11

Numerical techniques for radiation transport

In this chapter we give brief discussions of the main solution algorithms for radiation hydrodynamics problems, some of which are very quick and approximate and some of which represent the best attempts at accuracy. It is unfortunately true that "you get what you pay for" in these calculations, and accuracy comes at a considerable cost. The earlier material in this book has brought out quite a few different processes that complicate the endeavor, such as the effects of fluid velocity on radiation quantities, the complicated spectral dependence of the opacity, non-LTE, refraction, and polarization. These effects are not too hard to include singly, although with some effort, but accounting for all of them has not seemed to be a practical objective up to the present time. And of course these difficulties are compounded many-fold in higher-dimensional geometries. Our discussion of algorithms will begin with the low-budget methods that may be priced just right for many purposes, after some preliminary observations about solution strategy. Some general references on this subject are the following: The conference volume *Astrophysical Radiation Hydrodynamics* (Winkler and Norman, 1982) is a good place to start. A meeting that included presentations on many of the current advanced hydrodynamics methods was the 12th Kingston Meeting on Theoretical Astrophysics held in Halifax in 1996 (Norman, 1996). Starting points for surveying the advanced numerical methods in radiation transport are the pair of books by Kalkofen (1984,1987) and the workshop proceedings *Stellar Atmospheres: Beyond Classical Models* (Crivellari, Hubeny, and Hummer, 1991). The most comprehensive review of the astrophysical methods to date is provided by the 2002 Tübingen workshop *Stellar Atmosphere Modeling* (Hubeny, Mihalas, and Werner, 2003).

11.1 Splitting hydrodynamics and radiation

Operator splitting is a time-honored method for calculating initial value problems that consist of different kinds of physics, of which at least some must be treated in

an implicit fashion. Early descriptions of the application of this idea to radiation hydrodynamics are found in the cepheid and RR Lyrae stellar pulsation calculations by Christy (1966) and Cox *et al.* (1966). Another radiation hydrodynamic calculation from around the same time is the supernova model of Colgate and White (1966).

The idea is simple: advance physical process (A) as if it were the only activity during the time step, then use that result as the starting point and advance physics (B) for the same time step as if it were the only activity, and so on through all the processes. To fix some of these ideas, suppose we represent all the variables in our problem, represented in some discrete way in space, as a vector $\mathbf{X}$, and suppose it satisfies a system of equations we write as

$$\frac{d\mathbf{X}}{dt} = \mathbf{A}[\mathbf{X}] + \mathbf{B}[\mathbf{X}] + \cdots, \tag{11.1}$$

where $\mathbf{A}$ and $\mathbf{B}$ are operators that perform different kinds of physics. We discretize time using the set $t^1, t^2, \ldots, t^n, \ldots$. The *explicit* method of time differencing is approximately this:

$$\frac{\mathbf{X}^{n+1} - \mathbf{X}^n}{\Delta t} = \mathbf{A}[\mathbf{X}^n] + \mathbf{B}[\mathbf{X}^n] + \cdots. \tag{11.2}$$

When we write $\mathbf{A}[\mathbf{X}^n]$ we mean that no information *later* in time than t^n is included. The information from several prior time steps may be combined to provided an estimate of the forward difference that is of higher order than the first in Δt. Including all the physical processes we want to is no problem in this approach since we need only keep track of the time derivative contributions from all the processes, calculated by taking spatial derivatives and doing the integrations, then add these up at the end to get the total amount by which to advance $\mathbf{X}$. The failure mode of this approach is that it is almost always subject to a time-step constraint imposed either by the condition for numerical stability, or by accuracy considerations. This takes the form

$$\Delta t \left\| \frac{\delta \mathbf{A}}{\delta \mathbf{X}} \right\| < c, \tag{11.3}$$

where c is some numerical value rather smaller than 1, and likewise for the other operators.

In terms of the particular physical processes we need to consider, the stability limits that arise from this reasoning are the Courant limit, $c_s \Delta t/\Delta x < 1$, from the hydrodynamics, and a radiation Courant limit $c\Delta t/\Delta x < 1$ if we were so brave as to do radiation transport with explicit time differencing. If we use the radiation diffusion approximation, then the stability limit is $K_R \Delta t/(\rho C_v (\Delta x)^2) < 1$. We may or may not be able to live with the Courant limit; it depends on how long

we want to evolve the problem compared with the hydrodynamic time scale. The radiation limit is usually the one that hurts. The radiation diffusion limit can be factored in this way:

$$\frac{K_R \Delta t}{\rho C_v (\Delta x)^2} = \frac{16\sigma T^4}{3\rho C_v T V} \frac{1}{\kappa_R \rho \Delta x} \frac{V \Delta t}{\Delta x}, \tag{11.4}$$

where we have introduced a typical flow speed V. The first factor on the right-hand side is, apart from the factor 16/3, the inverse of the *Boltzmann number* for the flow. This factor can easily be of order 100. The second factor is the reciprocal of the optical depth of a zone. This optical depth certainly becomes as small as unity. The third factor is the time step compared with the flow time across a zone; we would hope that this would be about unity. So the inverse Boltzmann number makes this stability criterion too large by a factor that may be 100. Therefore explicit radiation diffusion is not a good idea. We reach a similar conclusion using the explicit radiation transport equation. In a non-LTE problem we face the stiffness of the kinetics equations, on which we have commented earlier.

There is a negative aspect of implicit time differencing. The rather large truncation error associated with using a time step large compared with the natural time scales based on $\|\delta \mathbf{A}/\delta \mathbf{X}\|$ makes the numerical representation quite dissipative, in the sense that noise is filtered out. This numerical dissipation can suppress real instabilities in the problem, and eliminate real high-frequency components that are physically significant. The validity checks that are applied as the problem runs smoothly along with giant time steps may fail to reveal that high-frequency modes would develop if they were allowed to. Of course, suppressing the high-frequency modes is precisely why one wants to use the implicit method in the first place, but there may not then be a way to verify that the significant results are being obtained correctly.

We turn then to implicit differencing. We try this:

$$\frac{\mathbf{X}^{n+1} - \mathbf{X}^n}{\Delta t} = \mathbf{A}[\mathbf{X}^{n+1}] + \mathbf{B}[\mathbf{X}^{n+1}] + \cdots . \tag{11.5}$$

Now we face a major computational challenge. The values we want to find, $\mathbf{X}^{n+1}$, appear in the (nonlinear) functions on the right-hand side, and to make matters worse, they appear in every term. Notwithstanding the obstacles, this approach has been used very successfully in a number of astrophysical problems such as stellar pulsation and protostar collapse. The attack on the problem is direct: set up all the equations as a nonlinear system for the unknowns $\mathbf{X}^{n+1}$ and apply the multivariate Newton–Raphson method. The initial guess for $\mathbf{X}^{n+1}$ might be taken to be $\mathbf{X}^n$. At each step the Jacobian matrix elements are calculated, the largest part of the cost, and the linear system for the next set of corrections to $\mathbf{X}^{n+1}$ is solved by

direct elimination. The notable successes of this method have been in 1-D spherical geometry. Having only one spatial dimension is kind to direct elimination as a method of solving a banded linear system: the cost scales linearly with the number of zones and as the cube of the matrix bandwidth, i.e., of the number of variables per zone. Direct elimination is much more costly in two dimensions, and we begin to look for different solution methods. There are also problems with ensuring Newton–Raphson convergence; it may be necessary to severely restrict the time step to ensure rapid convergence.

Here then is operator splitting. If the time derivative of $\mathbf{X}$ is split into k pieces, then there are k partial time steps to advance from time t^n to time t^{n+1}:

$$\begin{aligned}
\frac{\mathbf{X}^{n+1/k} - \mathbf{X}^n}{\Delta t} &= \mathbf{A}[\mathbf{X}^{n+1/k}], \\
\frac{\mathbf{X}^{n+2/k} - \mathbf{X}^{n+1/k}}{\Delta t} &= \mathbf{B}[\mathbf{X}^{n+2/k}], \\
&\vdots \\
\frac{\mathbf{X}^{n+1} - \mathbf{X}^{n+(k-1)/k}}{\Delta t} &= \mathbf{F}[\mathbf{X}^{n+1}].
\end{aligned} \tag{11.6}$$

This does in fact converge to a solution of the differential equation as $\Delta t \to 0$, as we can see by adding the equations and Taylor-expanding the right-hand sides about $\mathbf{X}^n$. However, it is only first-order accurate. The order of accuracy is improved, when there are just two operators $\mathbf{A}$ and $\mathbf{B}$, by alternating cycles on which $\mathbf{A}$ is done first, then $\mathbf{B}$, with cycles that do the operators the other way around (called *Strang splitting*). With more than two operators the practice is to do $ABC\ldots$ on one cycle and $\ldots CBA$ on the next.

What are the advantages? Some parts of the physics may not be stiff at all, and those operators may be advanced using an explicit equation, leaving the implicit differencing for the parts that *are* stiff. When an implicit equation has to be solved for one variable in the splitting method the bandwidth of the linear system is much reduced. It is nine times faster to solve three linear systems with bandwidth one than to solve one linear system with bandwidth three. If you do not really need to solve two of the three systems in the split case, the gain is a factor 27. The lower dimensionality of the nonlinear system helps greatly with the robustness of the Newton–Raphson convergence. The disadvantage is that the error of the time differencing is increased, and it may not be very easy to estimate. Strang splitting helps with this, but there can still be the problem that $\mathbf{A}$ might move the solution in the wrong direction, creating an error that $\mathbf{B}$ has to correct. This problem is not usually *too* severe, but it must be watched for.

The stellar pulsation calculations for RR Lyrae stars by Christy (1966) and for cepheids by Cox *et al*. (1966) were made up of spherical Lagrangian zones with just three unknowns per zone: the radius, velocity, and temperature. The calculations proceeded in a staggered way with time, with the velocity being updated first, a half-time-step later the radii were updated, and finally the temperatures. Only the temperature equation was implicit. The Newton–Raphson method applied to the material energy equation led to a tri-diagonal system for the temperature corrections, which is about as easy as linear systems get.

With two or more space dimensions the choices become more painful. The considerations about time step limits still apply, so perhaps the hydrodynamics can be done explicitly, although some of the modern methods (e.g. Godunov's method, see Section 3.2.2) may still use splitting as a convenience. The radiation equations are a major problem now. The radiation diffusion equation has the character of an elliptic equation in space after the time differencing is done, and this is not at all as easy to solve as a scalar two-point boundary-value problem in one dimension. So even if the operator splitting is applied and the radiation equation is treated separately, the solution requires an iterative linear system solver such as the conjugate gradient (CG) or alternating-direction implicit (ADI) method. It will also be seen below that the adequacy of diffusion as a substitute for properly angle-dependent radiative transfer is more questionable in two and three dimensions than in one.

The coupling of the material temperature to the radiation field, through the material internal energy equation, has the helpful property that it involves only local quantities, apart from the advection term (which is lumped with the hydrodynamics processes in the splitting method), unless thermal conductivity must be considered. Often the conduction flux is negligible, so on the temperature coupling step the material temperature, or at least the Newton–Raphson correction to it, can be eliminated using a local equation so only the radiation field remains to be found from a large system of equations. This reduces the dimensionality by a factor 2, which is important when the solution cost varies as the cube of the number of unknowns per zone.

11.2 Thermal diffusion

Now we begin to walk through a hierarchy of increasing sophisticated and more costly, but not necessarily more accurate, algorithms for solving radiation hydrodynamics problems. We begin with the method Christy, Cox, Colgate, and others used, thermal diffusion, also called equilibrium diffusion. In this method the radiation field is removed from the problem and replaced using the relations derived for the diffusion limit, (6.66), (6.72), and (6.78), although the second order corrections in $E^{(0)}$ and $P^{(0)}$ are usually ignored, along with the relativistic corrections

to $\mathbf{F}^{(0)}$. The combined energy equation for matter and radiation is used, which in effect adds aT^4/ρ to the internal energy, $aT^4/3$ to the pressure, and includes $\mathbf{F}^{(0)}$ as a flux. The advection parts of this having already been treated, what is left is an implicit equation for the temperatures at the advanced time step. The key part of making this equation implicit is using the advanced-time temperatures in the flux. That is, the equation looks something like this after discarding the advection flux and the work term:

$$\frac{\rho^{n+1}e^{n+1} + E^{n+1} - \rho^n e^n - E^n}{\Delta t} - \nabla \cdot \left[K_R(\rho^{n+1}, T^{n+1}) \nabla T^{n+1} \right] = 0. \quad (11.7)$$

We repeat that this equation is not complete since the unnecessary terms for the present discussion have been dropped. The flux term in this equation is certainly not centered in time as it should be to make it second order accurate. That would be true if the flux in square brackets were replaced by the arithmetic average of the values at t^n and t^{n+1}, called Crank–Nicholson differencing. But that form, in the limit $K_R \Delta t/(\rho C_v (\Delta x)^2) \gg 1$, is susceptible to nonlinear numerical instabilities. Then we might try a weighted average, with a somewhat larger weight applied to the t^{n+1} flux. That does seem to solve the instability problem, but the solution remains noisier than if the fully backward-differenced form is used, as given first. This is another example of deliberately choosing the more dissipative numerical representation as a trade-off to obtain the highest possible time step.

Equation (11.7) is solved, as discussed earlier, using the Newton–Raphson method. The spatial derivative operators are first represented in whatever second order accurate form is permitted by the nature of the spatial zoning. In Eulerian calculations the ordinary centered second derivative formula can be used. There is some question about the proper spatial centering of the K_R factor, and different choices may be made. The Jacobian matrix that emerges when the equations are linearized is the matrix of the system that is to be solved. If the opacity-variation parts of the linearization could be ignored, then the system could be arranged to be symmetric and positive-definite, a very great advantage for the application of iterative solvers. Sometimes it is proposed to lag the opacities in time for just that reason. Good results have also been obtained including the opacity terms using nonsymmetric solvers, such as the direct elimination method in one dimension.

The failure mode of the thermal diffusion approximation is its poor performance in optically thin regions. Even the RR Lyrae pulsation calculations of 1965 revealed the shortcomings of the method, because the temperature throughout the atmosphere of the star was spuriously forced to a constant value by the use of a diffusion approximation in an optically thin region. In reality the temperature becomes decoupled from the radiation field, as discussed in Section 7.2.

11.3 Eddington approximation

The major objection just mentioned to thermal diffusion is removed if the assumptions (6.66), (6.72), and (6.78) are replaced with the simple closure relation $P = E/3$ and E and $\mathbf{F}$ are retained as variables. The equations determining them are (6.51) and (6.52), although the $1/c$ terms are usually dropped in the latter. The advection and work terms are also often dropped from the energy equation, but, as discussed earlier, this commits the errors of ignoring radiation energy density and work in the overall energy budget. The frequency integral of the opacity multiplied by the flux has to be approximated, since the spectral distribution of $\mathbf{F}_\nu$ is unknown; guided by thermal diffusion the Rosseland mean is used, leading to

$$\mathbf{F} = -\frac{c}{3\kappa_R \rho}\nabla E. \tag{11.8}$$

The energy coupling term on the right-hand side of (4.29) is approximated as in (8.89), or perhaps with the Rosseland mean here too.[1] The final result for the combined moment equations is this:

$$\rho\frac{D(E/\rho)}{Dt} + \frac{E}{3}\nabla\cdot\mathbf{u} - \nabla\cdot\left(\frac{c}{3\kappa_R\rho}\nabla E\right) = \kappa_P\rho c(aT^4 - E). \tag{11.9}$$

We need to stress at this point that the radiation quantities here are those in the comoving frame, even though the superscripts $^{(0)}$ have been dropped to reduce the clutter in the equations. Only by using comoving radiation are we permitted to evaluate the opacity and emissivity *sans* velocity effects.

We describe now how the temperature update proceeds when the (gray) Eddington approximation as just described is used. We repeat the internal energy equation given earlier,

$$\frac{\partial\rho e}{\partial t} + \nabla\cdot(\rho\mathbf{u}e) + p\nabla\cdot\mathbf{u} = -\kappa_P\rho c(aT^4 - E). \tag{11.10}$$

With operator-split hydrodynamics the advection and work terms in (11.10) have already been evaluated at the point the temperature update is being done. Everything else in this equation is local, so when the equation is linearized for the Newton–Raphson procedure there is just a simple linear equation to solve to obtain the correction to T in terms of that for E. Then this can be substituted into the linearized form of (11.9), which remains an elliptic equation for the corrections to E, at least provided the terms arising from the variation of the opacity with T are not too large. After the substitution of δT in terms of δE the structure of (11.9) is

[1] In the diffusion limit both E_ν and B_ν have the same spectral distribution, while the spectral dependence of $cE_\nu - 4\pi B_\nu$ is proportional to $(dB_\nu/dT)/\kappa_\nu$, which provides some justification for using the Rosseland mean in this place.

identical to the thermal diffusion equation apart from certain differences in the coefficients that correct the small-optical-depth errors in thermal diffusion. The cost to solve the elliptic equation is unchanged.

The significant technology issues connected with both thermal diffusion and Eddington-approximation calculations are: (1) making a finite-difference or finite-element representation of the partial differential equation, and (2) solving the resulting large sparse linear system of equations. Rapid progress has occurred in both areas and we will discuss this in Section 11.4. The computational techniques needed for radiation diffusion are not materially different from those applied to other engineering problems involving elliptic operators, such as heat conduction, electrostatics, and viscous incompressible fluid flow.

The boundary conditions required for (11.9), and also (11.7), come from reasoning similar to that leading to (5.33). We will repeat the argument in somewhat greater generality, to allow for the specification of an intensity of radiation that is incident on the problem at the boundary. We let I_B be this incident intensity, and if $\mathbf{n}_B$ is the unit outward-directed normal vector for a piece of the boundary, then I_B is defined for ray directions $\mathbf{n}$ that obey $\mathbf{n}\cdot\mathbf{n}_B < 0$, i.e., that point inward. Now, we do not know how much radiation will shine *out* of the problem at the boundary, but suppose we knew what the average cosine was for this *outward* intensity. That is, we think we are given everything we want to know about the *inward* intensity, but we make an *ansatz* about the *outward* intensity. The *ansatz* is

$$\frac{\int\limits_{\mathbf{n}\cdot\mathbf{n}_B>0} d\Omega\, \mathbf{n}\cdot\mathbf{n}_B I(\mathbf{n})}{\int\limits_{\mathbf{n}\cdot\mathbf{n}_B>0} d\Omega\, I(\mathbf{n})} = \langle\mu\rangle, \tag{11.11}$$

a value we think we know. If the integrals in the numerator had been over all solid angles instead of the outward hemisphere then the ratio would have been $\mathbf{n}_B\cdot\mathbf{F}/cE$. We add and subtract integrals over the inward hemisphere to make it look somewhat like that and find

$$\frac{\mathbf{n}_B\cdot\mathbf{F} + \int\limits_{\mathbf{n}\cdot\mathbf{n}_B<0} d\Omega\, |\mathbf{n}\cdot\mathbf{n}_B| I_B(\mathbf{n})}{cE - \int\limits_{\mathbf{n}\cdot\mathbf{n}_B<0} d\Omega\, I_B(\mathbf{n})} = \langle\mu\rangle. \tag{11.12}$$

This is the desired result. When it is rearranged it becomes a linear relation connecting the normal component of $\mathbf{F}$ with E at the boundary, possibly including an inhomogeneous term when there is incident radiation. It is a boundary condition of mixed type, i.e, neither Neumann ($\mathbf{n}_B\cdot\mathbf{F}$ specified) nor Dirichlet (E specified). We frequently want the boundary condition when there is vacuum or a "black

absorber" outside so that $I_B = 0$. Then the relation is simply

$$\mathbf{n}_B \cdot \mathbf{F} = \langle \mu \rangle cE. \tag{11.13}$$

Oh yes, what do we take for $\langle \mu \rangle$? The most popular value is $1/2$, and an argument for this was suggested in the earlier discussion of the exact Hopf function. Equation (11.13) with the choice $\langle \mu \rangle = 1/2$ is the Milne boundary condition. When (11.13) is combined with the Fick's law formula (11.8) for the flux, the vacuum boundary condition takes the form

$$E = -\frac{1}{3\langle \mu \rangle \kappa_R \rho} \frac{\partial E}{\partial n}, \tag{11.14}$$

in which $\partial E/\partial n$ is the normal derivative of E. A geometrical picture that goes with this equation is that a linear extrapolation of E outside the boundary reaches a value of zero at a distance $1/(3\langle \mu \rangle)$ mean free paths from the boundary; this is the *extrapolation length* implicit in the boundary condition. The extrapolation length is 2/3 of a mean free path if $\langle \mu \rangle$ is taken to be 1/2, and it is $1/\sqrt{3}$ mean free paths if $\langle \mu \rangle$ is taken to be $1/\sqrt{3}$. In a scattering-dominated diffusion problem the energy density in the interior, i.e., deeper within the medium than the boundary layer, is approximately proportional to the solution of Milne's first problem, which is $E \propto \tau + q(\tau) \approx \tau + q_\infty$, in which $q(\tau)$ is the Hopf function. If this relation is extrapolated to the place where $E = 0$, then the extrapolation length must be $q_\infty \approx 0.710\,45$ mean free paths. The most commonly used value for the extrapolation length is 2/3.

There is one more boundary condition that applies in other cases, and that is the reflection or symmetry condition. If a perfect mirror or perfect diffuse reflector is applied to the surface then the incoming intensity is forced to be exactly equal to the outgoing intensity and the flux vanishes. This also occurs when a piece of the boundary is part of a plane of reflection symmetry. Thus $\mathbf{n}_B \cdot \mathbf{F} = 0$ for those parts of the boundary. Notice that it is only the normal component of the flux that vanishes in this case. This boundary condition is exactly the Neumann type. Dirichlet boundary conditions do not seem to be quite physical. This is a statement that the full-sphere average of the intensity at a boundary point is a specified value. What is unphysical here is that it commits the problem to enter a conspiracy with the agents outside the boundary to make the average of their respective contributions to E come out to a given value. If the outside world is just a thermal bath, then the incoming intensity is the corresponding Planck function, but the emergent intensity is whatever it is, and the average will not necessarily be the same Planck function. The comparable specification of a nonzero value for the normal flux is more physical. The significance of this specification is that the agents outside have a battery that releases energy at a definite rate, and they capture whatever energy comes out through the boundary and give that back plus

the energy released by their battery. When we put an inner boundary radius on a stellar atmosphere problem, and replace all of the star within that radius with a boundary condition, we are making an assumption like this. In this case the "battery" actually exists and is the nuclear energy source at the center of the star.

The Eddington approximation, unlike thermal diffusion, gives quite reasonable results in the optically-thin parts of the star. This is not to say that it is *accurate*, just that it is qualitatively correct. As we saw earlier, it gives an error of order 20% in the Eddington factor at $\tau = 0$ in the Milne problem. It yields a wave equation for light waves, which is qualitatively correct, for which the wave speed is $c/\sqrt{3}$, which is off by 42%. As Mihalas and Mihalas (1984, p. 518) say, in discussing radiative waves with thermal relaxation, "In our opinion the Eddington approximation should always yield results that are at least qualitatively correct."

One path toward making the Eddington approximation more accurate is to include an Eddington factor, which we discuss in Section 11.5.

11.4 Diffusion solvers

Solving a diffusion problem in one dimension that has been put into finite-difference form using a centered three-point formula such as this:

$$-A_i J_{i-1} + B_i J_i - C_i J_{i-1} = D_i, \tag{11.15}$$

is very simple indeed. The forward and back recursion scheme given by

$$E_i = \frac{C_i}{B_i - A_i E_{i-1}}, \tag{11.16}$$

$$F_i = \frac{D_i + A_i F_{i-1}}{B_i - A_i E_{i-1}}, \tag{11.17}$$

$$J_i = F_i + E_i J_{i+1} \tag{11.18}$$

is solved in the forward direction to obtain the Es and Fs, then a back substitution using the third equation gives the unknowns. If the tri-diagonal matrix $(-A_i \quad B_i \quad -C_i)$ is diagonally-dominant, so $B_i > |A_i| + |C_i|$, the recursion is guaranteed to be stable. This condition is almost always met with centered differencing of diffusion equations, so our problem is solved. The tri-diagonal recursion is so efficient that only a handful of floating-point operations are needed to obtain each of the unknowns we want; that is as good as it gets. So the 1-D problem is solved. Life is more difficult in two and three dimensions, and that is the topic of this section.

First, let us consider what *not* to do, if efficiency is the goal. A finite-difference formula representing a diffusion equation in two dimensions very often connects five or nine neighboring points on a more-or-less rectangular grid. The matrix of this system of linear equations has one row for each equation, and nonzero entries

in that row in all the columns corresponding to mesh points that are coupled to the point in the middle. If the mesh points are ordered raster-fashion, going across in the x direction first, then up in y, the neighbor points in x to the middle point produce matrix entries immediately adjacent to the diagonal. But the neighbor points up or down in y produce matrix elements separated from the diagonal by about N_x columns, where N_x is the size of the mesh in the x direction. Both normal Gaussian elimination applied to this matrix and block-tri-diagonal elimination with $N_x \times N_x$ blocks lead to a solution cost of order $N_x^3 N_y$ operations. This is a cost that is N_x^2 operations per unknown, which is thousands of times worse than the 1-D case.

How much better can we do? By using iterative linear solution methods the cost can be brought down to something like $N_x^2 N_y$, or N_x operations per unknown. Some methods may do even better than this, but then it depends on how well-conditioned the matrix is. The (relatively) good news is that in three dimensions the scaling for the iterative methods is also of order N_x operations per unknown. Here is a laundry list of linear solver methods that we may want to discuss: conjugate gradient, conjugate gradient preconditioned by different methods, Chebyshev, ORTHOMIN, which is also known as the GMRES method, multigrid, and multigrid with a selection of preconditioners. These are all methods for solving large sparse linear systems. A system of nonlinear equations leads, by applying the Newton–Raphson method, to such a sparse linear system. But it may be that it is painful and expensive to actually compute and store the Jacobian matrix that is needed at each iteration. The Newton–Krylov method(s) are a way of carrying out the Newton iterations simultaneously with the GMRES or other linear solver iterations. All these methods will be discussed briefly in the remainder of this section. A study of a few promising candidate solvers for a radiation diffusion problem was reported by Baldwin *et al*. (1999).

We will follow the discussion by Saad (1996). Our goal is to solve a linear system of equations

$$Ax = b, \tag{11.19}$$

for a vector of unknowns x, which in most cases consist of one unknown per spatial cell in a 2-D or 3-D mesh. The cells, and the unknowns, are ordered in some way, such as in the raster scan.

11.4.1 Jacobi, Gauss–Seidel, and successive overrelaxation (SOR) method

These are the simplest, oldest, and poorest of the available methods. The idea is to separate A into its diagonal, subdiagonal and superdiagonal parts. That is,

$$A = -E + D - F, \tag{11.20}$$

in which D is the diagonal of A, $-E$ is the lower-triangular matrix that is the subdiagonal part of A, and $-F$ is the upper-triangular superdiagonal part of A. The E elements are the terms that couple a given cell to cells that come earlier in the raster scan, and F contains the couplings to cells that come later. In considering Jacobi iteration the linear system is written in this way:

$$Dx = b + Ex + Fx. \tag{11.21}$$

Then we solve by a process of iteration in which the current guess for x is put in on the right-hand side, and the diagonal system is solved for the next guess:

$$Dx^{k+1} = b + Ex^k + Fx^k. \tag{11.22}$$

With luck, this Jacobi iteration will converge. Clearly, if the E and F matrices are small in some sense compared with D, then there should be good convergence. More precisely, the method will converge if the largest, in magnitude, of the eigenvalues of the matrix $D^{-1}(E + F)$ is less than unity. Since for common finite-difference representations of the diffusion operator D is just *equal* to $E + F$ plus source-sink terms that may be small, this eigenvalue may be only slightly less than unity.

The Gauss–Seidel method is described by this equation:

$$-Ex^{k+1} + Dx^{k+1} = b + Fx^k. \tag{11.23}$$

So half of the off-diagonal part of A is kept on the left-hand side for the iteration. This is just about as easy to perform as the Jacobi iteration. For each iteration you scan through the mesh, updating the cells one at a time. When x is corrected in each cell, the new value replaces the old one, and the new value will be used for updating cells that come later in the scan. Only the cells that follow the given one will have just the prior iteration data available. In this case the convergence depends on the eigenvalues of $(D - E)^{-1}F$. Again, the eigenvalues are anticipated to be just slightly less than unity. It is found that they may be twice as far from unity as the eigenvalues for Jacobi iteration, which will cut the number of required iterations in half.

The thing that helps out the convergence of Gauss–Seidel (it would help for Jacobi too, but it is usually used with Gauss–Seidel) is SOR. For SOR, compared with Gauss–Seidel, some of the diagonal part D of A is put on the right-hand side of the equation along with the F part, and the rest of D and the E part are kept on the left:

$$\left(\frac{1}{\omega}D - E\right)x^{k+1} = b + \left(F - \frac{\omega - 1}{\omega}D\right)x^k. \tag{11.24}$$

With $\omega > 1$ this makes the corrections somewhat larger and accelerates the convergence; it can also make the corrections *too* large, and produce divergence (if $\omega > 2$). The optimum value for ω turns out to be

$$\omega = \frac{2}{1 + \sqrt{1 - \lambda^2}}, \tag{11.25}$$

in terms of the largest eigenvalue λ for Jacobi iteration. When ω has this optimum value the SOR eigenvalue becomes $\omega - 1$. SOR can yield a huge gain in convergence rate. When the Jacobi eigenvalue is 0.999, and the Gauss–Seidel eigenvalue is 0.998, then SOR has an eigenvalue of 0.914 provided ω is set to 1.914. That is a speed-up of 89 times. Empirically estimating the Jacobi eigenvalue and the optimum ω is not simple, however.

A useful extension of Jacobi iteration is *block Jacobi* iteration, for which the unknowns are partitioned into some number of groups, and for each iteration the equations belonging to each group are solved for the unknowns for that group using prior values of the unknowns in other groups. This is employed in parallel solution techniques for large systems for which the spatial domain is decomposed into subdomains, and each subdomain is given to a separate processor, or to a set of shared-memory processors. Block Jacobi iteration is by far the simplest method for solving the linear system in this case.

11.4.2 Alternating-direction implicit (ADI) method

In 2-D diffusion problems the matrix A often has the structure of a tri-diagonal matrix in the y direction combined with a tri-diagonal matrix in the z direction, as expected for an operator like

$$-\frac{\partial^2}{\partial y^2} - \frac{\partial^2}{\partial z^2}. \tag{11.26}$$

The matrix A also usually contains some diagonal pieces, such as source-sink terms for radiation diffusion. The essence of the ADI method is to perform two 1-D solutions per full iteration cycle, one in which the z operator is put onto the right-hand side, and a judiciously chosen diagonal component is added to both sides, leading to a tri-diagonal system in y on each z line. The second half of the iteration cycle is the reverse. Since the cost of a tri-diagonal solve is of the same order as the number of unknowns, one full ADI cycle has about the same cost as one SOR cycle. Saad (1996) mentions the result that the optimum convergence rate for ADI, with the best choice of that judiciously-chosen diagonal component, is the same as with a symmetrized SOR, in which the roles of E and F are switched on alternate iterations, using the optimum ω in that case.

It is interesting to consider what the convergence rates actually are, and how they depend on the mesh. For a simple Poisson equation problem with Dirichlet boundary conditions, on a $N \times N$ mesh, the Jacobi eigenvalue is about $\lambda = 1 - \pi^2/(2N^2)$. This means the optimum SOR eigenvalue is roughly $1 - 2\pi/N$. In order to reduce the initial error by a factor 10^6 the number of iterations will need to be $n_{\text{iter}} \approx 3\ln(10)N/\pi \approx 2.2N$. This is the basis for thinking that the iteration count may scale with the size of the mesh.

11.4.3 Krylov methods in general

A great many of the current iterative solution methods fall under the general description of "Keep multiplying the matrix into the current residual, and at each step combine all the vectors together in some way to get the next guess." This collection of vectors, which has the generic form $\{v_0, Av_0, A^2v_0, \ldots, A^{m-1}v_0\}$, spans what is called a Krylov subspace. What distinguishes the different methods in this class is the "combine all the vectors together" part. A recurrent theme is to choose the next iterate so that the error, or the residual, will be orthogonal to the Krylov subspace. Since the subspace becomes steadily larger as the iteration proceeds, the error can be quenched fast and faster.

11.4.4 CG method

The CG method of Hestenes and Stiefel (1952) and Lanczos (1952) is a very useful method for symmetric positive-definite matrices (all the eigenvalues are positive). Diffusion problems can in principle always lead to symmetric positive-definite systems of finite-difference equations, but in the application this is not always true. When it is, then the CG method is an excellent choice, usually with a suitable preconditioner. The CG algorithm, like the other Krylov methods, repeatedly corrects the current estimate of x by trying a displacement in the direction of a vector p, the search direction, which varies from iteration to iteration. The correct distance to move along the p direction for the next iterate is determined so that the new residual will be orthogonal to the Krylov subspace built up in steps from the initial residual. A wonderful property of the symmetric system is that if the new residual is just made orthogonal to the previous one, then orthogonality to the whole subspace is guaranteed. Then to find the new search direction the new residual is orthogonalized with respect to the previous search direction. This also guarantees orthogonality with all the previous ones. Given the new search direction, the next iteration can begin.

The mathematical expression of the algorithm is the following, where (f, g) is the notation for the vector inner product, which might also be written $f^{\mathrm{T}}g$, with $^{\mathrm{T}}$

standing for the transpose:

$$\alpha_j = \frac{(r_j, r_j)}{(Ap_j, p_j)}, \tag{11.27}$$

$$x_{j+1} = x_j + \alpha_j p_j, \tag{11.28}$$

$$r_{j+1} = r_j - \alpha_j A p_j, \tag{11.29}$$

$$\beta_j = \frac{(r_{j+1}, r_{j+1})}{(r_j, r_j)}, \tag{11.30}$$

$$p_{j+1} = r_{j+1} + \beta_j p_j. \tag{11.31}$$

The iteration begins with any good choice for x, and with $r = p = b - Ax$. At each step there is one matrix-vector multiplication required, and two inner products. The total number of floating-point operations is about equal to the number of nonzero elements in A.

The convergence rate varies as the iterations proceed, but the worst-case estimate depends on the condition number κ of A. The condition number is defined as $\kappa = \lambda_{\max}/\lambda_{\min}$ in terms of the largest and smallest eigenvalues of A. The eigenvalues are all real and positive. The error after many iterations is multiplied by the factor $(\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$ each iteration. For the Poisson problem mentioned above, the condition number of A is something like $\kappa = N^2/\pi^2$, with the result that the error amplification factor is $\approx 1 - 2\pi/N$, which is the same as for optimum SOR and about the same as optimum ADI. The advantage of CG is that there are no parameters to tune, the Achilles heel of the latter methods.

11.4.5 GMRES, ORTHOMIN, Ng, BCG, and Chebyshev methods

We turn to an algorithm that can be used for nonsymmetric matrices, which unfortunately often occur even when they ought not. This is the GMRES method. The idea of this method and a couple of its variants is that the m-dimensional Krylov subspace based on A and the initial residual vector r_0 is built up. The dimension m of the subspace may have to be chosen at the outset. Then a procedure (Gram–Schmidt or Householder's method (Saad, 1996)) is used to orthogonalize the vectors $r_0, Ar_0, \ldots, A^{m-1}r_0$ to form a set of basis vectors. Given this orthonormal set, it is easy to select a candidate solution x_m by adding to x_0 a linear combination of the basis vectors, where, in the case of GMRES, the L_2 norm of the residual $b - Ax_m$ is minimized. If this answer is not good enough, and it will not be if $m \ll N$, then the choices are: (1) start over again with x_m in place of x_0, or (2) keep on going, with the orthogonalization procedure applied to only the most recent k vectors. The latter modification is called quasi-GMRES or QGMRES by Saad (1996). The orthogonalization procedure in the basic method, and especially

in QGMRES, becomes complex when the goals of well-conditioned numerical operations and storage minimization are taken into account.

There is a reorganized form of GMRES, called generalized conjugate residual (GCR), that recursively defines search direction vectors p_j such that all the vectors Ap_j are orthogonal. These take the place of the orthonormal basis in the GMRES method. The Krylov subspace is the same in the two cases, and both methods minimize the same norm of the residual, and therefore should be algebraically equivalent. The GCR algorithm is described by

$$\alpha_j = \frac{(r_j, Ap_j)}{(Ap_j, Ap_j)}, \tag{11.32}$$

$$x_{j+1} = x_j + \alpha_j p_j, \tag{11.33}$$

$$r_{j+1} = r_j - \alpha_j Ap_j, \tag{11.34}$$

$$\beta_{ij} = -\frac{(Ar_{j+1}, Ap_i)}{(Ap_i, Ap_i)} \quad \text{for } i = 0, 1, \ldots, j, \tag{11.35}$$

$$p_{j+1} = r_{j+1} + \sum_{i=0}^{j} \beta_{ij} p_i. \tag{11.36}$$

The residual is available at each step, so it is easy to decide when to stop iterating. Unfortunately, unlike CG, the projection process involves more and more terms as j increases. The GCR algorithm, like GMRES, can either be stopped and restarted, or the projections can be limited to the most recent k vectors, *viz.,* the loop on i and the sum over i can be limited to $i = j - k + 1, \ldots, j$. The GCR algorithm is called ORTHOMIN(k) in that case. See Vinsome (1976).

A method due to Ng (1974) has been used by Olson, Auer, and Buchler (1986); it is described by Auer (1987) and compared by him with ORTHOMIN (Auer, 1991). It has very much the same flavor as ORTHOMIN, and is described as follows. A certain number k of simple relaxation iterations $x^{n+1} = x^n + b - Ax^n$ are performed, and the residuals $r^n = b - Ax^n$ are recorded. Then it is required that a new candidate for x given by

$$x = x_k - \sum_{p=0}^{k-1} \alpha_p (x^k - x^p) \tag{11.37}$$

should yield the minimum possible residual with respect to possible choices of the coefficients α_p. When this is worked out (see Auer (1991)) it implies that the final residual is r_k projected orthogonal to the space spanned by the vectors $r_k - r_p$, $p = 0, \ldots, k-1$. It turns out that Ng's method is identical to GMRES with $m = k$ and with a restart after each k iterations. The difference with ORTHOMIN(k) is that ORTHOMIN keeps on going without a restart, but uses a truncated orthogonalization.

Another wrinkle on Krylov-space methods is the bi-conjugate gradient (BCG) method. This uses a method due to Lanczos to develop two Krylov subspaces, one based on A and the other based on its transpose A^T. Sequences of basis vectors are chosen that are mutually orthogonal rather than orthogonal within each set. The logic is very similar to that of CG, but the solution at each step does not minimize the norm of a residual as in the CG case. Saad (1996) provides the details.

The final Krylov-type method we wish to discuss is the Chebyshev method described by Manteuffel (1977, 1978). The idea behind the Chebyshev method is that the residual at the nth step of the iteration is equal to the matrix $T_n[(d - A)/c]/T_n(d/c)$ multiplied by the initial residual. This matrix is a combination of powers of A up to the nth degree, so this is a Krylov-subspace method like the others we have discussed. The $T_n(z)$ are the complex Chebyshev polynomials and c and d are constants that are estimated based on knowledge of the eigenvalue spectrum of A. First of all, the method will not work unless all the eigenvalues have positive real part (A is positive definite). Then c and d should be such that an ellipse with its center at d and foci located at $d \pm c$ should be the smallest one possible that encloses all the eigenvalues. (If the major axis of the ellipse is aligned with the imaginary axis then c can be imaginary.) The Chebyshev polynomials have a maximal property, i.e., of the polynomials of a given degree that are bounded by 1 in the interval $-1 \leq z \leq 1$, they are the largest possible outside that range. This translates into making the residual as small as possible. The recurrence relation for Chebyshev polynomials leads to this setup of the iteration method:

$$x_n = x_{n-1} + dx_{n-1}, \tag{11.38}$$

$$r_n = b - Ax_n, \tag{11.39}$$

$$p_2^n = \frac{c^2 p_1^{n-1}}{4d - c^2 p_1^{n-1}}, \tag{11.40}$$

$$p_1^n = \frac{1 + p_2^n}{d}, \tag{11.41}$$

$$dx_n = p_2^n dx_{n-1} + p_1^n r_n. \tag{11.42}$$

The starting values are $p_1^0 = 2/d$, $p_2^0 = 0$ and $dx_0 = r_0/d$.

The estimation of c and d can be problematic. Calvetti, Golub, and Reichel (1994) provide an efficient algorithm for estimating the convex hull of the eigenvalues of A based on modified moments that are computed as the iteration proceeds. After a certain fixed number of iterations, or sooner if the residuals begin to increase, the convex hull estimate is updated, new values are derived for c and d, and the iteration is restarted.

The Chebyshev method has been successfully used in ALTAIR (Castor, Dykema, and Klein, 1992) for iterative solution of the system of kinetic equations

for the atomic populations, and also to accelerate the net radiative bracket iterations.

11.4.6 Multigrid method

The multigrid method is not simply a method, it is a whole field of research. The reader is recommended to visit the web site `http://casper.cs.yale.edu/mgnet/www/mgnet.html` and consult the references listed there, such as the text by Wesseling (1992) and the tutorials of Henson (1987, 1999). The following discussion is aimed at merely giving the flavor of multigrid methods, and the literature must be consulted for the details. A system of linear equations $Ax = b$ may describe an elliptic PDE such as the radiation diffusion problem. The matrix A quite possibly has nice properties such as being symmetric and positive definite. Multigrid is the name for a method in which the solution $x = A^{-1}b$ is approximated in this way:

$$x \approx P A_{\text{coarse}}^{-1} R b, \tag{11.43}$$

in which A_{coarse} is a substantially smaller matrix than A, and P and R are rectangular matrices. The matrix R is called the restriction matrix, because it restricts or projects the vector it acts on to a smaller-dimensional subspace of the space containing x and b. The matrix P is called the prolongation matrix because it prolongs or interpolates the data from the subspace into the original larger space. In order to fix the ideas we can think of A as being a finite-difference operator on a fine mesh with a mesh spacing h, and A_{coarse} is the similar operator on the mesh with spacing $2h$, i.e., with every second mesh line omitted. In this picture P would be written as I_{2h}^{h} and R as I_{h}^{2h}. For a 2-D problem the dimension of A_{coarse} would be $1/4$ as large as that of A, and in three dimensions it would be$1/8$.

The coarse system, with mesh spacing $2h$, may still be too big to be solved readily. The coarsening process can then be applied recursively, so that

$$x \approx I_{2h}^{h} I_{4h}^{2h} \cdots I_{H}^{H/2} A_{H}^{-1} I_{H/2}^{H} \cdots I_{2h}^{4h} I_{h}^{2h} b. \tag{11.44}$$

The idea is that the mesh-space doubling proceeds to the point that the linear system on the coarsest mesh H is trivial to solve. The whole process of evaluating x using (11.44) is called a V-cycle; the picture is that the system size goes down, down, down as the restriction operators are applied to b, then the coarsest system is solved directly, after which the prolongation operators are applied up, up, up to give the answer on the finest mesh. The step-doubling and step-halving picture is only generic, of course. The restriction and prolongation operators can be anything that is convenient for the problem at hand, the only requirement being that they are readily applied and leave a small enough system at the coarsest level.

There are additional requirements if the matrices at every level are to preserve the symmetric positive-definite property of A itself. One of these is the symmetry condition

$$I_{2h}^{h} \propto (I_{h}^{2h})^{T}. \tag{11.45}$$

The replacement of the solution $x = A^{-1}b$ by the V-cycle does not solve the system exactly since the fine grid is finer than the coarse grid for a reason: the answer is more accurate. Thus it is still necessary to apply some relaxation of the solution on the finest grid. After one or more applications of the relaxation equation $x^{k+1} = x^k + r^k$, with $r^k = b - Ax_k$, there will be a final residual r. This is the quantity that should be used in place of b at the beginning of the V-cycle. At the end of the cycle, the result x is really the correction Δx that should be added to the solution on the finest scale. In fact, one relaxation operation can be applied at each level of refinement, during the down-down-down part of the V-cycle, so that the residual from that operation becomes the right-hand side for the system at the next-coarser level. Then on the up-up-up half of the V-cycle the prolonged corrections from the coarser level are added to the stored solution at that level to be passed on to the next finer level.

The multigrid methods succeed because relaxation at the finest scale quickly reduces the short-wavelength errors in the solution, while not affecting very much the long-wavelength error modes. But the hierarchical multigrid treatment extinguishes those long-wavelength errors. The cost of applying one V-cycle is just a modest factor larger than one relaxation cycle at the finest scale, so this improvement in reducing long-wavelength errors is almost free. Multigrid methods can have an iteration count that is well below the size N of the mesh; instead of being of order 100, iteration counts $\approx$10 are not unusual.

The multigrid application that is discussed by Baldwin *et al.* (1999), is more complicated than we have just described, and indicates how varied the multigrid concept can be. The multigrid approach in this case is called semicoarsening multigrid, or SMG. The "semi" in the name refers to the fact that in the 2-D problems considered only one direction, say x, is coarsened. The coarse operator at each level still includes the full fine-scale coupling in the other direction. The update operations at each level include a tri-diagonal solve in the y direction. Baldwin *et al.*, describe using SMG by itself, and also including one step of CG iteration before and after the V-cycle, i.e., using SMG as a preconditioner for CG. This turned out to be the most efficient of all the methods Baldwin *et al.*, compared on several of their test problems. Its competitors were simple SMG and one of the variations of preconditioned CG that will be described below.

11.4.7 Preconditioning

The topic of this subsection has already been mentioned several times, so we should find out what is meant by preconditioning. Recall that the CG method has an asymptotic convergence ratio given by $(\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$, where $\kappa = \lambda_{\max}/\lambda_{\min}$. The ratio is the condition number of A. A matrix A with a large value of κ is called ill-conditioned; a matrix with a small one (i.e., close to unity) is called well-conditioned. For an ill-conditioned matrix the convergence ratio is very close to unity, which means that CG, or any other iterative method, will be very slow to converge. In the other extreme, a matrix with a condition number of unity is already very close to a scalar times the identity matrix, which makes the convergence of the iterative methods immediate. Preconditioning is then the name for an operation that will take an ill-conditioned matrix and make it into a well-conditioned one, or at least better.

Consider again the simple relaxation method,

$$x^{k+1} = x^k + b - Ax^k = b + (I - A)x^k. \tag{11.46}$$

If A is very close to the identity matrix the convergence will be swift. Suppose A is not very close to the identity, but that another matrix M is at hand that is close enough to A that $M^{-1}A$ is reasonably close to the identity. Then if M^{-1} is applied to the linear system before setting up the relaxation equation, the iteration becomes

$$x^{k+1} = x^k + M^{-1}(b - Ax^k) = M^{-1}b + (I - M^{-1}A)x^k. \tag{11.47}$$

We now hope that the condition number of $M^{-1}A$ is much closer to unity than was the condition number of A, and that therefore the convergence ratio will be much less. In applying preconditioning we do not actually demonstrate the matrix M^{-1}, or multiply it by A. We just have to be able to solve a linear system with matrix M. The preconditioned iteration of whatever kind goes like this. First evaluate the residual in the accurate linear system using the current x:

$$r^k = b - Ax^k. \tag{11.48}$$

Then solve the following system for the preconditioned residual:

$$M\tilde{r}^k = r^k. \tag{11.49}$$

Now proceed with the chosen iteration method using $\tilde{r}^k$ where the residual would normally appear. Iteration methods like CG and GMRES require being able to multiply a vector p by A, and they generate the residual using a recursion relation. In this case, for the preconditioned system, p is first multiplied by A, and then the

linear system $Mv = Ap$ is solved for the vector v that is put in the place where Ap should be.

In the previous subsection we discussed using multigrid as a preconditioner, and in the application by Baldwin *et al.* (1999) it was used to precondition CG. The second simplest of all preconditioners is the diagonal of the matrix, $M = D$. (The simplest is no preconditioning.) Diagonal preconditioning of simple relaxation is just Jacobi iteration. Diagonal preconditioning of CG may also be a good thing to do, as shown by Baldwin *et al.* CG is unaffected by a scale factor applied to the entire linear system, but diagonal scaling will balance the diagonal elements in different rows of the matrix, and according to Gershgorin's theorem[2] this should help balance the eigenvalues and make the condition number smaller. But there are better preconditioners for CG, as we see next.

Probably the most important preconditioner is incomplete LU factorization (ILU). We recall that LU factorization is the decomposition of A in this way: $A = LU$, in which L is a lower-triangular matrix with a unit diagonal, and U is an upper-triangular matrix. The factorization makes it very easy to then solve a linear system $Ax = b$, since $b = U^{-1}(L^{-1}b)$ can be evaluated by first doing $v = L^{-1}b$ recursively in the forward direction, then doing $U^{-1}v$ by recursion in the backward direction. If A is a sparse matrix, so that the nonzero elements in each row span quite a large number of columns on either side of the diagonal (for example, about N_x on each side for a 2-D diffusion problem with a 5-point or 9-point stencil), the LU decomposition will produce matrices L and U in which the intervening elements that are zero in A have become nonzero. That is, there is fill-in of the sparsity pattern. This is why the exact LU decomposition of A is expensive to do.

ILU is defined in this way. Perform a normal LU decomposition, except that at each point of the elimination process, if a nonzero value would be generated for some element of L or U where you do not want one, then discard that element and proceed. The "where you do not want one" part gives you some latitude. The commonly adopted choice is to throw away any elements that are places where the element of A vanishes. That is, "where you do not want one" is any place where $a_{ij} = 0$. Saad (1996) discusses the pseudo-code for achieving this. Saad also proves that the ILU factorization is a well-conditioned operation if A is an M-matrix.[3] Some of the iterative methods such as CG can be proved to be convergent only if A is an M-matrix. The result of ILU preconditioning of an M-matrix is also an M-matrix.

[2] The eigenvalues of a matrix lie in the union of the circles in the complex plane centered on each diagonal element with a radius equal to the sum of the absolute values of the remaining elements in the corresponding row; the same is true for columns.

[3] A matrix A is an M-matrix if it has a positive diagonal, negative off-diagonal elements, and the elements of A^{-1} are positive. The last condition can be replaced by the condition that the spectral radius of $I - D^{-1}A$ is less than unity, where D is the diagonal of A.

ILU is used as a preconditioner by letting the matrix M discussed earlier be LU, where L and U are the results of the ILU process. The ILU decomposition would be done once and for all and stored during the iterations. The solution of $Mv = Ap$ discussed above is evaluated as $v = U^{-1}(L^{-1}Ap)$. Saad provides examples of applying GMRES with ILU preconditioning as just described, and it shows big gains for several of the examples over simple GMRES.

Saad also discusses other variants of ILU. One of these is ILUT, which stands for incomplete LU factorization with thresholding. The ILUT algorithm takes two parameters. The first is a tolerance such that a fill-in element is discarded if its magnitude is less than the specified tolerance times the norm of the row. The second parameter is the maximum number of fill-in elements that will be kept, based on a list in which the elements are ordered by decreasing magnitude. Sample calculations by Saad show that ILUT is significantly more robust, and also faster, than simple ILU. For the test cases in Baldwin *et al.*, ILUT–GMRES performed fairly well, but was generally outperformed by preconditioned CG and multigrid. Of course, ILUT–GMRES is a method that works on nonsymmetric matrices for which CG variations do not, and many implementations of multigrid do not either.

For positive definite symmetric matrices the method of LU factorization can be modified somewhat to preserve the symmetry in the factors. It also allows only one of the factors to be stored. This is the Cholesky decomposition,

$$A = LL^T, \tag{11.50}$$

in which a single lower-triangular matrix L appears, and U has been replaced by the transpose of L. The algorithm for performing the Cholesky decomposition is almost the same as for LU except the square root of the diagonal element is extracted at each step and used to scale the column rather than using the diagonal element itself. The *incomplete Cholesky* (IC) preconditioner, introduced by Meijerink and van der Vorst (1977), is arrived at by performing a Cholesky factorization and discarding fill-in elements exactly as in ILU. Simple IC decomposition, with all fill-in elements discarded, is shown by Meijerink and van der Vorst to always succeed if A is an M-matrix. The implementation of IC-preconditioned conjugate gradient (ICCG) by Kershaw (1978) for radiation diffusion problems shows a great superiority over Gauss–Seidel, ADI and block-SOR. In Baldwin *et al.* a threshold variation ICT of IC preconditioning was applied with criteria the same as the ILUT factorization just described. In their sample calculations a threshold of 10^{-4} was allowed and a generous limit on the number of fill-in elements. The ICT-CG method turned out to be quite competitive with multigrid as the best method.

11.4.8 Nonlinear systems; Newton–Krylov method

The generic PDE for radiation diffusion is

$$\frac{\partial E}{\partial t} - \nabla \cdot [D(T)\nabla E] = \kappa\rho(4\pi B - cE), \tag{11.51}$$

where the diffusion coefficient D is a function of the material temperature T, which is determined by

$$\frac{\partial e(T)}{\partial t} = \kappa(cE - 4\pi B). \tag{11.52}$$

The dependences of D on T, of B on T, of κ on T, of e on T, and even of D on E and ∇E, are all quite nonlinear. The nonlinearity persists even if the equilibrium-diffusion assumption $cE = 4\pi B$ is made. To ensure stability this equation is discretized in time implicitly, as discussed earlier. That means that the time-advanced E^{n+1} will appear in the diffusion coefficient, as well as in the ∇E factor, as in

$$E^n - E^{n+1} + \Delta t \nabla \cdot \left[D\left(\frac{T^n + T^{n+1}}{2}\right) \nabla \left(\frac{E^n + E^{n+1}}{2}\right)\right] = 0, \tag{11.53}$$

in which T^{n+1} comes from E^{n+1} by an auxiliary calculation. This is to be solved for E^{n+1}. This is the prototype nonlinear diffusion problem.

The solution choices are: (1) lag D by using $D(T^n)$ instead of the time-centered form; (2) use the time-centered D but solve the equation in a Picard iteration in which D is updated after each solve;[4] (3) use Newton–Raphson iteration on E^{n+1}, including the variation of T^{n+1} and therefore D with E^{n+1}. Choice (1) is unacceptable because of inaccuracy and possible problems with thermal instability. For choice (2) the convergence is not nearly as good as with choice (3).

The Newton–Raphson method for a system of nonlinear equations $F(X) = 0$ is the iteration

$$J(X^n)X^{n+1} = J(X^n)X^n - F(X^n), \tag{11.54}$$

in which J stands for the Jacobian matrix $J = \partial F/\partial X$. The Jacobian of this system will be one of those large, sparse matrices, so this linear system that must be solved for each Newton iteration falls in the category we have been discussing. But an additional consideration is that the Jacobian matrix elements themselves may be costly to evaluate. In some other nonlinear problems, not radiation diffusion, it may be quite difficult to explicitly perform the differentiation for the Jacobian. This is where the Newton–Krylov method comes to the rescue.

[4] Picard iteration is an elegant way of saying "substitute the unknown back in and do it again."

Recall that all the Krylov-subspace methods for solving linear systems depend on the matrix A (here to be replaced by the Jacobian J) only through its products Av with specified vectors. But we have a way of doing Jv; it is

$$Jv = \lim_{\epsilon \to 0} \frac{1}{\epsilon}[F(X^n + \epsilon v) - F(X^n)]. \tag{11.55}$$

If we simply evaluate $[F(X^n + \epsilon v) - F(X^n)]/\epsilon$ with a suitable small ϵ, we should get an approximate value of Jv that is good enough for the purpose. The Newton–Krylov method, first applied to problems such as this by Brown and Saad (1990), is a double iteration with Newton iterations, and inner iterations using GMRES or another Krylov method, and using the relation just given to replace matrix-vector products.

The GMRES iterations may well need preconditioning to converge satisfactorily, but the preconditioners often explicitly use the matrix, e.g., as in ILU or IC preconditioning. What should be done about this? In a study of a preconditioned Newton–Krylov solution of equilibrium diffusion problems, Rider, Knoll, and Olson (1999) use one multigrid V-cycle based on the matrix A of the *linear* diffusion equation, i.e., the one with fully lagged coefficients, to precondition the GMRES iterations. It is not necessary to update A during the Newton–Raphson iterations. The amount of GMRES iteration within the Newton–Raphson loop is adjustable. It could fully converge the GMRES for every Newton iteration, or do only a single iteration, or something in between. Rider *et al*. choose to do a variable number of GMRES iterations, with the convergence tolerance tightening as the Newton iteration proceeds.

Jones and Woodward (2001) examine a problem of ground-water diffusion that is described using a nonlinear advection–diffusion equation. They apply the Newton–Krylov method using GMRES with two kinds of multigrid preconditioner: one uses pointwise coarsening and the other, the SMG from above, uses coarsening by planes through the mesh. The relaxation applied at each level is Gauss–Seidel. The GMRES convergence tolerance is adjusted dynamically as the Newton–Raphson converges. The results show that there is a trade-off between the simpler and cheaper pointwise multigrid and the more robust SMG.

The Chebyshev method can also serve as a nonlinear solver. Hyman and Manteuffel (1984) describe a nonlinear method that exploits the eventual linearity of Picard-type iteration $x^{n+1} = F(x^n)$ to apply ideas from the Chebyshev algorithm to estimate the optimum coefficients α and β in a relaxation equation

$$x^{n+1} = x^n + \alpha r^n + \beta(x^n - x^{n-1}), \tag{11.56}$$

where the residual is defined to be

$$r^n = x^n - F(x^n). \tag{11.57}$$

This performs quite well on a 3-D solution of Burgers's equation. Castor *et al*. (1992) use Chebyshev acceleration for the Picard iteration applied to the Net Radiative Brackets in the multilevel ETLA method (see Section 9.1.4) for non-LTE radiative transfer. The nonlinearity is severe in these problems, and the Jacobian is not readily obtained, nor is a preconditioner available. The Chebyshev acceleration method is the best that has been found for this problem.

Ng acceleration has also been used for nonlinear calculations. For example, the CONRAD code (MacFarlane, 1993) in use at the Fusion Technology Institute of the University of Wisconsin uses Ng acceleration for the level populations in a 1-D collisional-radiative equilibrium model based on single-flight escape probabilities.

11.5 Eddington factors and flux limiters

The Eddington-factor method, also called the variable Eddington factor (VEF) method, is simple in concept: if the precise ratio of the pressure tensor to the energy density were included as an *ad hoc* multiplier in the Eddington approximation equations, they would then become exact. This method of solving transport problems originated with the work of Gol'din (1964), under the name quasi-diffusion. The Eddington tensor nomenclature was introduced by Freeman *et al*. (1968).

Let the Eddington tensor be defined by

$$\equiv \quad E. \tag{11.58}$$

Substituting a relation like this into the monochromatic second moment equation in the comoving frame, (6.50), for example, leads to

$$\frac{1}{c}\frac{\partial \mathbf{F}_\nu}{\partial t} + \frac{1}{c}\nabla \cdot (\mathbf{u}\mathbf{F}_\nu) + c\nabla \cdot (\ {}_\nu E_\nu) = -k_\nu \mathbf{F}_\nu. \tag{11.59}$$

If, as in Section 11.3, we drop the $1/c$ terms here and solve for $\mathbf{F}_\nu$ which is substituted into the first moment equation, we get

$$\rho \frac{D(E_\nu/\rho)}{Dt} + E_\nu \ {}_\nu : \nabla \mathbf{u} - \nabla \cdot \left[\frac{c}{\kappa_\nu \rho} \nabla \cdot (\ {}_\nu E_\nu) \right] = 4\pi j_\nu - \kappa_\nu \rho c E_\nu. \tag{11.60}$$

A frequency-averaged version of (11.60) takes the place of (11.9), and can be used in the same way. The solution cost is much the same, except that the partial differential equation is not self-adjoint in general, and therefore cannot, even in principle, be approximated by a difference representation with a symmetric matrix. Thus we are compelled to use nonsymmetric solvers from the outset.

Some other features of the VEF equation are seen by taking certain limiting cases. If the material terms and the velocity terms are dropped in (11.59) and also in (4.27), and then $\mathbf{F}_\nu$ is eliminated between them, the result is

$$\frac{\partial^2 E_\nu}{\partial t^2} - c^2 \boldsymbol{\nabla} \cdot [\boldsymbol{\nabla} \cdot (\ _\nu E_\nu)] = 0. \tag{11.61}$$

This equation has wave solutions that locally obey the dispersion relation

$$\frac{\omega^2}{c^2} = \mathbf{k} \cdot \ _\nu \cdot \mathbf{k}. \tag{11.62}$$

For a radiation front propagating in the direction $\mathbf{n}$ the Eddington tensor will be just $\mathbf{nn}$, which means the dispersion relation is $\omega = \mathbf{k} \cdot \mathbf{n} c$. This means the group velocity is exactly c in the direction $\mathbf{n}$. So the wave speed comes out right.

A similar limit that is informative is to consider a vacuum region in steady state that has a radiation field passing though it. Apparently this field must satisfy

$$\boldsymbol{\nabla} \cdot (\ _\nu E_\nu) = 0. \tag{11.63}$$

If we expand the tensor divergence we find

$$E_\nu \boldsymbol{\nabla} \cdot \ _\nu + \ _\nu \cdot \boldsymbol{\nabla} E_\nu = 0, \tag{11.64}$$

and if we then multiply on the left by the inverse of the matrix $\ _\nu$ and divide by E_ν we get

$$_\nu^{-1} \cdot \boldsymbol{\nabla} \cdot \ _\nu = -\boldsymbol{\nabla} \log E_\nu. \tag{11.65}$$

With a general tensor $\ _\nu$ this equation will be inconsistent, because a condition for solubility is apparently

$$\nabla \times (\ _\nu^{-1} \cdot \boldsymbol{\nabla} \cdot \ _\nu) = 0. \tag{11.66}$$

Putting this argument differently, we can say that since under the stated conditions there is certainly a solution for the radiation field, then (11.66) must be satisfied. But suppose that the tensor has been obtained by some approximation procedure and does not exactly obey (11.66), and suppose that the region is not exactly a vacuum, but there are small values of absorptivity and emissivity that we let tend to zero. Then what we expect is that this limit is a singular limit, and that the solution that is found in the limit does not have a finite value of the flux. This sad expectation is supported by some numerical experience.

This problem significantly impairs the robustness of the VEF method. A possible remedy is the following. Suppose there were an integrating factor q_ν such

that

$$\nabla \cdot (\ _\nu E_\nu) = \frac{1}{q_\nu}\ _\nu \cdot \nabla(q_\nu E_\nu) \tag{11.67}$$

were true regardless of E_ν. Apparently the integrating factor would have to obey this equation

$$_\nu \cdot \nabla \log q_\nu = \nabla \cdot\ _\nu. \tag{11.68}$$

This does not seem like progress since the condition for this to be soluble is also (11.66). But now suppose that we extract a single scalar equation from this vector equation and obtain q_ν from it, then make the replacement (11.67) as an additional approximation. One possibility is to take the divergence of the equation, which gives

$$\nabla \cdot (\ _\nu \cdot \nabla \log q_\nu) = \nabla \cdot (\nabla \cdot\ _\nu), \tag{11.69}$$

as the equation from which q_ν is to be found. Given a numerical estimate of $_\nu$ we would evaluate the second derivative on the right-hand side, then solve the Poisson-equation-like PDE for $\log q_\nu$. The additive constant is unimportant since a multiplicative factor in q_ν cancels out when the integrating factor is used. Because $_\nu$ is symmetric and positive definite this self-adjoint elliptic equation is easy to solve using iterative methods. This approximation shows promise because the curl condition (11.66) is accurately obeyed in both the diffusion limit and the free-streaming limit.

If the formula for the flux is altered using (11.67) then the VEF equation becomes

$$\rho \frac{D(E_\nu/\rho)}{Dt} + E_\nu\ _\nu : \nabla \mathbf{u} - \nabla \cdot \left[\frac{c\ _\nu}{\kappa_\nu \rho q_\nu} \cdot \nabla(q_\nu E_\nu) \right] = 4\pi j_\nu - \kappa_\nu \rho c E_\nu, \tag{11.70}$$

which also now has the nice property of being self-adjoint. In fact, it is this property that makes the equation well-posed for any $_\nu$ in the limit $k_\nu \to 0$.

This approach was introduced by Auer in spherical geometry, for which the curl condition is exactly satisfied, as a means of improving the conditioning of the VEF equation (11.60). We will return to that below in discussing spherical problems.

The question before us now is: where does the Eddington tensor come from? There are two general philosophies. The first is to use an analytic model based on the problem geometry that attempts to capture the main features of the tensor as they depend on that geometry. Let's take spherical geometry as the illustration. At a particular point we set up a local Cartesian coordinate system with the z axis in the radial direction, and x and y tangential. Axial symmetry about z means that the tensor is diagonal and the x and y diagonal elements are equal. Thus the tensor

has the form

$$= \begin{pmatrix} \frac{1-f}{2} & 0 & 0 \\ 0 & \frac{1-f}{2} & 0 \\ 0 & 0 & f \end{pmatrix}, \tag{11.71}$$

since the trace must be unity. In other words, we let the rr component of the tensor be f, then the two transverse components are $(1-f)/2$. We speak of f as *the* Eddington factor. This is defined in the same way as the Eddington factor in slab geometry discussed earlier. A simple analytic model for f of the type we are discussing is the formula that comes from assuming that the photosphere of a star radiates an equal intensity in all directions, and that the space above the photosphere is completely transparent. Thus at a point located above the photosphere the radiation field is constant within a certain cone and zero outside it. The half-angle of the cone is $\theta = \sin^{-1}(R_p/r)$ in terms of the photospheric radius R_p and the local radius r. The cosine of this angle is $\mu = \cos\theta = \sqrt{1-(R_p/r)^2}$. Doing the integrations for P_{rr} and E leads to

$$f = \frac{1}{3}(1+\mu+\mu^2). \tag{11.72}$$

As r approaches R_p the value of μ tends to zero, so the Eddington factor f tends to $1/3$. So in this simple model we would adopt this formula for f in $r > R_p$ and set $f = 1/3$ in $r \leq R_p$. As a practical matter, it has been found that using such formulae for the Eddington factor ameliorates somewhat the errors in the Eddington approximation, but not enough.

The second general approach to the VEF method, now used exclusively, is to employ an auxiliary calculation that solves the radiative transfer equation as accurately as possible, with good resolution in angle space, and obtain the tensor point by point in space from the angle moments derived from this calculation. One may well ask, why bother with the VEF equation at all when an accurate angle-dependent transfer calculation will have to be done anyway? There are at least two reasons. One is that making the large set of radiation transport equations for many angles implicitly coupled through the material temperature leads to a system too costly to solve. For the auxiliary calculation the distribution of temperature is taken as given, which removes the implicitness and thus makes the transfer much cheaper. The second reason has to do with retardation, the presence of the time derivative term in the transport equation. The burden of carrying this term is severe – not in the cost of solving the spatial differencing with this small correction, but in the amount of storage required to carry all the intensities from one time step

to the next. It may be true that dropping retardation still produces a tensor that is sufficiently accurate, even though the flux and energy density in the auxiliary calculation might have unacceptable systematic errors as a result. The case for the VEF method has pros and cons, and is by no means closed. We will return to this discussion in connection with approximate operator iteration methods (ALI), also called preconditioning.

There is another modification to the Eddington approximation that is somewhat related to the use of Eddington factors, and is an alternative to it. This is the *flux limiter*. The primary reference on flux limiters and their connection to the Eddington factor in one dimension is Pomraning (1982). The idea is to discard the $\partial\mathbf{F}/\partial t$ term in the flux moment equation and make the Eddington approximation $_\nu = (E_\nu/3)$, but compensate the errors of these approximations by including a correction factor in the diffusion coefficient:

$$\mathbf{F}_\nu = -\frac{c}{\kappa_\nu \rho} \cdot \nabla E_\nu. \tag{11.73}$$

The tensor (in general) is the flux limiter. The only difference between and is which side of the divergence operator the tensor stands on; the inside for and the outside for . From this point on, the philosophies of flux limiters and Eddington factors begin to differ. There is no practical way to self-consistently calculate a flux limiter so as to produce agreement between solutions using (11.73) and accurate transport solutions. Instead, flux limiters are used in the way that analytically-based Eddington factors might be used but today are not. That is, a relatively simple formula is adopted for the flux limiter that captures some essential features of the problem, but which cannot be very accurate. Flux limiters are intended mainly to compensate for the omission of the $\partial\mathbf{F}/\partial t$ term. The raw Eddington approximation can give a flux that is arbitrarily large compared with cE if the gradient of E is large enough; this is something that can never happen if $\partial\mathbf{F}/\partial t$ is retained. This problem is corrected by making sure that becomes small when ∇E is large, so the flux is indeed limited to be no larger than cE. The tensor is invariably chosen to be a scalar tensor, i.e., just a scalar factor D, and this is considered to be a function of the dimensionless quantity

$$R = \frac{|\nabla E_\nu|}{\kappa_\nu \rho E_\nu}. \tag{11.74}$$

A small value of R means that the ordinary diffusion flux is small compared with cE_ν, and therefore no limiting should be necessary, and D should be $1/3$. A large value of R means that the physical limit on the flux is violated by the ordinary flux

formula, and limiting is needed. The proper limit $\mathbf{F}_\nu \to cE_\nu$ is obtained if

$$D(R) \to \frac{1}{R} \quad \text{for} \quad R \to \infty. \tag{11.75}$$

The literature on flux limiters has become extensive, and we will quote three here:

$$D(R) = \begin{cases} \dfrac{1}{3+R} & \text{sum} \\ \dfrac{1}{\max(3,R)} & \text{max} \\ \dfrac{1}{R}\left(\coth R - \dfrac{1}{R}\right) & \text{Levermore} \end{cases} \tag{11.76}$$

The sum and the max flux limiters are just formulae chosen to have the right limits. Levermore's flux limiter is derived from an application of the Chapman–Enskog method of kinetic theory (Levermore, 1979, Levermore and Pomraning, 1981). Levermore's theory modifies the definitions given here by including a factor of the scattering albedo ϖ in the denominator of the definition of R, and the result for D then contains a factor ϖ in the denominator as well. The effect of the albedo is to leave both limits of the flux limiter unchanged, but to change the typical value of $E/|\nabla E|$ where D switches from one limit to the other, from one mean free path for the total absorptivity to one scattering mean free path. Thus in a problem with almost pure absorption the value of R would be large and the flux would be set to cE_ν even at large optical depth. This is appropriate if there is no internal source of radiation and only a beam incident at the boundary, but it is clearly wrong in the more usual case with an internal source.

There are some comments to be made about the application of flux limiting in an implicit radiation diffusion problem. It is evident that the global geometry of the problem, which determines the angular distribution of the radiation field and implicitly both the Eddington factor and the tensor , cannot be encompassed by formulae like (11.76). That is, no matter how much skill is employed in selecting D, the error will none-the-less be similar to the 20% error of the Eddington approximation in general. For this reason there seems little to choose between the alternative expressions. The second point is that D is a nonlinear function of E_ν and its gradient, and this adds additional nonlinearity to (11.9) beyond that due to the temperature. Furthermore, in two or three dimensions the gradients in the different directions are combined in the diffusion coefficient since D depends on the norm of the gradient. The alternative of using a function for each coordinate direction that depends on that component of the gradient alone can lead to a flux vector that makes a large angle with ∇E_ν, and this is unphysical. The better approach is

either to deal with the nonlinearities using Newton–Raphson or to lag the value of D in time.

11.6 Method of discrete ordinates

This method was originally introduced by Chandrasekhar (1960) to solve the standard problems of monochromatic or gray scattering in 1-D slab geometry, and it is particular to that geometry. We refer to Section 5.2 for the definition of the angle cosine μ and the formulation of the relation between the source function S and the mean intensity J, (5.25). The idea of the method of discrete ordinates is to choose a set of discrete values of $\mu = \{\pm\mu_i, i = 1, \ldots, n\}$ and a quadrature formula

$$\int_{-1}^{1} d\mu\, f(\mu) \to \sum_{i=1}^{n} w_i [f(-\mu_i) + f(\mu_i)]. \tag{11.77}$$

The quadrature will always be normalized so that

$$\sum_{i=1}^{n} w_i = 1. \tag{11.78}$$

Chandrasekhar confined himself to the even-order Gaussian quadratures on $[-1, 1]$ for which the values of μ_i are the positive zeroes of the Legendre polynomials $P_{2n}(\mu)$. The first approximation has a single value of μ_i which is the zero of $P_2(\mu)$, namely $1/\sqrt{3}$. Much more accurate results are obtained with other quadrature schemes, such as subdividing $[0, 1]$ into a large number of subintervals and applying three-point Gaussian quadrature on each subinterval. The Hopf function in Section 5.3 was obtained using twelve points chosen in this way with four subintervals; its accuracy is about 5.5 significant figures.

When this discrete set of angles is used for the transfer equation it takes this form

$$\pm\mu_i \frac{dI_i^{\pm}}{d\tau} = I_i^{\pm} - S. \tag{11.79}$$

For a fully computational approach to this problem this equation is then written in finite-difference form leading to a linear system of equations for $I_i^{\pm}$ at a set of discrete depths τ_j, which are solved in a standard way. Chandrasekhar's method does the analysis of the differential equations analytically. Equations (11.79) can be solved formally and substituted into the quadrature formula defining S. The result is exactly Milne's first integral equation, except that the E_1 kernel has been

replaced according to

$$\frac{1}{2}E_1(|x|) \to \sum_i \frac{w_i}{\mu_i} \exp\left(-\frac{|x|}{\mu_i}\right). \tag{11.80}$$

This philosophy can be applied in a much more general way: If we are faced with any convolution-type integral equation on a full-space or a half-space, we can find approximations to the kernel in the form of a sum of exponentials and proceed exactly as for the Milne problem. For example, the non-LTE problem of line scattering with complete redistribution, which leads to an integral equation with the kernel $K_1(\tau)$ as described earlier, is treated in exactly this way. This is developed at some length in the book by Ivanov.

The next step in the analysis involves finding the function $T(z)$ that approximates $1 - \tilde{K}_1(i/z)$. Since the Fourier transform of $(a/2)\exp(-a|x|)$ is $1/(1 + k^2/a^2)$, the representation of $T(z)$ in the conservative scattering ($\varpi = 1$) case is

$$T(z) = 1 - \sum_{i=1}^{n} \frac{w_i}{1 - \mu_i^2/z^2} = \sum_{i=1}^{n} \frac{w_i \mu_i^2}{\mu_i^2 - z^2}. \tag{11.81}$$

We observe that $T(z)$ must be a rational function of z since it is a sum of rational functions. It has $2n$ poles, at the points $z = \pm\mu_i$. Since apparently $T(z) \propto 1/z^2$ for $z \to \infty$, it must have the form $R_{2n-2}(z)/S_{2n}(z)$, where R and S are polynomials of the indicated degrees. Therefore $T(z)$ should have $2n - 2$ zeroes $\pm z_\alpha, \alpha = 1, \ldots, n - 1$. The computational work at this point is to actually use an algebraic root finder to evaluate (to high precision!) these roots. Once they are found $T(z)$ can be expressed in factored form as

$$T(z) = \frac{\prod_{\alpha=1}^{n-1}(1 - z^2/z_\alpha^2)}{\prod_{i=1}^{n}(1 - z^2/\mu_i^2)}, \tag{11.82}$$

where the multiplicative constant factor has been adjusted to satisfy $T(0) = 1$. Now it is simple to factor $T(z)$ into parts analytic in the left and right half-planes,

$$T(z) = \frac{1}{H(z)H(-z)}, \tag{11.83}$$

with

$$H(z) = \frac{\prod_{i=1}^{n}(1 + z/\mu_i)}{\prod_{\alpha=1}^{n-1}(1 + z/z_\alpha)}. \tag{11.84}$$

The problem is now essentially solved, because $H(\mu)$ is the angle dependence of the emergent intensity and $H(1/p)/p$ is the Laplace transform of the source function, for half-space problems. The Laplace inversion can be done easily with the method of residues since the poles of H, the points $-z_\alpha$, are already known.

There is a relation between all the roots and the quadrature points that comes from noting that $T(z) \sim -1/3z^2$ for $z \to \infty$ assuming that the quadrature formula (11.77) is accurate for the integral $\int \mu^2 d\mu = 1/3$. It is

$$\frac{\prod_{i=1}^{n} \mu_i}{\prod_{\alpha=1}^{n-1} z_\alpha} = \frac{1}{\sqrt{3}}. \tag{11.85}$$

Using this relation we see that the expansion of $H(z)$ for large z is

$$H(z) \sim \sqrt{3}(z + q_\infty), \tag{11.86}$$

where the constant q_∞ is given by

$$q_\infty = \sum_{i=1}^{n} \mu_i - \sum_{\alpha=1}^{n-1} z_\alpha. \tag{11.87}$$

Setting $p = 1/z$ thus gives the behavior of the Laplace transform of S at small p, namely

$$\tilde{S} \sim \sqrt{3} S(0) \left(\frac{1}{p^2} + \frac{q_\infty}{p} \right), \tag{11.88}$$

and therefore S for large τ is given by

$$S \sim \sqrt{3} S(0)(\tau + q_\infty). \tag{11.89}$$

In other words the large-τ value of the Hopf function comes out immediately from the solution of the characteristic equation $T(z) = 0$ for the roots z_α.

11.7 Spherical symmetry

Spherical radiative transfer is one step up in complexity from radiative transfer in slab geometry. Good reviews of spherical radiative transfer are found Mihalas (1978), p. 250ff, and Mihalas and Mihalas (1984), Section 83. For spherical problems all the scalars, such as opacities, temperature, source function, etc., are functions only of the radius (and perhaps time), but the intensities are functions of radius and μ, where μ is defined as the radial component of the direction vector $\mathbf{n}$. However, spherical coordinates are curvilinear, which means that μ varies along a

straight ray. We introduce coordinates based on the rays, which are the path length along the ray measured from the point of closest approach to the center,

$$z \equiv r\mu, \tag{11.90}$$

and the impact parameter of this ray relative to the center,

$$p \equiv r\sqrt{1-\mu^2}. \tag{11.91}$$

As a photon moves along the ray, p remains constant but r and μ vary as z increases by the distance traveled. The inverses of the relations giving z and p in terms of r and μ are

$$r = \sqrt{p^2+z^2} \tag{11.92}$$

and

$$\mu = \frac{z}{\sqrt{p^2+z^2}}. \tag{11.93}$$

The derivatives of these with respect to z at constant p give the variations of r and μ along the path:

$$\left(\frac{\partial r}{\partial z}\right)_p = \frac{z}{\sqrt{p^2+z^2}} = \mu, \tag{11.94}$$

$$\left(\frac{\partial \mu}{\partial z}\right)_p = \frac{p^2}{(p^2+z^2)^{3/2}} = \frac{1-\mu^2}{r}. \tag{11.95}$$

The correct form of the transport equation omitting velocities in (r, μ) coordinates is therefore

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mu\frac{\partial I_\nu}{\partial r} + \frac{1-\mu^2}{r}\frac{\partial I_\nu}{\partial \mu} = j_\nu - k_\nu I_\nu. \tag{11.96}$$

Forming the first two moments of the transport equation is easy, since the μ-derivative term can be integrated by parts. We find

$$\frac{\partial E_\nu}{\partial t} + \frac{\partial F_\nu}{\partial r} + \frac{2F_\nu}{r} = 4\pi j_\nu - cE_\nu, \tag{11.97}$$

$$\frac{1}{c}\frac{\partial F_\nu}{\partial t} + c\frac{\partial P_\nu}{\partial r} + c\frac{3P_\nu - E_\nu}{r} = -k_\nu F_\nu, \tag{11.98}$$

which we would have expected from the general geometry relations given earlier if we were familiar with the form of a tensor divergence in spherical symmetry.

Much work has been done with radiative transfer in spherical symmetry in the steady-state case. We will pursue this briefly by dropping the time-derivative terms.

The moment equations then become

$$\frac{1}{r^2}\frac{\partial r^2 F_\nu}{\partial r} = 4\pi j_\nu - cE_\nu \tag{11.99}$$

and

$$c\frac{\partial P_\nu}{\partial r} + c\frac{3P_\nu - E_\nu}{r} = -k_\nu F_\nu. \tag{11.100}$$

Here is where the Eddington factor

$$f_\nu \equiv \frac{P_\nu}{E_\nu} \tag{11.101}$$

can be introduced, as well as Auer's integrating factor q_ν defined by

$$\log q_\nu = \int \frac{dr}{r}\left(3 - \frac{1}{f_\nu}\right). \tag{11.102}$$

This allows the flux to be calculated from E_ν using the divergence-like formula

$$F_\nu = -\frac{c}{k_\nu q_\nu}\frac{\partial}{\partial r}\left(q_\nu f_\nu E_\nu\right). \tag{11.103}$$

The spherical version of the VEF method then proceeds using this formula for F_ν and either (11.97) or (11.99). This part of the problem is then no more costly than slab geometry.

The question remains of how to calculate the Eddington factor. One approach is to select an angle mesh μ_i as well as a radius mesh r_j and convert (11.96) or its steady-state equivalent into a set of finite difference equations. By choosing an upwind form of differencing these equations can be solved in a single sweep from the upwind side in the downwind direction. (This idea of sweeping in the upwind-to-downwind direction will be explained more below, in connection with S_N methods.) We will not discuss this more here, because the accuracy turns out to be bad, and a better method is available.

The better method is to use the (p, z) variables as the coordinates. The mesh is actually constructed by finding the intersection points of the circles $r = r_j$ with the rays $p = p_i$. The z values come out to be $z_{i,j} = \pm\sqrt{r_j^2 - p_i^2}$. The two possible signs correspond to the two directions of propagation on the ray at a given radius, but we can also think of a long ray that starts outside the star at negative z, comes inward as z increases through negative values, reaches the point of closest approach at $z = 0$, then passes out again as z increases through the positive values. Such a mesh is illustrated in Figure 11.1. The transfer equation in these coordinates is

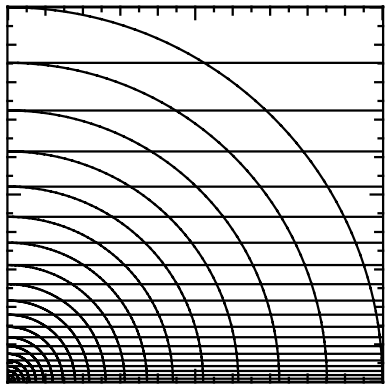

Fig. 11.1 Representation of the p, z mesh for spherical problems. Only the hemisphere $z \geq 0$ is shown.

simply

$$\left(\frac{\partial I_\nu}{\partial z}\right)_p = j_\nu - k_\nu I_\nu. \tag{11.104}$$

This is used in the auxiliary calculation for the VEF method to find I_ν given values of k_ν and j_ν. Sometimes this step is called the *formal solution*. The integration can be done by marching along each ray in the direction of increasing z. Alternatively, the Feautrier variables (see Section 5.6) $\jmath_\nu$ and $\hbar_\nu$ may be used, so that $\jmath_\nu$ obeys a two-point boundary-value problem,

$$\left(\frac{\partial}{\partial z}\right)_p \left[\frac{1}{k_\nu}\left(\frac{\partial \jmath_\nu}{\partial z}\right)_p\right] = k_\nu \jmath_\nu - j_\nu. \tag{11.105}$$

The boundary conditions are given on the symmetry plane at $z = 0$ – where $\hbar_\nu$ vanishes – and the outside radius. If there is an inner-boundary radius, then a condition there may replace the $z = 0$ condition for some rays.

However the intensity in the p, z coordinates is obtained, the angle moments are then calculated by

$$E_\nu(r_i) = \frac{2\pi}{cr_i}\int_{-r_i}^{r_i} dz_{i,j}\, I(p_i, z_{i,j}), \tag{11.106}$$

$$F_\nu(r_i) = \frac{2\pi}{r_i^2}\int_{-r_i}^{r_i} dz_{i,j}\, z_{i,j} I(p_i, z_{i,j}), \tag{11.107}$$

$$P_\nu(r_i) = \frac{2\pi}{cr_i^3}\int_{-r_i}^{r_i} dz_{i,j}\, z_{i,j}^2 I(p_i, z_{i,j}). \tag{11.108}$$

Because the mesh in $z_{i,j}$ is quite uneven, some care has to be given to making the spatial differencing of the transfer equation and the quadrature over z sufficiently accurate.

11.8 Escape probability methods

The use of the single-flight escape probability as a replacement for solving the equation of radiative transfer has the status of a numerical technique, since it enables solving coupled radiation hydrodynamics problems, or non-LTE problems with large numbers of level populations and radiative transitions that would be prohibitively expensive to solve with more detailed methods. At the simplest level these might be 0-D atomic kinetic equations, perhaps coupled with an energy equation for the evolution of the temperature. The "golden rule" of using single-flight escape probabilities in these situations was described earlier (see Section 9.3), and justified by Irons's theorem:

$$z \to p_{\rm esc}. \tag{11.109}$$

The quantity z is the net radiative bracket, $z = 1 - \bar{J}/S$, and $p_{\rm esc}$ is the two-sided single-flight escape probability. For a spectral line $p_{\rm esc}$ is given by

$$p_{\rm esc} = \frac{1}{2}\left[K_2(\tau) + K_2(\tau_0 - \tau)\right] \tag{11.110}$$

in terms of the kernel function defined by (9.18). Irons's theorem says that z and $p_{\rm esc}$ are equal in the mean sense. So the results should not be *too* bad in a 0-D or "one-zone" model if the mean value of $p_{\rm esc}$ is used to make the replacement $\bar{J} \to (1 - p_{\rm esc})S$ in the kinetic equations, and also to approximate the term in the material energy equation

$$4\pi \int_0^\infty d\nu\, k_\nu (J_\nu - S_\nu) \approx -4\pi \sum_{\rm lines} k_L\, p_{\rm esc} S. \tag{11.111}$$

The results are much less satisfactory if z is replaced point-by-point with $p_{\rm esc}$ in a spatially-distributed model in a hydrodynamic simulation, for example. Even in the 0-D case, the approximation of the radiation field in photoionization continua using escape probabilities is not as accurate as for lines, and far from satisfactory.

An exception to the statement about escape probabilities not being accurate point-by-point may be high-velocity flows for which the Sobolev approximation (see Section 6.8) is valid. The Sobolev approximation may give results at the 10% level of accuracy while for static media the accuracy of the normal escape probability approximation is a factor 2 in good cases. However, Hummer and

Rybicki (1982) have given a rather pessimistic assessment of the accuracy of the simple Sobolev approximation in certain cases.

Another class of methods that has been proposed by Athay (1972b), Frisch and Frisch (1975) and Canfield, Puetter, and Ricchiazzi (1981), may fill the gap where point-by-point results of reasonable accuracy are needed but it is prohibitively expensive to solve all the transfer equations. It is reviewed by Rybicki (1984), and also by Athay, Frisch, and Canfield in the same volume. Athay calls his method *probabilistic radiative transfer*, the term also used by Canfield, while Rybicki prefers to call the method *second order escape probability*. Second order in Rybicki's nomenclature means that the method is derived from a quadratic integral formula rather than from a linear one, not that it is the second member in a systematic expansion. Neither term quite suggests what the method is without further explanation. The essence of the method is to obtain a first order differential equation for $\bar{J}$ (or z) as a function of optical depth in which the single-flight escape probability appears as a coefficient. This can be solved much more cheaply than doing detailed transfer calculations because there are no frequencies or angles to consider, and the equation is first order not second.

The derivation by Canfield *et al*. is the following. There is a quadratic exact integral of the Milne equation that was suggested by Frisch and Frisch; it is

$$\int_\sigma^\infty d\tau\, \frac{\partial \bar{J}}{\partial \tau} S(\tau) = \int_\sigma^\infty d\tau\, S(\tau) \frac{\partial}{\partial \tau} \int_\sigma^\infty d\tau'\, K_1(|\tau - \tau'|) S(\tau') + \frac{1}{2} S_\infty^2. \tag{11.112}$$

On the approximation that $S(\tau)$ is slowly varying on the scale of the width of the kernel K_1, the two factors of S can be factored out of the integrals on the right-hand side. What results is

$$\int_\sigma^\infty d\tau\, \frac{\partial \bar{J}}{\partial \tau} S(\tau) = -\frac{1}{2} S(\sigma)^2 [1 - K_2(\sigma)] + \frac{1}{2} S_\infty^2. \tag{11.113}$$

The K_2 function can be identified as $2p_{\rm esc}$ for the semi-infinite medium on the basis of (11.110). Notice that this escape probability is two-sided. Differentiating the equation and dividing by S leads to

$$\frac{\partial \bar{J}}{\partial \tau} = \frac{\partial S}{\partial \tau} - \frac{\partial p_{\rm esc}}{\partial \tau} S - 2p_{\rm esc} \frac{\partial S}{\partial \tau}. \tag{11.114}$$

This is the basic equation of the probabilistic radiative transfer/second order escape probability method. An equivalent equation was given by Frisch and Frisch (1975). Athay's (1972a) form is somewhat different. As noted by Rybicki, the integral of this relation, assuming $\bar{J}(\infty) = S_\infty$, gives

$$\bar{J}(\tau) = (1 - p_{\rm esc}) S(\tau) + \int_\tau^\infty d\tau'\, p_{\rm esc} \frac{\partial S}{\partial \tau'}. \tag{11.115}$$

Another way of writing the same equation is

$$z = p_{\rm esc} - \frac{1}{S}\int_{\tau}^{\infty} d\tau'\, p_{\rm esc}\frac{\partial S}{\partial \tau'}. \tag{11.116}$$

The first term in this relation is the ordinary escape-probability approximation. The second term is responsible for the improved accuracy of the second-order approximation.

If the relation $S = (1-\epsilon)\bar{J} + \epsilon B$ for the two-level atom is used to eliminate $\bar{J}$ from (11.114), assuming B is constant, then S must obey this equation:

$$(1-\epsilon)^{-1}\frac{\partial S}{\partial \tau} = \frac{\partial S}{\partial \tau} - \frac{\partial p_{\rm esc}}{\partial \tau}S - 2p_{\rm esc}\frac{\partial S}{\partial \tau}. \tag{11.117}$$

The integral of this equation that gives $S = B$ for $p_{\rm esc} = 0$ is

$$S = \frac{\sqrt{\epsilon}B}{\sqrt{\epsilon + (1-\epsilon)2p_{\rm esc}}}, \tag{11.118}$$

in exact agreement with Ivanov's approximation (9.26) for a semi-infinite slab ($\tau_0 \to \infty$).

This method has proved to be very useful for things like modeling quasar broad-emission-line spectra (Canfield *et al.*,1981). Canfield, McClymont, and Puetter (1984) describe applications of the method as well as an extension to finite slabs.

11.9 S_N methods

At this point we would like to distinguish long-characteristic methods from short-characteristic methods for solving the transfer equation. A long-characteristic method means that we march along a single straight ray to solve the equation, although the ray direction may be changing in terms of components of **n** along the local coordinate directions. An example of this was just seen in the p, z coordinates for spherical-symmetry problems. A short characteristic method is one in which a bundle of rays is created at each mesh point, each one of which goes in the direction of a certain **n** with respect to the local coordinates. The rays in this bundle are extended in the upwind direction only as far as the next spatial cell. Each spatial node has its own bundle, and these do not connect from one node to the next for two reasons: the ray directions are not parallel with the vectors joining neighboring nodes, and the ray directions at one node are not necessarily parallel to those at the neighboring node. The name S_N is often used for short-characteristic methods. The slab geometry case is an illustration of both long- and short-characteristic methods, because in this case the ray segments from all the nodes do join into long rays. For other geometries this is not true.

The S_N method is developing very rapidly at this time. Pomraning (1973) describes this method at an early state. The problems associated with the unhappy choice between inaccuracy (step differencing) and negative solutions (typified by the diamond-difference method) have vanished today, through the use of the discontinuous finite-element method (e.g., Dykema, Klein, and Castor (1996)) and the new corner-balance method of Adams (1997) and Castrianni and Adams (1998). The state of the S_N methods in 2002, including fast iterative solutions of implicit scattering problems, is reviewed by Adams and Larsen (2002).

One example of short characteristics has been encountered already, the spherical equation in r, μ coordinates. The S_N approach regards this as an advection problem in r, μ space. The equation can actually be cast into conservative form as

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\mu I_\nu\right) + \frac{\partial}{\partial \mu}\left(\frac{1-\mu^2}{r} I_\nu\right) = j_\nu - k_\nu I_\nu. \tag{11.119}$$

One approach to forming the S_N equations is to integrate this equation over a radial volume element $[r_i, r_{i+1}]$ and an angle element $[\mu_j, \mu_{j+1}]$. The "surface fluxes", $r^2\mu I$ in radius and $(1-\mu^2)I/r$ in μ, for that radius–angle cell are represented as interpolated values along the mesh lines, where preference is given to the values on the side of the mesh line corresponding to the cell from which the radiation is flowing. This is the "upwind differencing" idea. It should be familiar from the discussion of cell-centered advection in the Eulerian hydrodynamics methods, Section 3.2. The fluxes can be made first-order accurate, which makes the method second-order accurate, by doing the interpolation appropriately.

Another way to do it is to expand the intensity in a set of basis functions for each r, μ cell, such as the four functions needed to represent I with bilinear interpolation. Substituting this expansion for I into the transport equation yields a residual function that ideally would be zero everywhere, but of course will not be in practice. By taking projections of the residual on the basis functions (or perhaps another set) we derive enough equations to determine the unknown expansion coefficients. This is the finite element method. By allowing every cell to have its own set of expansion coefficients independent of its neighbors, i.e., by not enforcing continuity of I between neighbor cells, the number of degrees of freedom is increased and with this the ability to represent sharp changes in the intensity is improved. The "upwinding" enters in this version of the finite element method when the surface terms that arise from the integration over a cell are systematically evaluated using the variables on the upwind side of the cell boundary. This is the discontinuous finite element method.

Assuming that upwind differencing has been applied, the equation(s) for a given r, μ cell couple in the values for the neighbor cell at smaller r (if $\mu > 0$) or larger r (if $\mu < 0$) and at smaller μ. That means that it is possible to do a raster scan of

the whole mesh in the proper order and find that all the neighbor-cell data that are needed at each point have already been computed. Thus one scan through the mesh is sufficient to evaluate all the intensities. This is what we mean by "sweeping" the mesh.

The radiation transport method in ZEUS-2D, an axial-symmetry 2-D Eulerian radiation hydrodynamics code (Stone and Norman, 1992a,b; Stone, Mihalas, and Norman, 1992), illustrates some of the short-characteristics ideas in two space dimensions. The characteristics of the transport equation in rz geometry are hyperbolae opening in the r direction, which means that either the angle advection is treated separately from spatial differencing, as just discussed, or the curved paths must be tracked. In ZEUS-2D the problem is solved by using the axial-symmetry extension of the pz coordinate system used for spherical geometry, as in Section 11.7. Tangent planes parallel to the z axis take the place of the rays with impact parameter p in the spherical case. There are as many tangent planes as there are zones in the r direction in the mesh. The slices by these tangent planes through the hydrodynamic mesh define the transport mesh in each plane. The number of cells in the z direction is the same as in the hydrodynamic mesh, while the number in the lateral direction varies from twice the number of radial zones to two, depending on the distance between the slice and the z axis. The transport on each tangent plane is computed for several values of n_z, the direction cosine of the ray with respect to the z axis. So using this transformation reduces the axial symmetry problem to one of transport in xy geometry with a Cartesian mesh.

The short characteristics method of Stone *et al.* (1992) applied in ZEUS-2D has features in common with the Mihalas, Auer, and Mihalas (1978) (MAM) method, and especially with the method of Kunasz and Auer (1988). The salient features of MAM are the following. The intensity is described by pointwise values located at the mesh nodes. For each of the chosen angles a ray is drawn through a given node O in that direction, both upwind and downwind, and it intersects the far sides of the neighboring zones at an upwind point M and a downwind point P. The transport equation is written in Feautrier form, see Section 5.6, along this characteristic, and Auer's (1976) Hermite differencing of the Feautrier equation is used, which gives a fourth-order accurate relation connecting the Feautrier values j and the source functions S at the three points M, O, P. Points M and P are not at nodes, so both j and S at these points are represented by quadratic interpolation of the values on the appropriate sides of the nine-node stencil surrounding O. The result is a nine-point differencing of the Feautrier form of the transport equation for this ray direction. Taken all together, for a given direction, the result is a block-tri-diagonal linear system for the unknowns j. Since the source functions may depend on all the js through the scattering term $\bar{J}$, the Rybicki elimination method (Rybicki, 1971) is applied. The downsides of this approach are two: (1) The block-tri-diagonal

system is very expensive to solve. This is unavoidable with the Feautrier form. (2) The quadratic interpolations will produce ringing and can, and often do, yield negative intensities.

Kunasz and Auer (1988) depart from MAM by using the first order form of the equation of transfer – rather than the Feautrier form used by MAM – which is integrated (exactly) with the relation

$$I_O = I_M \exp(-\Delta\tau) + \int_0^{\Delta\tau} d\tau' \, S(\tau') \exp[-(\Delta\tau - \tau')], \qquad (11.120)$$

in which $S(\tau')$ must be represented by an interpolation function. They consider the alternatives of using linear interpolation of S between points M and O, or using parabolic interpolation based on the three points M, O, P. The values of S at points M and P may be given by linear or parabolic interpolation also. The parabolic interpolations for the upwind point M will sometimes make use of data one point beyond the nine-point stencil on the upwind face of the box defined by the nine points. This results in a total of 13 points on which data may be used to obtain I_O. Unlike the case with Feautrier differencing, the intensities can be calculated in a downwind sweep in the Kunasz and Auer method. This makes the operation count scale linearly with the product $N_x N_z$, where N_x and N_z are the number of mesh lines in those directions, whereas the MAM method scales as $N_x^3 N_z^2$. There is a heavy penalty for using the Feautrier variables in two or three dimension. The computational results in Kunasz and Auer (1988) indicate an unfortunate trade-off between accuracy and positivity in problems with discontinuous data, such as the searchlight beam. They point out that more typical problems are forgiving in this respect.

The differences in ZEUS-2D with respect to Kunasz and Auer (1988) are: linear interpolation replaces quadratic interpolation for the values at point M, and S is represented by linear interpolation along the ray. The solution of the transfer equation for the segment MO, given by (11.120), becomes

$$I_O = I_M \exp(-\Delta\tau) + (S_O - S_M) \exp(-\Delta\tau) + \frac{S_O - S_M}{\Delta\tau}[1 - \exp(-\Delta\tau)]. \qquad (11.121)$$

This differencing is first-order accurate, and it is not consistent with the diffusion limit. That is, the relation $J_\nu \approx S_\nu - \nabla \cdot (\nabla S_\nu / k_\nu)/(3k_\nu)$ will not be obeyed in the optically thick limit, although $\mathbf{F}_\nu \approx -\nabla S_\nu/(3k_\nu)$ may be. The scheme is also not conservative. These objections are addressed by using the transport solution only for the purpose of obtaining the Eddington tensor, see Section 11.5. The Eddington tensor is incorporated in a conservative cell-centered differencing of the radiation energy equation.

Two approaches to S_N radiation transport that do not use short-characteristic finite differences are the upstream corner-balance method (UCB) described by Adams (1997) and the nonlinear corner-balance method (NLCB) of Castrianni and Adams (1998). The Adams (1997) method is quite similar to the bilinear discontinuous finite element (BLD) method used by Dykema *et al.* (1996); see also Castor, Dykema, and Klein (1991). Adams (1997) reviews a variety of different methods. A general feature of these methods is to retain second-order accuracy while preserving positivity as much as possible. A very general result is that second-order accuracy and strict positivity (I can never be negative whatever the source function is, provided $S \geq 0$) are mutually exclusive in a linear algorithm. Nonlinear algorithms can have simultaneous second order accuracy and positivity, and indeed NLCB does.

A sample of how the BLD method works is the 1-D problem. Let us say we want to solve

$$\frac{\partial I}{\partial \tau} = -I + S \tag{11.122}$$

on a mesh with nodes $\tau_{1/2}, \tau_{3/2}, \ldots$, so that zone i is bounded by $\tau_{i-1/2}$ and $\tau_{i+1/2}$. We focus on zone i, and define $x = (\tau - \tau_{i-1/2})/(\tau_{i+1/2} - \tau_{i-1/2})$. For the linear discontinuous method the intensity in zone i is represented by

$$I = I_i^- + (I_i^+ - I_i^-)x. \tag{11.123}$$

That is, the value at $\tau_{i-1/2}$ is I_i^- and the value at $\tau_{i+1/2}$ is I_i^+. These variables are all independent, so at each node $i + 1/2$ the intensity is double-valued, having the value I_i^+ on the left and the value I_{i+1}^- on the right. The interpolation for I can be described by saying that we expand in a set of two basis functions; the basis function associated with the left-hand node is $w_0 = 1 - x$, and the function associated with the right-hand node is $w_1 = x$. The Galerkin prescription for the finite-element method is to substitute (11.123) into (11.122) and then form the projections on the two basis functions. There is upwinding built into this procedure: the integration over the interval $[\tau_{i-1/2}, \tau_{i+1/2}]$ is extended at its lower limit (upwind side) infinitesimally into the interval $[\tau_{i-3/2}, \tau_{i-1/2}]$. This brings in the value I_{i-1}^+ when $\partial I/\partial \tau$ is integrated. Carrying out these operations leads to the following equation, which is repeated for $k = 0, 1$:

$$\frac{1}{\Delta\tau}\left[(I_i^- - I_{i-1}^+)w_k(0) + (I_i^+ - I_i^-)\int_0^1 w_k(x)\,dx\right]$$
$$= (S_i^- - I_i^-)\int_0^1 w_k(x)(1-x)\,dx + (S_i^+ - I_i^+)\int_0^1 w_k(x)x\,dx. \tag{11.124}$$

It is perfectly possible to use this pair of equations as-is, but stability and positivity are improved by making a modification that is called "mass lumping". This consists of replacing $S_i^+ - I_i^+$ on the right-hand side of the $k = 0$ equation by $S_i^- - I_i^-$, and replacing $S_i^- - I_i^-$ on the right-hand side of the $k = 1$ equation by $S_i^+ - I_i^+$. The final result is

$$\frac{1}{\Delta\tau}\left[I_i^- - I_{i-1}^+ + \frac{1}{2}(I_i^+ - I_i^-)\right] = \frac{1}{2}(S_i^- - I_i^-), \tag{11.125}$$

$$\frac{1}{\Delta\tau}\left[\frac{1}{2}(I_i^+ - I_i^-)\right] = \frac{1}{2}(S_i^+ - I_i^+). \tag{11.126}$$

The solution of this pair of equations is

$$I_i^- = \left(1 + \Delta\tau + \frac{1}{2}\Delta\tau^2\right)^{-1}\left[I_{i-1}^+ + \Delta\tau\left(I_{i-1}^+ + \frac{1}{2}(S_i^- - S_i^+)\right) + \frac{1}{2}\Delta\tau^2 S_i^-\right], \tag{11.127}$$

$$I_i^+ = \left(1 + \Delta\tau + \frac{1}{2}\Delta\tau^2\right)^{-1}\left[I_{i-1}^+ + \frac{1}{2}\Delta\tau(S_i^- + S_i^+) + \frac{1}{2}\Delta\tau^2 S_i^+\right]. \tag{11.128}$$

The second equation can be solved recursively for all the I_i^+, from which the I_i^- follow using the first equation. We can see from the form of the equations that it is quite possible that this linear discontinuous method is second order accurate, and indeed it is. The average $I_i = (I_{i-1}^+ + I_i^-)/2$ is the adopted nodal value. This quantity is not guaranteed to be positive. If the source function increases dramatically in zone i, then I_i^- can become negative, and therefore so can the nodal average.

Adams (1997) describes first the simple corner balance (SCB) method. This introduces the idea of a "corner", which is a subdivision of a mesh cell obtained in this way: Define the center of the cell in some way, such as by averaging the coordinates of the vertices. Then in two dimensions draw lines from the center to the midpoints of all the sides of the cell, which can be an arbitrary polygon. These lines then divide the cell into corners, with each corner containing one of the vertices. In three dimensions the cell is sliced up by planes that connect the midpoints of two edges and the center of the cell, but these may be further divided into tetrahedra, depending on the method. In one dimension there are two "corners" per cell, the left half and the right half. With quadrilateral cells in two dimensions there are four corners per cell, and in a hexahedral mesh in three dimensions there are eight corners per cell. There can be 48 tetrahedra per hex, which is quite an obstacle to using tets, despite their attractive simplicity. The concept underlying the corner-balance methods is that the intensity is regarded as constant in each corner. The transport equation is treated by applying conservative finite-volume differencing

on each corner. The node- or edge- or face-centered fluxes must be specified, and the usual S_N choices are: (1) diamond, which means that the edge flux is derived from the average intensity of the cells on each side; and (2) step, which means that the upwind intensity is used. Diamond is second order with ringing and the possibility of negative intensities, while step is first order and positive. In the corner-balance method the edges or faces of a corner can be either exterior, meaning that the adjacent corner belongs to another cell, or interior, meaning that the adjacent corner is in the same cell. The SCB method applies step to exterior edges and diamond to interior edges.

As pointed out by Adams (1997), for slabs and for Cartesian meshes in two dimensions, the equations of the SCB method are identical to the fully mass-lumped linear discontinuous method, i.e., to (11.125) and (11.126) in one dimension. There are some drawbacks of the SCB method. The first is that if the mesh is distorted in two or three dimensions (and SCB is no longer the same as lumped BLD in that case) then the wrong effective diffusion coefficient is obtained in the optically-thick limit. The boundary condition in the optically-thick limit may not be the proper diffusion boundary condition, depending on the cell geometry. The final drawback is that a linear system solve is required to obtain the corner intensities for all the corners in the cell; this is 2×2 in one dimension, see (11.125) and (11.126), but becomes a much more costly 8×8 in three dimensions. Adams addresses these problems with his UCB method.

The essence of Adams's UCB modification is to replace the diamond choice for the interior faces of the corners with an upwind expression that is not step, but a form that is itself derived by considering the optically-thick limit. Since it is upwind, this means that all the corners in the cell are upwind-differenced and can be solved in sequence, thus avoiding the linear system for all the corners of SCB. The numerical results shown by Adams (1997) illustrate the robustness but poor performance in some cases of SCB, the accuracy in most cases of non-mass-lumped BLD but the negativity it gives in a problem with complicated geometry, and the just-right behavior of UCB.

In both spherical geometry and 2-D axial-symmetry geometry the angle-derivative question must be faced, as discussed above. In spherical geometry the polar angle $\cos^{-1}\mu$ decreases along the ray in the direction of propagation. In axial-symmetry geometry the inclination of the ray to the symmetry axis z is constant, but the azimuthal angle of the ray with respect to the radial direction decreases along the ray. The relations are formally the same in the two cases, apart from a factor $(1 - n_z^2)^{1/2}$ to project path length into the x–y plane in the cylindrical case. The current S_N methods such as UCB and BLD do not apply a finite-volume method in angle, as described earlier, but represent the angle derivative in finite-

difference form. Some of the background of this question is found in Chapter 9 of Richtmyer and Morton (1967).

This is illustrated for axial symmetry as follows. Let the direction vector have a component μ in the $+z$ direction, and let the angle between the ray's projection on the x–y plane and the radial direction be ϕ. (In much of the transport literature ϕ is used for the scalar flux, i.e., our $4\pi\bar{J}$, and ψ is used for the angular flux, our I, but that should not be a confusion here.) The impact parameter of the ray, a constant quantity, is $p = \sqrt{1-\mu^2}r\sin\phi$. Therefore $d\phi/dr$ along the ray is $d\phi/dr = -\tan\phi/r$. This means the transfer equation becomes

$$\mu\frac{\partial I}{\partial z} + \sqrt{1-\mu^2}\cos\phi\,\frac{\partial I}{\partial r} - \frac{\sqrt{1-\mu^2}}{r}\sin\phi\,\frac{\partial I}{\partial\phi} = -kI + kS. \quad (11.129)$$

This can also be written in conservative form as

$$\frac{\partial}{\partial z}(\mu I) + \frac{1}{r}\frac{\partial}{\partial r}\left(\sqrt{1-\mu^2}r\cos\phi I\right) - \frac{1}{r}\frac{\partial}{\partial\phi}\left(\sqrt{1-\mu^2}\sin\phi I\right) = -kI + kS. \quad (11.130)$$

It is the differencing of the last term on the left-hand side that is the question. In S_N schemes μ and ϕ are assigned specific values, and there will be, in general, several direction vectors with different values of ϕ for each value of μ; see below. If these ϕ values for the discrete directions are denoted by $0 < \phi_1 < \phi_2 < \cdots < \phi_K < \pi$, then we can define a mesh of cells in ϕ by $\phi_{k+1/2} \approx (\phi_k + \phi_{k+1})/2$ plus $\phi_{1/2} = 0$ and $\phi_{K+1/2} = \pi$. The average of the term in question in the transfer equation over the interval $[\phi_{k-1/2}, \phi_{k+1/2}]$ is given by

$$\frac{\sqrt{1-\mu^2}}{r\,\Delta\phi_k}\left[(\sin\phi I)_{k-1/2} - (\sin\phi I)_{k+1/2}\right], \quad (11.131)$$

with $\Delta\phi_k = \phi_{k+1/2} - \phi_{k-1/2}$. The flux-like quantity $(\sin\phi I)_{k+1/2}$ apparently vanishes at the ends, $k = 0$ and $k = K$. The question is, what value to assign points that are interior to the range? As is frequently the case in transport theory, the two common choices are step and diamond. With diamond differencing, the cell-center intensity is assumed to be the average of the cell-edge values on either side, $I_k = (I_{k-1/2} + I_{k+1/2})/2$; this is written as $I_{k-1/2} = 2I_k - I_{k+1/2}$ and used to evaluate the cell-edge flux at $k - 1/2$. The calculation procedure begins with a special starting calculation at $\phi = \pi$, for which the ϕ-flux vanishes, and this is used to provide $I_{K+1/2}$. The intensities at smaller values of k and ϕ are then found recursively. With step differencing the replacement $I_{k+1/2} = I_{k+1}$ is made. Diamond has been the choice in the traditional S_N method, *viz.,* Carlson (1963). Further information is available in Lewis and Miller (1984).

The average of the second term on the left-hand side of the transfer equation over $[\phi_{k-1/2}, \phi_{k+1/2}]$ leads to a coefficient $\sqrt{1-\mu^2}\langle\cos\phi\rangle_k$ that is identified with

one of the specified direction cosines of the angle set; likewise, $\Delta\phi_k$ must be the azimuthal angle factor in the angular quadrature weight. Both these quantities are determined for a given quadrature set (see the next section). In the case that the intensity is precisely uniform and isotropic the curvature effects that arise from the second and third terms in the transfer equation must exactly cancel, which means that the effective coefficients $(\sin\phi)_{k+1/2}$ must satisfy this recursion relation:

$$(\sin\phi)_{k+1/2} - (\sin\phi)_{k-1/2} = \Delta\phi_k \langle\cos\phi\rangle_k. \tag{11.132}$$

This also is discussed by Lewis and Miller (1984). Morel and Montry (1984) point out that the curvature terms do not quite cancel out in forming the diffusion limit of the S_N equations with either step or diamond differencing, and this causes anomalous dips in the solution for $r \to 0$. They propose a weighted-diamond differencing in which instead of a 50–50 average $I_k = (I_{k-1/2} + I_{k+1/2})/2$, a weighted average is used with weights that depend on the mis-centering of $\cos\phi_k$ in $[\langle\cos\phi_{k-1/2}\rangle, \langle\cos\phi_{k+1/2}\rangle]$. This has proved to be quite successful.

It is often required to solve the S_N equations with a source function that includes scattering, either true scattering or with an effective source term in which thermal emission has been linearized and expressed in terms of the absorbed radiation. Source iteration (lambda iteration, Jacobi iteration) involves estimating the source function, solving the S_N equations for the intensities, using these to evaluate the scattering term(s) and thus obtain a new source function, and then repeating to convergence. As we have said before and will repeat again, this kind of iteration can be very slow to converge. The preconditioning methods that are often employed to speed up convergence are: (1) Eddington tensor (see Section 11.5), (2) diffusion synthetic acceleration, and (3) transport synthetic acceleration (TSA). The Eddington tensor method is not exactly an acceleration scheme, since in this method the material coupling is to the radiation moments obtained from the moment-equation solution, not to the S_N intensities. The Eddington tensor method, coupled with a BLD S_N scheme to obtain the tensor, is described in Klein *et al*. (1989). The coupled tensor diffusion equations describing a line scattering problem are solved by Klein *et al*. using another iterative method: the double splitting iteration. In this method two preconditioners are used, one after the other, on each iteration. The first preconditioner includes all nonlocal spatial coupling, but lags the frequency coupling; this is equivalent to Lambda iteration. The second preconditioner does the reverse; it includes all the frequency coupling but lags the spatial coupling. This is Jacobi iteration in the spatial sense. The double splitting iteration is found to perform well. Solving the 2-D tensor moment problem posed by the first preconditioner requires one of the solvers discussed in Section 11.4. Klein *et al*. use ORTHOMIN. The DSA method is based in the work of Kopp (1963) and later developed by Reed (1971) and especially Alcouffe (1976). In DSA a suitably defined

diffusion operator is used to pre-condition the scattering terms in S_N, in just the way to be described in Section 11.11. It is found with some of the S_N schemes, particularly if the mesh is distorted, that the diffusion operator fails as a preconditioner; i.e., the spectral radius of the preconditioned iteration matrix is too large, and the iterations converge poorly or diverge. In such cases a helpful approach is to use a consistently-differenced S_N method of much lower order, perhaps S_2, as a preconditioner. This leads to the TSA method described by Ramoné, Adams, and Nowak (1997). In some perverse cases even this method may fail without user adjustments. This may be expected to be the case if the problem at hand is afflicted with ray effects (see the next section). In such cases the available angle sets are simply inadequate to describe the solution, and different angle sets will differ markedly from each other; it is no surprise if the effectiveness of the acceleration is poor when this happens.

It should be obvious that there is a lot of computational engineering that goes into doing these calculations, certainly in two and three dimensions, and the interested reader should consult the literature to learn the current state of the art.

11.10 What are the angles? The bad news

The nomenclature of S_N and in some cases the actual values of the angles come from the early neutron transport work, summarized by Carlson (1963), with later improvements summarized in Carlson (1970). In this work Carlson explains how the direction vectors, which correspond to certain points on the unit sphere, can be chosen to obey some important integral constraints, and especially the symmetry requirement of being unchanged under permutations of the three coordinate directions X, Y, Z. This symmetry requirement means that the pattern of the directions in the first octant, $x > 0$, $y > 0$, $z > 0$, must have three-fold symmetry: a rotation about the direction (1, 1, 1) by 120° should take the pattern into itself. The additional assumption that the directions should be arranged in rows at constant latitude, i.e., of constant direction cosine with respect to any one of the three coordinate axes, means that in the simplest case the directions should resemble the graph of a triangle number. The triangle numbers are $1, 3, 6, 10, 15, \ldots, k(k+1)/2, \ldots$. These correspond to: rows of one direction; rows of one and two directions; rows of one, two, and three directions; and so on. For the kth triangle array there will be k different positive direction cosines with respect to one of the axes in the first octant. Taking into account the reflected directions that have negative values of the direction cosine means that the kth triangle array gives $N = 2k$ different direction cosines. This is the definition of N. The number of directions in one octant is therefore $k(k+1)/2 = N(N+2)/8$. We observe that N is always an even number in this nomenclature; N is the number of rows (i.e., the number of polar angles) in

two octants. The triangle-number angle sets are Carlson's Set A. He also describes Set B, which differs from Set A in that the three vertices of the triangle are clipped off. With $N/2$ still taken for the number of surviving rows, the total number of directions per octant becomes $(N+8)(N-2)/8$. For a given $N \geq 6$ there are somewhat more angles in Set B than in Set A. The results for Sets A and B are tabulated by Carlson for N up to 8, and with some difficulty the general relations can be worked out.

In later work, Lathrop and Carlson (1965) and Carlson (1970) derived other quadrature sets that, unlike Sets A and B, exactly integrate certain higher-degree polynomials in the components of the direction vector. This is important when calculating transport using a highly-anisotropic scattering phase function, which is very often expanded in Legendre polynomials of the cosine of the scattering angle. The Carlson (1970) sets are called by the name "level-symmetric quadrature," and denoted symbolically by LQ_N. These have the triangle-number shape, with $N(N+2)/8$ directions per octant. The LQ_N quadrature integrates exactly all even polynomials in n_x, n_y, n_z up to degree $N-2$. The Lathrop and Carlson (1965) quadratures make the additional assumption that the quadrature weight for a given direction is the sum of three weights associated with the three direction cosines (as in Carlson (1963)) with the additional requirement that even polynomials in n_x up to the Nth degree be integrated exactly.

How many octants are needed to describe the radiation in a given problem? The symmetry of the radiation field, at a general location in space, is not as high as the spatial symmetry of the problem itself, since a symmetry element of the spatial structure does not correspond to a symmetry element of the radiation field at this particular location unless the location lies *on* the element. So, for example, a spherically-symmetric spatial structure has symmetry elements of the full rotation group, but only the rotations about an axis that passes through the center of symmetry and the point of observation are symmetries of the radiation field. The symmetry group of the radiation field for a spherical problem turns out to be $C_{\infty v}$ in Schoenflies notation. Thus the radiation field is axially symmetric, but the full range of polar angles from $-\pi$ to π is required. The same group and the same angles apply to 1-D slab geometry. But the symmetry is less for 1-D cylindrical geometry, the geometry of an infinite cylinder, since in that case the only symmetry elements of the spatial structure that leave the point of observation fixed are the reflections in a horizontal plane and a vertical plane, and a rotation about the radial direction by π, plus combinations of these. In other words, the group of the radiation field is C_{2v}. The radiation field is 2-D, but only two octants are needed to describe it: one with $n_r > 0$ and one with $n_r < 0$. That means a total of $N(N+2)/4$ directions for the typical S_N angle set. In the 2-D spatial geometries, xy and rz, there is only a single symmetry element that leaves the point of

observation fixed, namely a reflection, corresponding to the group C_{1h}. The symmetry plane is the x–y plane in xy geometry, and the plane through the z-axis and the observation point in rz geometry. Thus four of the eight octants are required to describe the radiation field in either of the 2-D geometries, with $N(N+2)/2$ directions in all. If the spatial structure of the problem has any less symmetry than the ones so far mentioned, then the radiation field has no symmetries whatsoever at the general point of observation, and all eight octants are essential, with $N(N+2)$ directions.

The problem of finding an adequate angle set in two or three dimensions is much more severe than in one dimension. Recall that by using a total of 36 directions it was possible to get the 1-D Hopf function to something like six significant figures. The sad truth is that it is not at all hard to imagine a 2-D problem for which 36 directions will not even give one significant figure in the result.

Imagine that we are calculating the transport of hydrogen line and free–free emission in the solar corona, and that the source is a solar flare, where a large of amount of thermal energy has been deposited in a region perhaps only tens of kilometers in size. We want to calculate the UV radiation field that results on the opposite side of a supergranulation cell that is 30 000 km across. The clever solar physicist would do a hand calculation to get this answer, but what happens if the code is asked to do it?

The angle subtended by the flare-heated region at the observation point is less than (100 km)/(3×10^4 km), or 1/300 rad = 0.2 degree. If our trusty code is using S_N and has chosen an angle set that can accommodate the unexpected flare wherever it might occur, then the number of angle points it will have to use must be $4\pi/(1/300)^2$, which is 10^6! This might seem ridiculous, but it is true. If, say, a mere few hundred directions are used, which are spaced by about 9 degrees on the sphere, then as we move away from the flare the local flux as we compute it will fall off as $1/r^2$ until we are about 700 km away, and after that the results rapidly become worse. If we decide to walk away along one of the ray directions of our discrete set then we will find that the flux does not fall off very rapidly at all beyond 700 km. If, on the other hand, we walk away in a direction that falls between the rays the flux falls off much faster than $1/r^2$. By the time we are 3×10^4 km away there is a disagreement of several orders of magnitude between the flux in the ray directions and the directions in between.

The general name for this difficulty with S_N calculations is called "ray effects." There is no easy fix for this problem. It goes away once the angle set is sufficiently dense to resolve all the features that the solution contains.

A concept that could help is the idea of adaptive directions. Suppose the angle set at each spatial point is made dynamic, and at each time it moves toward the directions that would resolve the "important" features in the radiation field. This

might do it, but it is not very easy to implement. How do you define "important"? Is this based on the gross magnitude of the intensity? What happens if you are interested in a critical feature that is not very large in magnitude? What happens when there are events in opposite hemispheres, both of which call for attention? What happens when there is a sudden onset, as in a flare? The directions may start wheeling toward the proper direction, but not get to where they are useful in resolving the flare until it is over. Or if the dynamic response is made brisk, the directions may chatter in a dreadful way and destroy the accuracy of the calculation. Finally, adaptive directions can end up making it impossible to perform the S_N sweep just discussed, which implies a large penalty in computing cost.

The summary of the angle crisis is as follows. Determine the smallest linear size of the important spatial inhomogeneities of absorptivity or emissivity for your problem. You would like to be able to resolve in angle the radiation produced by this object. Then decide on the largest viewing distance you really care about; this is no larger than the diameter of the problem, but it is also no larger than the longest mean free path for radiation you care about. If you view from a greater distance than that, you cannot see the object anyway. Divide the size by the distance and convert that to an angle. If your angle set cannot resolve that, then you are fooling yourself when you say that you are calculating radiation transport. You might just as well use diffusion; the answers would be no worse, and much cheaper to compute.

11.11 Implicit solutions – acceleration

The objective in this section is to describe preconditioning methods for radiative transfer. A general name that applies to many of these is accelerated lambda iteration (ALI), and this will be defined below. But first let us review the list of variables and equations we need to solve for the radiation hydrodynamics problem. There are the mass and momentum density and the corresponding conservation equations, which we may or may not choose to treat by operator splitting. There is the material temperature and its corresponding internal energy equation (or perhaps the total energy equation), which we are rather sure we cannot split. There are all the intensities that are needed to describe the radiation field for us. Depending on our needs these may be as few as a single frequency-integrated energy density, or as many as there are combinations of dozens to hundreds of frequencies with tens to thousands of angles. The intensities are surely coupled implicitly to the material temperature. If the problem is non-LTE then the level populations are another large set of unknowns that are locally coupled to the radiation field. That is, they enter the problem in a way similar to the temperature. For all except the smallest of these

problems the Jacobian matrix of the set of nonlinear equations is too large for a direct solution. The point of the acceleration methods is to solve the linearized system iteratively, and the heart of the iterative methods is the way of preconditioning the iteration.

To fix the ideas consider a steady-state scattering problem with gray radiation and a constant albedo $\varpi < 1$, in other words the second Milne problem. The language introduced many years ago for the operation of acting upon the source function with $E_1(|\tau' - \tau|)/2$ is "applying the lambda operator". (See Kourganoff (1963).) In other words we define a linear operator Λ in this way:

$$\Lambda_\tau[S] \equiv \frac{1}{2}\int_0^\infty d\tau'\, E_1(|\tau' - \tau|)S(\tau'). \tag{11.133}$$

In terms of Λ, Milne's second problem is

$$S = (1 - \varpi)B + \varpi\,\Lambda[S]. \tag{11.134}$$

The name lambda has become attached to various iterative methods for solving the Milne problem. The first to consider is lambda iteration. This is the following operation applied repeatedly to convergence:

$$S^{n+1} = (1 - \varpi)B + \varpi\,\Lambda[S^n]. \tag{11.135}$$

This will indeed converge since $\varpi < 1$ and also the smallest eigenvalue λ of Λ is unity (for a full-space or half-space problem) or larger. The eigenvalue is defined as a value for which the equation

$$u = \lambda\Lambda[u] \tag{11.136}$$

has an admissible solution. This can take a very large number of iterations to converge. A physical picture of the iteration count is the following. Each step of the iteration is like letting a photon have one flight. Imagine releasing photons throughout the problem and tracking them through flight after flight until they all have either been quenched, through the destruction probability $1 - \varpi$, or have escaped. This number of flights will be roughly whichever is less of $1/(1 - \varpi)$ and $\mathcal{T}^2$, where $\mathcal{T}^2$ is the total optical thickness of the atmosphere. This is usually unacceptably large.

In linear algebra language lambda iteration is called Jacobi iteration. We can generalize our thinking about Milne's second problem by saying that S stands for any of the variables that describe the material, such as density, velocity, temperature or level populations, and the relation that gives S in terms of J stands for the hydrodynamic conservation laws and atomic kinetics equations that determine these material variables in terms of the radiation field. The lambda operator stands for the transport equation that determines the radiation field in terms of the mate-

rial variables. Milne's second equation is thus a stand-in for the statement that all the material properties are self-consistent with the radiation they determine.

The answer to the question, what do you do when Jacobi iteration is too slow, is precondition. Here is how to precondition. We use our inventiveness and find a cheap but accurate operator Λ^* to approximate Λ. We suppose that after n steps we have an approximation S^n, but we are going to get the exact answer on the $(n+1)$th step. That will be the case if

$$\begin{aligned} S^{n+1} - S^n &= (1-\varpi)B + \varpi\Lambda[S^{n+1}] - S^n \\ &= (1-\varpi)B + \varpi\Lambda[S^{n+1} - S^n] + \varpi\Lambda[S^n] - S^n \\ &= \varpi\Lambda^*[S^{n+1} - S^n] + (1-\varpi)B + \varpi\Lambda[S^n] - S^n \\ &\quad + \varpi(\Lambda - \Lambda^*)[S^{n+1} - S^n]. \end{aligned} \tag{11.137}$$

We get our preconditioned iteration by neglecting the last term in the last equality on the grounds that it is a small operator applied to a small difference. Thus the accelerated iteration is

$$(1 - \varpi\Lambda^*)[S^{n+1} - S^n] = (1-\varpi)B + \varpi\Lambda[S^n] - S^n. \tag{11.138}$$

The quantity on the right-hand side is the residual in the Milne equation after n iterations. The corresponding formula for $\bar{J}$ itself is

$$\bar{J}^{n+1} = \Lambda^*[S^{n+1}] + \bar{J}^n - \Lambda^*[S^n]. \tag{11.139}$$

Without the factor involving Λ^* on the left-hand side, the correction to S^n is just set equal to the residual, which is Jacobi iteration. Thus this factor is what provides the acceleration. As we see from the derivation, if the approximate lambda operator is accurate, then convergence is immediate. The name for methods like this is ALI or approximate operator iteration (AOI), depending on the author. The ALI methods have been described in many places. A place to start is Kalkofen (1987).

The amplification factor for this iteration, i.e., the factor by which the error is multiplied each time, depends on the eigenvalues of this operator:

$$(1 - \varpi\Lambda^*)^{-1}\varpi(\Lambda - \Lambda^*). \tag{11.140}$$

We call the acceleration gentle if the operator $\varpi\Lambda^*$ is small in some sense, and aggressive if it is close to the unit operator. Since the operator $\varpi\Lambda$ itself is close to the unit operator in those situations in which we most need acceleration, the acceleration has to be aggressive to do any good. Too aggressive is bad, however. If $\varpi\Lambda^*$ is more than half-way from $\varpi\Lambda$ to the unit operator the accelerated iteration diverges.

The origin of the methods in this class is the work of Cannon (1973). From 1981–1986 this was picked up and extended by many others, including:

Scharmer (1981); Scharmer and Carlsson (1985); Werner and Husfeld (1985); Olson *et al.* (1986); Hamann (1985, 1986); Rybicki (1984); and Olson and Kunasz (1987). The variations are considerable, but a common theme is to let Λ^* be either a diagonal operator or one that couples nearest neighbors. The most "aggressive" diagonal that does not produce instability turns out to be approximately this:

$$\Lambda^* \approx 1 - \bar{p}_{\rm esc}(\text{zone}), \tag{11.141}$$

where $\bar{p}_{\rm esc}(\text{zone})$ is the zone-average single-flight escape probability from the given zone to any of its neighbors. The escape probability can either be calculated from expressions involving E_2 functions, integrals over line profiles corrected for the velocity field, etc., or obtained by just computing the lambda matrix in detail using the S_N equations or what you will, and discarding everything but the diagonal of it. In fact, the diagonal can be found by doing only a few local calculations so the wasted effort on the off-diagonal parts need not be done. (See Rybicki and Hummer (1991), Appendix B.) If $\bar{p}_{\rm esc}(\text{zone})$ is overestimated compared with the ideal value then the iteration becomes sluggish. If it is underestimated by a factor 2 the iteration will diverge. Thus a somewhat careful calculation of it is indicated; see Olson and Kunasz (1987).

The approximation (11.141) has an interesting consequence when it is used in the non-LTE rate equations. Equation (11.139) for $\bar{J}$, which determines the photoabsorption rate, becomes

$$\bar{J}^{n+1} \approx [1 - \bar{p}_{\rm esc}(\text{zone})]S^{n+1} + \bar{J}^n - [1 - \bar{p}_{\rm esc}(\text{zone})]S^n. \tag{11.142}$$

The net photoabsorption rate is calculated from

$$\mathcal{R}_{21} = N_2(A_{21} + B_{21}\bar{J}^{n+1}) - N_1 B_{12}\bar{J}^{n+1} \tag{11.143}$$

but we can simplify this expression by identifying S^{n+1} with the value computed from N_1 and N_2 using (9.6). Doing this yields

$$\mathcal{R}_{21} = N_2 A_{21}\bar{p}_{\rm esc}(\text{zone}) - (N_1 B_{12} - N_2 B_{21})\{\bar{J}^n - [1 - \bar{p}_{\rm esc}(\text{zone})]S^n\}. \tag{11.144}$$

We can pick off the coefficients of N_1 and N_2 on the right-hand side to be the effective radiative rate coefficients that are put into the kinetic equations. This approach has been used by Rybicki and Hummer (1991, 1992), and non-LTE calculations of supernova spectra using their method have been made by Hauschildt and colleagues (Hauschildt, Storzer, and Baron, 1994; Hauschildt, Baron, and Allard, 1997; Baron and Hauschildt, 1998).

The several non-LTE stellar atmosphere codes built on the ALI principle that were in existence in 1990 were reviewed by Hummer and Hubeny (1991). There is a technical point about these codes that is of interest in the present context. It is that many of the codes substitute the ALI approximation (11.139) into the rate equations, but then take into account the dependence of the coefficients in Λ^* on the level populations, thus forming a nonlinear system for the populations. In these methods (e.g., Werner (1987) and Carlsson (1986)), there is a double iteration loop, with outer ALI iterations and inner Newton–Raphson iterations. The Rybicki and Hummer (1991, 1992) modification lags the Λ^* coefficients, which makes the rate equations linear and therefore no Newton–Raphson iteration is necessary. The cost is additional ALI iterations, which in Rybicki and Hummer (1991, 1992) are minimized by using Ng (1974) acceleration. Dreizler and Werner (1991) describe a double-loop method that minimizes Newton–Raphson cost by using a quasi-Newton method due to Broyden (1965), in which the inverse Jacobian $J^{-1} = (\partial F/\partial x)^{-1}$ is approximated by a matrix B^{-1} that is formed recursively by adding on at each iteration a rank-1 matrix derived from Δx, ΔF, and the previous B^{-1}. This quasi-Newton method then has the flavor of the Newton–Krylov method described earlier. The TLUSTY code of Hubeny and Lanz (1995) represents a hybrid of ALI and traditional linearization, in that the radiative transitions can be treated with or without ALI at the user's option. A double iteration with inner Newton–Raphson iterations is used. These authors remark that they have found the approach of lagging the approximate lambda operator to fail for some problems

The VEF method (see Section 11.5) is another acceleration technique. In this case the approximate operator is the tensor diffusion operator, which should be quite accurate except that it may not reflect the changes that occur during the time step. We may suppose that one application of the tensor diffusion operator is sufficient to correct the estimated mean intensity instead of the tens of iterations that may be required with a diagonal approximate operator; however, the cost of a diffusion solution is much greater. The trade-off between cost per iteration and the iteration count may favor one method or the other in different problems. Alcouffe's diffusion synthetic acceleration method is a variation of this, but here the tensor is omitted and the unmodified Eddington operator is used as the accelerator (Alcouffe, 1976).

The implicit coupling of multifrequency radiation transport to the material energy equation is included within the general ALI framework as was hinted earlier. Equation (11.142) can be put not only into the kinetic equations, but into the internal energy equation as well, and used to derive the correction to the material temperature. No matrices dimensioned by frequency need to be solved in this procedure. Other types of acceleration can be used for the material energy equation,

however. The multifrequency gray method is one such. The frequency-integrated implicit coupling equations are used with mean opacities based on the accurate multifrequency spectral distributions instead of the Planck and Rosseland means. The spectral distributions and the mean opacities are updated in the formal transfer part of the iteration cycle. This approach was used by Castor (1974a), and is discussed briefly by Pinto and Eastman (2000).

In general, there is now a well-filled storehouse of preconditioning methods to allow the solution of radiation hydrodynamic problems with large dimensions without directly solving any huge linear systems.

11.12 Monte Carlo methods

The Monte Carlo method for solving a linear transport equation like

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mathbf{n}\cdot\nabla I_\nu = j_\nu - k_\nu I_\nu \tag{11.145}$$

is really quite simple. We sample the distribution j_ν to create new particles in various zones with various frequencies and directions. We may also sample the boundary sources, if there are any. Then each particle is tracked through the problem until it leaves across a boundary or is destroyed. Every time step of the hydrodynamic problem each of the particles is tracked from zone to zone until the time for that step is used up, assuming the particle survives that long. When the particle is tracked through a particular zone, the optical thickness across the zone along the track is computed and used to sample whether the particle is destroyed in crossing the zone or not. At the end of the cycle the count of particles in each zone is an estimator of the energy density for that zone. Another, possibly less noisy, estimator is the sum of the track lengths for all the particles that crossed that zone during the time step. The effects of the fluid velocity, i.e., the Doppler and aberration effects, are easily taken into account. The particles are tagged with their fixed-frame frequency, but when a particle is tracked in a given zone it is transformed into the fluid frame for that zone to compute the probability of material interactions. When the emissivity is sampled, the fluid frame emissivity is used to get the frequency, and the direction is sampled, then the transformations are applied to get the fixed-frame values that the particle will carry. Compton scattering is included in full generality by sampling a relativistic electron velocity distribution and the Klein–Nishina cross section to determine whether a scattering event will occur in a given zone or not, and if it does, then the relativistic kinematics of the scattering process can be applied to find the new frequency and direction after scattering. In short, all these awkward processes can be accounted for in full, accurate detail.

A bit of concern arises with the need to implicitly couple the Monte Carlo radiation to the material temperature. The trick that was introduced by Fleck (see Fleck and Cummings (1971) and Fleck and Canfield (1984)) is to linearize the material energy equation and eliminate the temperature perturbation between that linearized equation and the linearization of the emissivity function. This produces an effective emissivity that is linear in the photoabsorption rate, much like the scattering source function in Milne's second problem or in the discussion of ALI methods. This is the "effective scattering" concept: absorption followed by thermal emission is treated like a scattering event. The frequency after emission is changed, however; it is resampled fom the thermal emission spectrum. It has been found by Larsen and Mercier (1987) that the Fleck–Cummings effective scattering formulation is inaccurate when the time step is longer than the radiative cooling time, which is when the implicit method is most needed. Some steps toward correcting this problem have been made by Ahrens and Larsen (2000), and by Martin and Brown (2001).

Can the "effective scattering" process be applied to non-LTE problems? This works perfectly well for resonance line scattering. Indeed, some of the most complete studies of line scattering with angle-dependent partial redistribution combined with fluid flow have been made this way. The failure point is when the particles must interact with excited atoms whose absorbing population is itself sensitive to the Monte Carlo estimate of a resonance line intensity. The combination of the noise and the nonlinear process produces large errors.

There are so many positives about the Monte Carlo method that it might seem surprising that any other method is used. The answer, of course, is the wildly exorbitant cost of doing such calculations. Here is one naive estimate of that cost. Let us say that we will be happy with our statistics if we can bin all the particles present on a given time step according to the zone they are in, the frequency they have, and the direction they are going, with perhaps 10^8 bins in all, not to be too greedy. For 1% statistics in every bin we would need 10^4 particles per bin, or 10^{12} in all. Now every particle crosses quite a few zones in a time step, and some dozens of floating point operations are needed for each zone that is crossed. So let us say 10^2–10^3 operations are needed per particle per time step. That means about 10^{14}–10^{15} operations in all per time step. What would it cost to do this calculation deterministically, using S_N, for instance? We have one intensity variable per photon bin, and we have to perform a few operations per variable per iteration cycle per time step. If 10^2 of the implicit coupling iterations are needed, then the work is 10^2–10^3 operations per variable per time step, or something like 10^{10}–10^{11} total operations per time step. That is very roughly 10^4 times less work than for Monte Carlo. But, say the Monte Carlo folks, requiring 10^4 particles per bin in every single bin is vastly more than you need if the number you want to estimate

is one global quantity, such as the total power out of the problem. Sure, you need 10^4 particles in a bin whose contents are an observable on which you are focusing your attention, but who cares about all those other bins that you will not examine in detail. However, radiation hydrodynamics in a nonlinear problem, and what assurance is there that extremely poor statistics on large parts of the radiation field will not cause a serious bias in the estimate of even that one global quantity you want? This is a difficult question, and knowledge in this area is mostly empirical. Monte Carlo still lives with the mantle of being a very expensive method.

Monte Carlo methods cannot be done justice in a few lines. Some of the astrophysical applications have been to Comptonization in x-ray sources (Pozdnyakov, Sobol', and Syunyaev, 1979) and to resonance line transport in the presence of a velocity gradient (Caroff, Noerdlinger, and Scargle, 1972).