu)^2 + 2(1-\mu)\mu + \mu^2] \sum_i (u_i^n)^2 = \sum_i (u_i^n)^2 ,$$

i.e., the L2 norm at n+1 is no greater than that at n, therefore the scheme is stable!

The condition $\mu(1-\mu) \ge 0$ gives

$$\mu = c\Delta t/\Delta x \ \le 1$$

which is the stability condition for this scheme. As we discussed earlier, it says that waves cannot propagate more than one grid interval during one $\Delta t$ in order to maintain stability.

**von Neumann method**

Read Tannehill et al, Section 3.6.1.

In a sense, we have already used this method to find stability of the 1-D diffusion equation – when we were proving the convergence of the solution using FTCS scheme.

We found then

$$u^{n+1} = [\ M(t)\ ]^{n+1}\ u^0$$

where M(t) = amplification factor.

If $M(t) \leq 1$ by some measure (M can be a matrix or a complex number), then $u^{n+1} \leq u^n$ and the solution cannot grow in time – the scheme is stable.

Essentially, von Neumann method expands the F.D.E. in a Fourier series, finds the amplification factor and determines under what condition the factor is less than or equal to 1 for stability.

Assumptions:

1. The equation has to be Linear with constant coefficients.
2. It is assumed that the solution is periodic.

With this method, the dependent variable is decomposed into a complex (or a real) Fourier series:

$$u(x,y,z,t) = \sum_{k,l,m} U(k,l,m)\exp[i(kx + ly + mz - \omega t)] \qquad (2)$$

where U is the complex amplitude and $\omega = \omega_R + \omega_I$ is the complex frequency.

In fact $\omega_R$ gives the wave propagation speed and $\omega_I$ gives the growth and decaying rate.

$$e^{-i\omega t} = e^{-i[\omega_R + i\omega_I]t} = e^{-i\omega_R t} e^{\omega_I t}$$

$e^{-i\omega_R t}$ - phase function of Fourier components
$e^{\omega_I t}$ - growth or decay rate

If $\omega_I > 0$, the solution will grow exponentially in time.

**Example 1**: 1-D diffusion equation with FTCS scheme.

$$u_i^{n+1} - u_i^n = \mu(u_{i-1}^n - 2u_i^n + u_{i+1}^n) \qquad (3)$$

where $\mu = \dfrac{K\Delta t}{(\Delta x)^2}$.

Let's examine a single wave k:

$$u_i^n = U_k e^{i(kx-\omega t)}$$
$$u_i^{n+1} = U_k e^{i(kx-\omega t)} e^{-i\omega\Delta t}$$
$$u_{i\pm1}^n = U_k e^{i(kx-\omega t)} e^{\pm ik\Delta x}$$

Substitute the above into (3) →

$$U_k e^{i(kx-\omega t)}(e^{-i\omega\Delta t} - 1) = \mu U_k e^{i(kx-\omega t)}[e^{ik\Delta x} - 2 + e^{-ik\Delta x}] \rightarrow$$

$$U_k e^{i(kx-\omega t)}[e^{-i\omega\Delta t} - 1 - 2\mu(\cos k\Delta x - 1)] = 0 \rightarrow$$

$$U_k e^{i(kx-\omega t)}[e^{-i\omega\Delta t} - 1 + 4\mu \sin^2(k\Delta x/2)] = 0$$

For non-trivial solution, we require

$$e^{-i\omega\Delta t} - 1 + 4\mu \sin^2(k\Delta x/2) = 0 \rightarrow$$

$$e^{-i\omega\Delta t} = 1 - 4\mu \sin^2(k\Delta x/2)$$

Here, $e^{-i\omega\Delta t}$ is actually the amplification factor, the same as the M discussed earlier.

$\lambda \equiv e^{-i\omega\Delta t} = u^{n+1}/u^n$ - the amplification factor.

For stability we require

$|\lambda| \le 1 \rightarrow$

$$-1 \le 1 - 4\mu \sin^2(k\Delta x/2) \le 1 \rightarrow$$

$\mu \le 1/2$ as before!

In practice, when $\mu$=1/2, the solution (amplification factor) switches between –1 and +1 for $2\Delta x$ waves ($\sin^2(k\Delta x/2) = 1$) very other step, which is unrealistic. The standard requirement is therefore

$\mu \le 1/4$.

Therefore, do not naively think diffusion terms in a numerical model does not cause numerical instability! When integrated stably, the diffusion term in CFD models tends to stabilize the solution by killing off/damping small scale waves, but when stability condition is not met, it tem itself will cause problem!

Read Pielke (1984) section 10.1.2 (handout, see web link).

### 2.3.6. Implicit Methods

Read Tannehill et al, second part of section 3.4.1.

So far, we have dealt with only explicit schemes which have the form of

$$u^{n+1} = f(u^n, u^{n-1}, ...).$$

With these schemes, the future state at each grid point is only dependent on the current and past time levels, therefore the solution can be obtained directly or explicitly. c.f., explicit functions such as $y = x^2$.

Implicit scheme involves variables of the future time level at more than one grid point (often resulting from finite difference of variable(s) at the future time level). Mathematically it can be expressed as

$$u^{n+1} = f(u^{n+1}, u^n, u^{n-1}, ...).$$

This is analogous to implicit functions such as $x = \sin(x)$. As one can imagine, implicit schemes are more difficult to solve. Usually matrix inversion is involved.

We will first look at the stability property of an implicit scheme.

**Example**. Consider the 1-D diffusion equation $u_t = K u_{xx}$ again.

It is approximated by the following F.D. scheme:

$$\delta_{+t} u_i = K[\alpha \delta_{xx} u_i^{n+1} + (1-\alpha)\delta_{xx} u_i^n] \qquad (4)$$

(Note: shorthand notations for F.D. are used. See Appendix. e.g., $\delta_{+t} u = (u^{n+1} - u^n)/\Delta t$ ).

- When $\alpha$ = 0, the scheme is explicit, and is the FTCS scheme discussed earlier.
- When $\alpha$ = 1/2, it is implicit and is called Crank-Nicolson scheme.
- For other values of $\alpha$, it is a general implicit scheme.

We can show that

$$\tau = K \frac{\partial^4 u}{\partial x^4}\left[ K\Delta t\left(\frac{1}{2} - \alpha\right) - \frac{(\Delta x)^2}{12}\right] + O(\Delta x^4 + \Delta t^2)$$

(show if for yourself!).

We can see that when $\alpha$ = 1/2, it is 2nd-order accurate in time and space. Otherwise, it's first-order in time – which is expected for un-centered time-differencing scheme (when $\alpha$=1/2, the right hand side is an averaged between the current and future time levels valid at n+1/2. Relative to this RHS, the LHS time difference becomes centered in time. We know that the simplest centered difference scheme is second-order accurate).

When [ ] = 0, the scheme becomes fourth-order in space.

Let's perform stability analysis on (4) using von Neumann method.

$$u_j^n = U_k e^{i(kx-\omega t)} = U_k e^{-i\omega n\Delta t} e^{ikj\Delta x} \equiv U_k \lambda^n e^{ikj\Delta x} \qquad (5)$$

Here $\lambda \equiv e^{-i\omega\Delta t}$. Note that we are now using j as the grid point index.

Substitute (5) into (4) →

$$U_k e^{ikj\Delta x}(\lambda^{n+1} - \lambda^n) = \mu U_k e^{ikj\Delta x}\left[\alpha\lambda^{n+1} + (1-\alpha)\lambda^n\right](e^{ik\Delta x} - 2 + e^{-ik\Delta x}) \rightarrow$$

Dividing $U_k e^{ikj\Delta x}\lambda^n$ on both sides, and rearranging →

$$\lambda - 1 = -4\mu\sin^2(k\Delta x/2)[\alpha\lambda + (1-\alpha)]$$

$$\lambda = \frac{1 - 4(1-\alpha)\mu\sin^2(k\Delta x/2)}{1 + 4\alpha\mu\sin^2(k\Delta x/2)}$$

Look at several cases:

Case I: $\alpha = 0$, $\lambda = 1 - 4\mu\sin^2(k\Delta x/2)$ → $\mu \le 1/2$ as before. The scheme is <u>conditionally</u> stable.

Case II: $\alpha = 1/2$ (Crank-Nicolson)

$$|\lambda| = \left|\frac{1 - 2\mu\sin^2(k\Delta x/2)}{1 + 2\mu\sin^2(k\Delta x/2)}\right| \le 1 \quad \text{for all values of } \mu,$$

therefore the scheme is <u>absolutely or unconditionally</u> stable.

Case III: $\alpha = 1$, the time difference is backward, relative to the RHS terms.

$$|\lambda| = \left| \frac{1}{1 + 4\mu \sin^2(k\Delta x/2)} \right| \le 1$$, again for all values of μ,

therefore the scheme is also absolutely stable. However, this scheme is only first-order accurate in time, as discussed earlier (consistent with the time difference scheme being un-centered).

In general, when $0 \le \alpha < 1/2$, it is required that $\mu \le 1/(2 - 4\alpha)$, therefore the scheme is conditionally stable. When $1/2 \le \alpha \le 1$, the scheme is unconditionally stable (it is sometimes referred to as the forward-biased scheme).

In the ARPS, the implicit diffusion scheme is an option for treating the vertical turbulent mixing terms. This treatment is necessary in order to remove the severe stability constraint from these terms when vertical mixing is strong inside the planetary boundary layer (PBL). The latter occurs when the PBL is convectively unstably and the non-local PBL mixing is invoked with the Sun and Chang (1986) parameterization. Parameter *alfcoef* in arps.input corresponds to 1-α here (see handout).

Finally, we note that for multi-time level schemes, there is usually multiple solutions for the amplification factor λ. some of them might represent spurious computational modes due to the use of extra (artificial) initial conditions.

The expression of λ can be too complicated so that a graphic plotting is needed to understand its dependency on wave number k. |λ| has to be no greater than 1 for all possible waves. The shortest wave resolvable on a grid has a wavelength of 2 Δ, and the longest is 2L, where L is the domain width.

Tridiagonal Solver

1-D implicit method often leads to tridiagonal systems of linear algebraic equations.

(In the ARPS, this appears twice – once when sound waves are treated implicitly in the vertical direction and once when the vertical turbulence mixing is treated implicitly).

For example, Eq. (4) can be rewritten as

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\alpha K\frac{u_{i-1}^{n+1}-2u_i^{n+1}+u_{i+1}^{n+1}}{\Delta x^2}+f_i^n \rightarrow \qquad (6)$$

where $f_i^n$ contains other known times at time level *n+1*. It can be rearranged into

$$u_i^{n+1}=\alpha\frac{\Delta tK}{\Delta x^2}(u_{i-1}^{n+1}-2u_i^{n+1}+u_{i+1}^{n+1})+\Delta tf_i^n+u_i^n \rightarrow$$

$$-\alpha\frac{\Delta tK}{\Delta x^2}u_{i-1}^{n+1}+(1+2\alpha\frac{\Delta tK}{\Delta x^2}u_i^{n+1})-\alpha\frac{\Delta tK}{\Delta x^2}u_{i+1}^{n+1})=\Delta tf_i^n+u_i^n\,.$$

Let $A_i=C_i=-\alpha\mu$, $B_i=(1+2\alpha\mu)$, and $d_i=\Delta tf_i^n+u_i^n$, where $\mu=\frac{\Delta tK}{\Delta x^2}u_i^{n+1}$, we have

$$A_iu_{i-1}^{n+1}+B_iu_i^{n+1}+C_iu_{i+1}^{n+1}=d_i \qquad (7)$$

for i = 1, 2, …, N-1, assuming the boundaries are at i=0 and N.

If we have Dirichlet boundary conditions, i.e., *u* at i=0 and N are known, then for i=1, the equation becomes

$$B_1u_1^{n+1}+C_1u_2^{n+1}=d_1-A_1u_0^{n+1} \qquad (8)$$

and for $i = N$, the equation is

$$A_{N-1}u_{N-2}^{n+1} + B_{N-1}u_{N-1}^{n+1} = d_{N-1} - C_{N-1}u_N^{n+1}. \qquad (9)$$

For $i$ = 2, 3, …., $N$-2, the equation remains of the form in Eq.(7).

If we write the equations (7-9) in a matrix form, we have

$$\begin{bmatrix} B_1 & C_1 & & & & & \\ A_2 & B_2 & C_2 & & & & \\ & . & . & . & & & \\ & & A_i & B_i & C_i & & \\ & & & . & . & . & \\ & & & & A_{N-2} & B_{N-1} & C_{N-2} \\ & & & & & A_{N-1} & B_{N-1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ . \\ u_i \\ . \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ . \\ D_i \\ . \\ D_{N-2} \\ D_{N-1} \end{bmatrix} \qquad (10)$$

where

$D_1 = \Delta t\ f_1^n - u_1^n - A_0 u_0^{n+1}$,
$D_{N-1} = \Delta t\ f_{N-1}^n - u_{N-1}^n - C_N u_N^{n+1}$,
$D_i = \Delta t f^n - u_i^n$ for i=2, …, N-2.

If we have Neumann boundary conditions, i.e., we know the gradient of u at the boundaries which in discretized form are $u_1 - u_0 = L$ and $u_N - u_{N-1} = R$. Plug these relations into Eq.(7) for *i*=1 and *i*=*N*-1, we obtain equations similar to (8) and (9):

$$(A_1 + B_1)u_1^{n+1} + C_1 u_2^{n+1} = d_1 + A_1 L \qquad (11)$$

$$A_{N-1}u_{N-2}^{n+1} + (B_{N-1} + C_{N-1})u_{N-1}^{n+1} = d_{N-1} - C_{N-1}R. \qquad (12)$$

In this case, the final coefficients in (10) are different for the first and last equation.

Since in each except for the first and last row of the coefficient matrix, only three elements are non-zero and the non-zero elements of the matrix are aligned along the diagonal axis, this system is called tridiagonal system of equations. It can be solved efficiently using Thomas Algorithm.

The procedure consists of two parts. First, Eq.(7) is manipulated into the following form:

$$\begin{bmatrix} 1 & C'_1 & & & & & \\ & 1 & C'_2 & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & & 1 & C'_i & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & & 1 & C'_{N-2} \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ u_i \\ \cdot \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} D'_1 \\ D'_2 \\ \cdot \\ D'_i \\ \cdot \\ D'_{N-2} \\ D'_{N-1} \end{bmatrix} \qquad (13)$$

in which the subdiagonal coefficients A are eliminated and the diagonal coefficients are normalized. For the first equation

$$C'_1 = \frac{C_1}{B_1}, \qquad D'_1 = \frac{D_1}{B_1}. \tag{14a}$$

For the general equations:

$$C'_i = \frac{C_i}{B_i - A_i C'_{i-1}}, \qquad D'_i = \frac{D_i - A_i D'_{i-1}}{B_i - A_i C'_{i-1}}. \tag{14b}$$

Equations in (14) represent a forward sweep step (see figure below). It is followed by a backward substitution step that finds solution $u_i$ from (13). The solution is:

$$\begin{aligned} u_{N-1} &= D'_{N-1} \\ u_i &= D'_i - u_{i+1} C'_i \quad \text{for } i \text{ from } N-2 \text{ to } 1. \end{aligned} \tag{15}$$

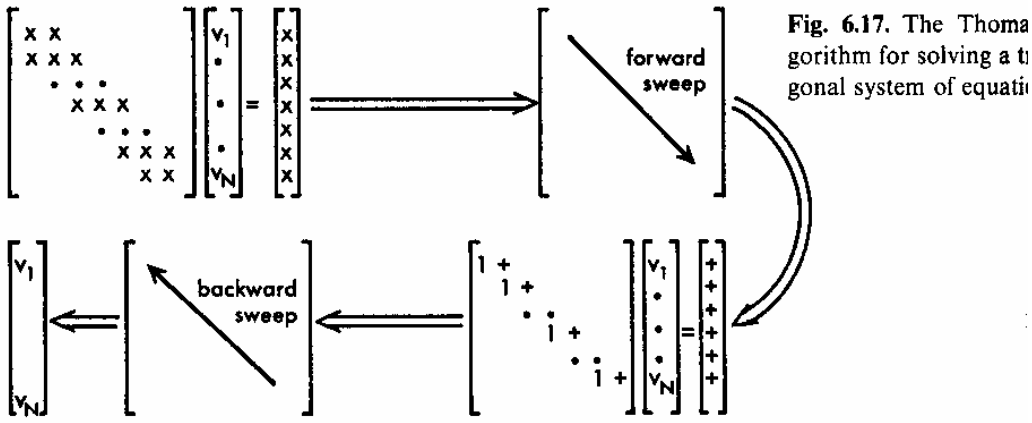


**Fig. 6.17.** The Thoma
gorithm for solving a t
gonal system of equati

Note both (14) and (15) involve reduction, the algorithm is inherently non-parallelizable. Fortunately, for multi-dimensional problems, multiple systems of equations often need to be solved, and one can exploit parallelism along other dimensions (e.g., *j* instead of *i* direction).

Read Section 4.3.3 of Tannehill et al and Appendix A.

### 2.3.7. Stability Analysis for Systems of Equations

When we are dealing with a system of equations, we can also apply the von Neumann method to find the stability property of a given F.D. scheme.

As with single equations, von Neumann can only be used for linear systems of equations. For nonlinear systems, linearization has to be performed first.

Without going into details, we point out that a system of linear equations can be expressed in a matrix form like

$$\frac{\partial \vec{u}}{\partial t} + [A]\frac{\partial \vec{u}}{\partial x} = 0 \qquad (16)$$

The equation is first discretized using certain F.D. scheme, $\vec{u}$ can be written in terms of a discrete Fourier series and the wave component is then substituted into the discrete equation to obtain something like:

$$\vec{U}_k^{n+1} = [M(\Delta t, \Delta x)]\vec{U}_k^n \qquad (17)$$

where $\vec{U}_k^n$ is the amplitude vector for wave *k* at time level *n*, and [M] is called the amplification matrix.

The scheme is stable when the maximum absolute eigenvalue of [M] is no greater than 1.

Why the maximum absolute eigenvalue?

Because as you saw earlier (in Chapter 1) that a system of equation like (17) can be transformed into a system of decoupled equations, and the eigenvalues of [M] become the amplification factors for each of the new dependent variables, $v_i$ (the element of vector $\vec{V}$), i.e., we can obtain from (17)

$$\vec{V}_k^{n+1} = [N]\,\vec{V}_k^{n},$$

where [N] is a diagonal matrix with the eigenvalues of [M] as its diagonal elements.

{[T]$^{-1}$ [M] [T] = [N] → $[T]^{-1}\vec{U}_k^{n+1} = [T]^{-1}[M][T][T]^{-1}\vec{U}_k^{n}$ → $\vec{V}_k^{n+1} = [N]\,\vec{V}_k^{n}$ where $\vec{V}_k^{n+1} = [N]^{-1}\vec{U}_k^{n}$. Therefore $(v_i)_k^{n+1} = \lambda_i (v_i)_k^{n}$ where $v_i$ is an element of vector $\vec{V}$.}

Since the system is stable only when all dependent variables remain bounded, the absolute value of the maximum eigenvalue has to be no greater than 1.

Read section 3.6.2 of Tannehil et al.

Appendix

Shorthand notations for discrete/finite difference operators and discretization identities

Notations:

$$\overline{A}^{nx} = \frac{A_{j+nx} + A_{j-nx}}{2}$$

$$\delta_{nx} x = \frac{A_{j+n/2} - A_{j-n/2}}{n\Delta x}$$

$$\delta_{+x} x = \frac{A_{j+1} - A_j}{\Delta x}$$

$$\delta_{-x} x = \frac{A_j - A_{j-1}}{\Delta x}$$

Identities:

$$\delta_{2x} A \equiv \delta_x \overline{A}^x \equiv \overline{\delta_x A}^x$$

$$A\delta_x B \equiv \delta_x (\overline{A}^x B) - \overline{B\delta_x A}^x$$

$$\overline{A}^x \delta_x A \equiv \delta_x (A^2/2)$$