Chapter 8

CHOICE OF TIME-MARCHING METHODS

In this chapter we discuss considerations involved in selecting a time-marching method for a specific application. Examples are given showing how time-marching methods can be compared in a given context. An important concept underlying much of this discussion is *stiffness*, which is defined in the next section.

8.1 Stiffness Definition for ODE's

8.1.1 Relation to λ-Eigenvalues

The introduction of the concept referred to as "stiffness" comes about from the numerical analysis of mathematical models constructed to simulate dynamic phenomena containing widely different time scales. Definitions given in the literature are not unique, but fortunately we now have the background material to construct a definition which is entirely sufficient for our purposes.

We start with the assumption that our CFD problem is modeled with sufficient accuracy by a coupled set of ODE's producing an A matrix typified by Eq. 7.1. Any definition of stiffness requires a *coupled* system with at least two eigenvalues, and the decision to use some numerical time-marching or iterative method to solve it. The difference between the dynamic scales in physical space is represented by the difference in the magnitude of the eigenvalues in eigenspace. In the following discussion we concentrate on the transient part of the solution. The forcing function may also be time varying in which case it would also have a time scale. However, we assume that this scale would be adequately resolved by the chosen time-marching method, and, since this part of the ODE has no effect on the numerical stability of

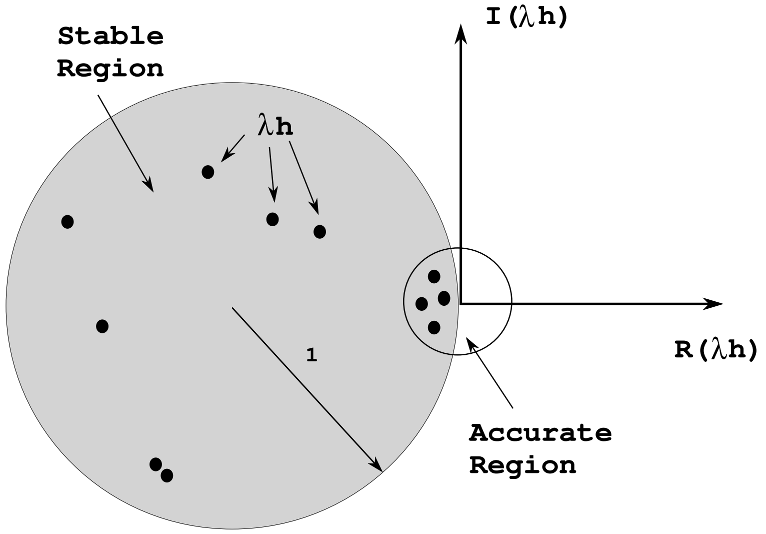

Figure 8.1: Stable and accurate regions for the explicit Euler method.

the homogeneous part, we exclude the forcing function from further discussion in this section.

Consider now the form of the exact solution of a system of ODE's with a complete eigensystem. This is given by Eq. 6.27 and its solution using a one-root, time-marching method is represented by Eq. 6.28. For a given time step, the time integration is an approximation in eigenspace that is different for every eigenvector $\vec{x}_m$. In many numerical applications the eigenvectors associated with the small $|\lambda_m|$ are well resolved and those associated with the large $|\lambda_m|$ are resolved much less accurately, if at all. The situation is represented in the complex λh plane in Fig. 8.1. In this figure the time step has been chosen so that time accuracy is given to the eigenvectors associated with the eigenvalues lying in the small circle and stability without time accuracy is given to those associated with the eigenvalues lying outside of the small circle but still inside the large circle.

The whole concept of stiffness in CFD arises from the fact that we often do not need the *time resolution* of eigenvectors associated with the large $|\lambda_m|$ in the transient solution, although these eigenvectors must remain coupled into the system to maintain a high accuracy of the *spatial resolution*.

8.1.2 Driving and Parasitic Eigenvalues

For the above reason it is convenient to subdivide the transient solution into two parts. First we order the eigenvalues by their magnitudes, thus

$$|\lambda_1| \le |\lambda_2| \le \cdots \le |\lambda_M| \tag{8.1}$$

Then we write

$$\begin{matrix}\text{Transient}\\ \text{Solution}\end{matrix} = \underbrace{\sum_{m=1}^{p} c_m e^{\lambda_m t}\, \vec{x}_m}_{\text{Driving}} + \underbrace{\sum_{m=p+1}^{M} c_m e^{\lambda_m t}\, \vec{x}_m}_{\text{Parasitic}} \tag{8.2}$$

This concept is crucial to our discussion. Rephrased, it states that we can separate our eigenvalue spectrum into two groups; one $[\lambda_1 \to \lambda_p]$ called the driving eigenvalues (our choice of a time-step and marching method must accurately approximate the time variation of the eigenvectors associated with these), and the other, $[\lambda_{p+1} \to \lambda_M]$, called the parasitic eigenvalues (no time accuracy whatsoever is required for the eigenvectors associated with these, but their presence must not contaminate the accuracy of the complete solution). Unfortunately, we find that, although time accuracy requirements are dictated by the driving eigenvalues, numerical stability requirements are dictated by the parasitic ones.

8.1.3 Stiffness Classifications

The following definitions are somewhat useful. An inherently stable set of ODE's is stiff if

$$|\lambda_p| \ll |\lambda_M|$$

In particular we define the ratio

$$C_r = |\lambda_M| \,/\, |\lambda_p|$$

and form the categories

$$\begin{array}{cccc} \text{Mildly-stiff} & & C_r & < 10^2 \\ \text{Strongly-stiff} & 10^3 < & C_r & < 10^5 \\ \text{Extremely-stiff} & 10^6 < & C_r & < 10^8 \\ \text{Pathologically-stiff} & 10^9 < & C_r & \end{array}$$

It should be mentioned that the gaps in the stiff category definitions are intentional because the bounds are arbitrary. It is important to notice that these definitions make no distinction between real, complex, and imaginary eigenvalues.

8.2 Relation of Stiffness to Space Mesh Size

Many flow fields are characterized by a few regions having high spatial gradients of the dependent variables and other domains having relatively low gradient phenomena. As a result it is quite common to cluster mesh points in certain regions of space and spread them out otherwise. Examples of where this clustering might occur are at a shock wave, near an airfoil leading or trailing edge, and in a boundary layer.

One quickly finds that this grid clustering can strongly affect the eigensystem of the resulting A matrix. In order to demonstrate this, let us examine the eigensystems of the model problems given in Section 4.3.2. The simplest example to discuss relates to the model diffusion equation. In this case the eigenvalues are all real, negative numbers that automatically obey the ordering given in Eq. 8.1. Consider the case when *all* of the eigenvalues are parasitic, i.e., we are interested only in the converged steady-state solution. Under these conditions, the stiffness is determined by the ratio λ_M/λ_1. A simple calculation shows that

$$\lambda_1 = -\frac{4\nu}{\Delta x^2}\,\sin^2\left(\frac{\pi}{2(M+1)}\right) \approx -\left(\frac{4\nu}{\Delta x^2}\right)\left(\frac{\Delta x}{2}\right)^2 = -\nu$$

$$\lambda_M \approx -\frac{4\nu}{\Delta x^2}\,\sin^2\left(\frac{\pi}{2}\right) = -\frac{4\nu}{\Delta x^2}$$

and the ratio is

$$\lambda_M/\lambda_1 \approx \frac{4}{\Delta x^2} = 4\left(\frac{M+1}{\pi}\right)^2$$

The most important information found from this example is the fact that the stiffness of the transient solution is directly related to the grid spacing. Furthermore, in diffusion problems this stiffness is proportional to the reciprocal of the space mesh size *squared*. For a mesh size $M = 40$, this ratio is about 680. Even for a mesh of this moderate size the problem is already approaching the category of strongly stiff.

For the biconvection model a similar analysis shows that

$$|\lambda_M|\,/\,|\lambda_1| \approx \frac{1}{\Delta x}$$

Here the stiffness parameter is still space-mesh dependent, but much less so than for diffusion-dominated problems.

We see that in both cases we are faced with the rather annoying fact that the more we try to increase the resolution of our spatial gradients, the stiffer our equations tend to become. Typical CFD problems without chemistry vary between the mildly and strongly stiff categories, and are *greatly* affected by the resolution of a boundary layer since it is a diffusion process. Our brief analysis has been limited to equispaced problems, but in general the stiffness of CFD problems is proportional to the mesh intervals in the manner shown above where *the critical interval is the smallest one in the physical domain.*

8.3 Practical Considerations for Comparing Methods

We have presented relatively simple and reliable measures of stability and both the local and global accuracy of time-marching methods. Since there are an endless number of these methods to choose from, one can wonder how this information is to be used to pick a "best" choice for a particular problem. There is no unique answer to such a question. For example, it is, among other things, highly dependent upon the speed, capacity, and architecture of the available computer, and technology influencing this is undergoing rapid and dramatic changes as this is being written. Nevertheless, if certain ground rules are agreed upon, relevant conclusions can be reached. Let us now examine some ground rules that might be appropriate. It should then be clear how the analysis can be extended to other cases.

Let us consider the problem of measuring the efficiency of a time–marching method for computing, over a fixed interval of time, an accurate transient solution of a coupled set of ODE's. The length of the time interval, T, and the accuracy required of the solution are dictated by the physics of the particular problem involved. For example, in calculating the amount of turbulence in a homogeneous flow, the time interval would be that required to extract a reliable statistical sample, and the accuracy would be related to how much the energy of certain harmonics would be permitted to distort from a given level. Such a computation we refer to as an *event.*

The appropriate error measures to be used in comparing methods for calculating an event are the *global* ones, Er_a, Er_λ and Er_ω, discussed in Section 6.6.5, rather than the local ones er_λ, er_a, and er_p discussed earlier.

The actual form of the coupled ODE's that are produced by the semi-discrete approach is

$$\frac{d\vec{u}}{dt} = \vec{F}(\vec{u}, t)$$

At every time step we must evaluate the function $\vec{F}(\vec{u}, t)$ at least once. This function is usually nonlinear, and its computation *usually consumes the major portion of the computer time required to make the simulation.* We refer to a single calculation of the vector $\vec{F}(\vec{u}, t)$ as a *function evaluation* and denote the total number of such evaluations by F_{ev}.

8.4 Comparing the Efficiency of Explicit Methods

8.4.1 Imposed Constraints

As mentioned above, the efficiency of methods can be compared only if one accepts a set of limiting constraints within which the comparisons are carried out. The follow assumptions bound the considerations made in this Section:

1. The time-march method is explicit.

2. Implications of computer storage capacity and access time are ignored. In some contexts, this can be an important consideration.

3. The calculation is to be time-accurate, must simulate an entire event which takes a total time T, and must use a constant time step size, h, so that

$$T = Nh$$

   where N is the total number of time steps.

8.4.2 An Example Involving Diffusion

Let the event be the numerical solution of

$$\frac{du}{dt} = -u \tag{8.3}$$

from $t = 0$ to $T = -\ln(0.25)$ with $u(0) = 1$. Eq. 8.3 is obtained from our representative ODE with $\lambda = -1$, $a = 0$. Since the exact solution is $u(t) = u(0)e^{-t}$, this makes the exact value of u at the end of the event equal to 0.25, i.e., $u(T) = 0.25$. To the constraints imposed above, let us set the additional requirement

- The error in u at the end of the event, i.e., the *global* error, must be $< 0.5\%$.

We judge the most efficient method as the one that satisfies these conditions and has the fewest number of evaluations, F_{ev}. Three methods are compared — explicit Euler, AB2, and RK4.

First of all, the allowable error constraint means that the global error in the amplitude, see Eq. 6.48, must have the property:

$$\left|\frac{Er_\lambda}{e^{\lambda T}}\right| < 0.005$$

Then, since $h = T/N = -\ln(0.25)/N$, it follows that

$$\left|1 - (\sigma_1(\ln(.25)/N))^N/.25\right| < 0.005$$

where σ_1 is found from the characteristic polynomials given in Table 7.1. The results shown in Table 8.1 were computed using a simple iterative procedure.

Method	N	h	σ_1	F_{ev}	Er_λ	
Euler	193	.00718	.99282	193	.001248	worst
AB2	16	.0866	.9172	16	.001137	
RK4	2	.6931	.5012	8	.001195	best

Table 8.1: Comparison of time-marching methods for a simple dissipation problem.

In this example we see that, for a given global accuracy, the method with the highest local accuracy is the most efficient on the basis of the expense in evaluating F_{ev}. Thus the second-order Adams-Bashforth method is much better than the first-order Euler method, and the fourth-order Runge-Kutta method is the best of all. The main purpose of this exercise is to show the (usually) great superiority of second-order over first-order time-marching methods.

8.4.3 An Example Involving Periodic Convection

Let us use as a basis for this example the study of homogeneous turbulence simulated by the numerical solution of the incompressible Navier-Stokes equations inside a cube with periodic boundary conditions on all sides. In this numerical experiment the function evaluations contribute overwhelmingly to the CPU time, and the number of these evaluations must be kept to an absolute minimum because of the magnitude of the problem. On the other hand, a complete event must be established in order to obtain meaningful statistical samples which are the essence of the solution. In this case, in addition to the constraints given in Section 8.4.1, we add the following:

- The number of evaluations of $\vec{F}(\vec{u}, t)$ is fixed.

Under these conditions a method is judged as best when it has the highest global accuracy for resolving eigenvectors with imaginary eigenvalues. The above constraint has led to the invention of schemes that omit the function evaluation in the corrector step of a predictor-corrector combination, leading to the so-called incomplete predictor-corrector methods. The presumption is, of course, that more efficient methods will result from the omission of the second function evaluation. An example is the method of Gazdag, given in Section 6.8. Basically this is composed of an AB2 predictor and a trapezoidal corrector. However, the derivative of the fundamental family is never found so there is only one evaluation required to complete each cycle. The λ-σ relation for the method is shown as entry 10 in Table 7.1.

In order to discuss our comparisions we introduce the following definitions:

- Let a k-evaluation method be defined as one that requires k evaluations of $\vec{F}(\vec{u}, t)$ to advance one step using that method's time interval, h.
- Let K represent the total number of allowable F_{ev}.
- Let h_1 be the time interval advanced in one step of a one-evaluation method.

The Gazdag, leapfrog, and AB2 schemes are all 1-evaluation methods. The second and fourth order RK methods are 2- and 4-evaluation methods, respectively. For a 1-evaluation method the total number of time steps, N, and the number of evaluations, K, are the same, one evaluation being used for each step, so that for these methods $h = h_1$. For a 2-evaluation method $N = K/2$ since two evaluations are used for each step. However, in this case, in order to arrive at the same time T after K evaluations, the time step must be twice that of a one–evaluation method so $h = 2h_1$. For a 4-evaluation method the time interval must be $h = 4h_1$, etc. Notice that as k increases, the time span required for one application of the method increases. *However, notice also that as k increases, the power to which σ_1 is raised to arrive at the final destination decreases;* see the Figure below. This is the key to the true comparison of time-march methods for this type of problem.

	0								T	u_N
$k=1$	|	•	•	•	•	•	•	•	|	$[\sigma(\lambda h_1)]^8$
$k=2$	|	$2h_1$	•		•		•		|	$[\sigma(2\lambda h_1)]^4$
$k=4$	|		$4h_1$		•				|	$[\sigma(4\lambda h_1)]^2$

Step sizes and powers of σ for k-evaluation methods used to get to the same value of T if 8 evaluations are allowed.

In general, after K evaluations, the global amplitude and phase error for k-evaluation methods applied to systems with pure imaginary λ-roots can be written[1]

$$Er_a = 1 - |\sigma_1(ik\omega h_1)|^{K/k} \tag{8.4}$$

[1]See Eqs. 6.38 and 6.39.

$$Er_\omega = \omega T - \frac{K}{k}\tan^{-1}\left[\frac{[\sigma_1(ik\omega h_1)]_{imaginary}}{[\sigma_1(ik\omega h_1)]_{real}}\right] \tag{8.5}$$

Consider a convection-dominated event for which the function evaluation is very time consuming. We idealize to the case where $\lambda = i\omega$ and set ω equal to one. The event must proceed to the time t = T = 10. We consider two maximum evaluation limits $K = 50$ and $K = 100$ and choose from four possible methods, leapfrog, AB2, Gazdag, and RK4. The first three of these are one-evaluation methods and the last one is a four-evaluation method. It is not difficult to show that on the basis of local error (made in a single step) the Gazdag method is superior to the RK4 method in both amplitude and phase. For example, for $\omega h = 0.2$ the Gazdag method produces a $|\sigma_1| = 0.9992276$ whereas for $\omega h = 0.8$ (which must be used to keep the number of evaluations the same) the RK4 method produces a $|\sigma_1| = 0.998324$. However, we are making our comparisons on the basis of global error for a fixed number of evaluations.

First of all we see that for a one-evaluation method $h_1 = T/K$. Using this, and the fact that $\omega = 1$, we find, by some rather simple calculations[2] made using Eqs. 8.4 and 8.5, the results shown in Table 8.2. Notice that to find global error the Gazdag root must be raised to the power of 50 while the RK4 root is raised only to the power of 50/4. On the basis of *global error* the Gazdag method is not superior to RK4 in either amplitude or phase, although, in terms of phase error (for which it was designed) it is superior to the other two methods shown.

	K	leapfrog	AB2	Gazdag	RK4
$\omega h_1 = .1$	100	1.0	1.003	.995	.999
$\omega h_1 = .2$	50	1.0	1.022	.962	.979

a. Amplitude, exact = 1.0.

	K	leapfrog	AB2	Gazdag	RK4
$\omega h_1 = .1$	100	$-.96$	-2.4	.45	.12
$\omega h_1 = .2$	50	-3.8	-9.8	1.5	1.5

b. Phase error in degrees.

Table 8.2: Comparison of global amplitude and phase errors for four methods.

[2]The σ_1 root for the Gazdag method can be found using a numerical root finding routine to trace the three roots in the σ-plane, see Fig. 7.3e.

Using analysis such as this (and also considering the stability boundaries) the RK4 method is recommended as a basic first choice for any explicit time-accurate calculation of a convection-dominated problem.

8.5 Coping With Stiffness

8.5.1 Explicit Methods

The ability of a numerical method to cope with stiffness can be illustrated quite nicely in the complex λh plane. A good example of the concept is produced by studying the Euler method applied to the representative equation. The transient solution is $u_n = (1 + \lambda h)^n$ and the trace of the complex value of λh which makes $|1 + \lambda h| = 1$ gives the whole story. In this case the trace forms a circle of unit radius centered at $(-1, 0)$ as shown in Fig. 8.1. If h is chosen so that all λh in the ODE eigensystem fall inside this circle the integration will be numerically stable. Also shown by the small circle centered at the origin is the region of Taylor series accuracy. If some λh fall outside the small circle but stay within the stable region, these λh are stiff, but stable. We have defined these λh as parasitic eigenvalues. Stability boundaries for some explicit methods are shown in Figs. 7.5 and 7.6.

For a specific example, consider the mildly stiff system composed of a coupled two-equation set having the two eigenvalues $\lambda_1 = -100$ and $\lambda_2 = -1$. If uncoupled and evaluated in wave space, the time histories of the two solutions would appear as a rapidly decaying function in one case, and a relatively slowly decaying function in the other. Analytical evaluation of the time histories poses no problem since e^{-100t} quickly becomes very small and can be neglected in the expressions when time becomes large. Numerical evaluation is altogether different. Numerical solutions, of course, depend upon $[\sigma(\lambda_m h)]^n$ and no $|\sigma_m|$ can exceed one for any λ_m in the coupled system or else the process is numerically unstable.

Let us choose the simple explicit Euler method for the time march. The coupled equations in real space are represented by

$$\begin{aligned} u_1(n) &= c_1(1 - 100h)^n x_{11} + c_2(1 - h)^n x_{12} + (PS)_1 \\ u_2(n) &= c_1(1 - 100h)^n x_{21} + c_2(1 - h)^n x_{22} + (PS)_2 \end{aligned} \tag{8.6}$$

We will assume that our accuracy requirements are such that sufficient accuracy is obtained as long as $|\lambda h| \leq 0.1$. This defines a time step limit based on *accuracy* considerations of $h = 0.001$ for λ_1 and $h = 0.1$ for λ_2. The time step limit based on *stability*, which is determined from λ_1, is $h = 0.02$. We will also assume that $c_1 = c_2 = 1$ and that an amplitude less than 0.001 is negligible. We first run 66 time steps with $h = 0.001$ in order to resolve the λ_1 term. With this time step the

λ_2 term is resolved exceedingly well. After 66 steps, the amplitude of the λ_1 term (i.e., $(1-100h)^n$) is less than 0.001 and that of the λ_2 term (i.e., $(1-h)^n$) is 0.9361. Hence the λ_1 term can now be considered negligible. To drive the $(1-h)^n$ term to zero (i.e., below 0.001), we would like to change the step size to $h = 0.1$ and continue. We would then have a well resolved answer to the problem throughout the entire relevant time interval. However, this is not possible because of the coupled presence of $(1-100h)^n$, which in just 10 steps at $h = 0.1$ amplifies those terms by $\approx 10^9$, far outweighing the initial decrease obtained with the smaller time step. In fact, with $h = 0.02$, the maximum step size that can be taken in order to maintain stability, about 339 time steps have to be computed in order to drive e^{-t} to below 0.001. Thus the total simulation requires 405 time steps.

8.5.2 Implicit Methods

Now let us re-examine the problem that produced Eq. 8.6 but this time using an unconditionally stable implicit method for the time march. We choose the trapezoidal method. Its behavior in the λh plane is shown in Fig. 7.4b. Since this is also a one-root method, we simply replace the Euler σ with the trapezoidal one and analyze the result. It follows that the final numerical solution to the ODE is now represented in real space by

$$\begin{aligned} u_1(n) &= c_1\left(\frac{1-50h}{1+50h}\right)^n x_{11} + c_2\left(\frac{1-0.5h}{1+0.5h}\right)^n x_{12} + (PS)_1 \\ u_2(n) &= c_1\left(\frac{1-50h}{1+50h}\right)^n x_{21} + c_2\left(\frac{1-0.5h}{1+0.5h}\right)^n x_{22} + (PS)_2 \end{aligned} \tag{8.7}$$

In order to resolve the initial transient of the term e^{-100t}, we need to use a step size of about $h = 0.001$. This is the same step size used in applying the explicit Euler method because here accuracy is the only consideration and a very small step size must be chosen to get the desired resolution. (It is true that for the same accuracy we could in this case use a larger step size because this is a second-order method, but that is not the point of this exercise). After 70 time steps the λ_1 term has amplitude less than 0.001 and can be neglected. Now with the implicit method we can proceed to calculate the remaining part of the event using our desired step size $h = 0.1$ without any problem of instability, with 69 steps required to reduce the amplitude of the second term to below 0.001. In both intervals the desired solution is second-order accurate and well resolved. It is true that in the final 69 steps one σ-root is $[1-50(0.1)]/[1+50(0.1)] = 0.666\cdots$, and this has no physical meaning whatsoever. However, its influence on the coupled solution is negligible at the end of the first 70 steps, and, since $(0.666\cdots)^n < 1$, its influence in the remaining 70 steps is even less. Actually, although this root is one of the principal roots in the system, its behavior for $t > 0.07$ is identical to that of a stable spurious root. The total simulation requires 139 time steps.

8.5.3 A Perspective

It is important to retain a proper perspective on a problem represented by the above example. It is clear that an unconditionally stable method can always be called upon to solve stiff problems with a minimum number of time steps. In the example, the conditionally stable Euler method required 405 time steps, as compared to about 139 for the trapezoidal method, about three times as many. However, the Euler method is extremely easy to program and requires very little arithmetic per step. For preliminary investigations it is often the best method to use for mildly-stiff diffusion dominated problems. For refined investigations of such problems an explicit method of second order or higher, such as Adams-Bashforth or Runge-Kutta methods, is recommended. These explicit methods can be considered as effective mildly stiff-stable methods. However, it should be clear that as the degree of stiffness of the problem increases, the advantage begins to tilt towards implicit methods, as the reduced number of time steps begins to outweigh the increased cost per time step. The reader can repeat the above example with $\lambda_1 = -10,000$, $\lambda_2 = -1$, which is in the strongly-stiff category.

There is yet another technique for coping with certain stiff systems in fluid dynamic applications. This is known as the *multigrid* method. It has enjoyed remarkable success in many practical problems; however, we need an introduction to the theory of relaxation before it can be presented.

8.6 Steady Problems

In Chapter 6 we wrote the OΔE solution in terms of the principal and spurious roots as follows:

$$\begin{aligned}
u_n = \; & c_{11}(\sigma_1)_1^n\,\vec{x}_1 + \cdots + c_{m1}(\sigma_m)_1^n\,\vec{x}_m + \cdots + c_{M1}(\sigma_M)_1^n\,\vec{x}_M + P.S. \\
& + c_{12}(\sigma_1)_2^n\,\vec{x}_1 + \cdots + c_{m2}(\sigma_m)_2^n\,\vec{x}_m + \cdots + c_{M2}(\sigma_M)_2^n\,\vec{x}_M \\
& + c_{13}(\sigma_1)_3^n\,\vec{x}_1 + \cdots + c_{m3}(\sigma_m)_3^n\,\vec{x}_m + \cdots + c_{M3}(\sigma_M)_3^n\,\vec{x}_M \\
& + \text{etc., if there are more spurious roots}
\end{aligned} \tag{8.8}$$

When solving a *steady* problem, we have no interest whatsoever in the transient portion of the solution. Our sole goal is to eliminate it as quickly as possible. Therefore, the choice of a time-marching method for a steady problem is similar to that for a stiff problem, the difference being that the order of accuracy is irrelevant. Hence the explicit Euler method is a candidate for steady diffusion dominated problems, and the fourth-order Runge-Kutta method is a candidate for steady convection dominated problems, because of their stability properties. Among implicit methods, the implicit Euler method is the obvious choice for steady problems.

When we seek only the steady solution, all of the eigenvalues can be considered to be parasitic. Referring to Fig. 8.1, none of the eigenvalues are required to fall in the accurate region of the time-marching method. Therefore the time step can be chosen to eliminate the transient as quickly as possible with no regard for time accuracy. For example, when using the implicit Euler method with local time linearization, Eq. 6.96, one would like to take the limit $h \to \infty$, which leads to Newton's method, Eq. 6.98. However, a finite time step may be required until the solution is somewhat close to the steady solution.

8.7 Problems

1. Repeat the time-march comparisons for diffusion (Section 8.4.2) and periodic convection (Section 8.4.3) using 2nd- and 3rd-order Runge-Kutta methods.

2. Repeat the time-march comparisons for diffusion (Section 8.4.2) and periodic convection (Section 8.4.3) using the 3rd- and 4th-order Adams-Bashforth methods. Considering the stability bounds for these methods (see problem 4 in Chapter 7) as well as their memory requirements, compare and contrast them with the 3rd- and 4th-order Runge-Kutta methods.

3. Consider the diffusion equation (with $\nu = 1$) discretized using 2nd-order central differences on a grid with 10 (interior) points. Find and plot the eigenvalues and the corresponding modified wavenumbers. If we use the explicit Euler time-marching method what is the maximum allowable time step if all but the first two eigenvectors are considered parasitic? Assume that sufficient accuracy is obtained as long as $|\lambda h| \leq 0.1$. What is the maximum allowable time step if all but the first eigenvector are considered parasitic?