Chapter 17

Cauchy-Riemann Equations

17.1 The differential problem

As a first simple exercise in multigridding a *system* of partial differential equations we have studied the elliptic system

$$u_x + v_y = F_1 \tag{17.1a}$$

$$u_y - v_x = F_2 \tag{17.1b}$$

in a domain Ω, where $u = u(x, y)$ and $v = v(x, y)$ are the unknown functions, the subscripts denote partial derivatives, and $F_i = F_i(x, y)$ are given functions. All functions are real. The homogeneous system $Fl \equiv F2 \equiv 0$ are the usual Cauchy-Riemann equations, which express analyticity of the complex function $u + iv$.

The matrix-operator form of (17.1) is

$$L \begin{pmatrix} u \\ v \end{pmatrix} := \begin{pmatrix} \partial_x & \partial_y \\ \partial_y & -\partial_x \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}, \tag{17.2}$$

where ∂_x and ∂_y are partial derivatives with respect to x and y, respectively. The determinant of L is the Laplace operator $-\Delta = -\partial_x^2 - \partial_y^2$. Hence (17.2) or (17.1) is a second-order elliptic system and its solution is determined by one condition along the boundary $\partial\Omega$. As such a boundary condition we can, for example, require

$$(u(x, y), v(x, y))_n = G(x, y), \qquad ((x, y) \in \partial\Omega) \tag{17.3}$$

where $(u, v)_n$ denotes the component of the vector (u, v) normal to the boundary in the outward direction. From (17.1a), (17.3) and the divergence theorem (or Stokes, or Gauss, formula) we get the "compatibility condition"

$$\int_\Omega F_1 dx dy = \int_{\partial\Omega} G ds. \tag{17.4}$$

If (17.4) holds then equations (17.1) or (17.2), with the boundary condition (17.3), is a well-posed problem: A unique solution exists and depends continuously on the data F_l, F_2 and G.

17.2 Discrete Cauchy-Riemann equations

Suppose we first try to approximate (17.1) by the central difference equations

$$\frac{u^h(x+h,y)-u^h(x-h,y)}{2h}+\frac{v^h(x,y+h)-v^h(x,y-h)}{2h}=F_1^h(x,y) \quad (17.5a)$$

$$\frac{u^h(x,y+h)-u^h(x,y-h)}{2h}-\frac{v^h(x+h,y)-v^h(x-h,y)}{2h}=F_2^h(x,y). \quad (17.5b)$$

The corresponding difference operator is

$$L^h=\begin{pmatrix} \mu_x^h\partial_x^h & \mu_y^h\partial_y^h \\ \mu_y^h\partial_y^h & -\mu_x^h\partial_x^h \end{pmatrix} \quad (17.6)$$

where the averaging and differencing operators are defined by

$$\begin{aligned} \mu_x^h\Phi(x,y) &:= \tfrac{1}{2}\left[\Phi(x+\tfrac{h}{2},y)+\Phi(x-\tfrac{h}{2},y)\right], \\ \mu_y^h\Phi(x,y) &:= \tfrac{1}{2}\left[\Phi(x,y+\tfrac{h}{2})+\Phi(x,y-\tfrac{h}{2})\right], \\ \partial_x^h\Phi(x,y) &:= \tfrac{1}{h}\left[\Phi(x+\tfrac{h}{2},y)-\Phi(x-\tfrac{h}{2},y)\right], \\ \partial_y^h\Phi(x,y) &:= \tfrac{1}{h}\left[\Phi(x,y+\tfrac{h}{2})-\Phi(x,y-\tfrac{h}{2})\right], \end{aligned} \quad (17.7)$$

hence

$$\mu_x^h\partial_x^h=\partial_x^{2h}, \qquad \mu_y^h\partial_y^h=\partial_y^{2h}$$

and

$$\det(L^h)=-(\mu_x^h\partial_x^h)^2-(\mu_y^h\partial_y^h)^2=-\Delta^{2h}$$

with the symbol (see §2.1)

$$\tilde{\det}\,L^h(\theta_1,\theta_2)=\frac{\sin^2(\theta_1)+\sin^2(\theta_2)}{h^2}. \quad (17.8)$$

This operator is not h-elliptic, since $\tilde{L}^h(\pi,0)=\tilde{L}^h(0,\pi)=\tilde{L}^h(\pi,\pi)=0$.

Indeed, the homogeneous (17.5) equations ($F_1^h\equiv F_2^h\equiv 0$) have the oscillatory solutions

$$u^h(\alpha h,\beta h)=C_0+C_1(-1)^\alpha+C_2(-1)^\beta+C_3(-1)^{\alpha+\beta} \quad (17.9a)$$

$$v^h(\alpha h,\beta h)=C_4+C_5(-1)^\alpha+C_6(-1)^\beta+C_7(-1)^{\alpha+\beta} \quad (17.9b)$$

which corresponds to nothing similar in the solution of the differential equation. Note, however, that solutions like (17.9) vanish *on the average*, i.e.,

$M^h u^h = M^h v^h = 0$ for a suitable local averaging operator M^h, such as $M^h = \mu_x^h \mu_y^h$ or $M^h = (\mu_x^h \mu_y^h)^2$. Hence, the solutions of (17.5) will be good solutions on the average. Such difference operators are called *quasi-elliptic* [BD79, §3.4]. See further remarks in §17.6.

Let us now construct an h-elliptic approximation L^h to (17.1). If the equations are to have the form

$$D_x^1 u^h + D_y^2 v^h = F_1^h \tag{17.10a}$$

$$D_y^3 u^h - D_x^4 v^h = F_2^h \tag{17.10b}$$

where D_x^j and D_y^j are some difference approximations to ∂_x and ∂_y, then $\det(L^h) = -D_x^1 D_x^4 - D_y^2 D_y^3$ should be an elliptic approximation to the Laplace operator $-\Delta$. The simplest such operator is the five-point operator which is obtained by taking either

$$D_x^1 = D_x^4 = \partial_x^h, \qquad D_y^2 = D_y^3 = \partial_y^h, \tag{17.11}$$

or one-sided differences such as

$$D_x^1 = \partial_x^{\mathrm{F}}, \quad D_y^2 = \partial_y^{\mathrm{F}}, \quad D_y^3 = \partial_y^{\mathrm{B}}, \quad D_x^4 = \partial_x^{\mathrm{B}} \tag{17.12}$$

where $\partial_x^{\mathrm{F}} := \mu^h \partial^h - \frac{h}{2} \partial^h \partial^h$ and $\partial_x^{\mathrm{B}} := \mu^h \partial^h + \frac{h}{2} \partial^h \partial^h$. Approximations like (17.12) do not give a central approximation to (17.1), and their truncation error is therefore $O(h)$. We thus prefer to use (17.11). This we can do only by using staggered grids for u^h and v^h. The grid we use and the positioning of the discrete variables are shown in Fig. 17.1.

With this positioning we can indeed approximate (17.1) by

$$\partial_x^h u^h + \partial_y^h v^h = F_1^h \quad \text{at cell centers } ① \tag{17.13a}$$

$$\partial_y^h u^h - \partial_x^h v^h = F_2^h \quad \text{at interior vertices } ② \tag{17.13b}$$

and the symbol is that of the 5-point Laplacian, namely,

$$\tilde{\det} L^h(\theta_1, \theta_2) = \frac{4}{h^2} \left(\sin^2 \frac{\theta_1}{2} + \sin^2 \frac{\theta_2}{2} \right). \tag{17.14}$$

This symbol vanishes only for $\theta_1 \equiv \theta_2 \equiv 0 (\mathrm{mod} 2\pi)$. Thus (17.13) is an elliptic (even R-elliptic – see [BD79, §3.6]) difference system.

For simplicity we consider here domains with boundaries along grid lines. It is then simple to discretize the boundary condition (17.3). On each boundary link (the heavy lines in Fig. 17.1) the variable (u, v) is already defined at the center of the link, so (17.3) is discretized to

$$(u^h, v^h)_n = G^h \text{ at midpoints of boundary links.} \tag{17.15}$$

Summing (17.13a) over all the cells of our domain we get the compatibility condition

$$\sum_{\text{cell centers}} F_1^h(x, y) = \sum_{\text{boundary midpoints}} G^h(x, y) \tag{17.16}$$

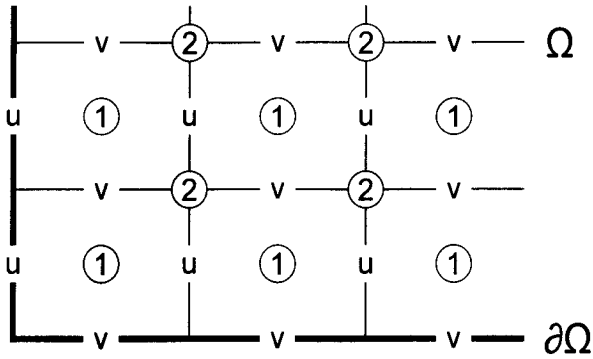

Figure 17.1. ***Discretization of Cauchy-Riemann equations.*** *A typical part of the grid is shown. The discrete unknown functions u^h and v^h and their computed approximations $\tilde{u}^h$ and $\tilde{v}^h$ (u and v in the figure) are defined at the centers of vertical and horizontal links, respectively. The first equation (17.13a) is centered at cell centers, where its right-hand side, F_1^h, is defined, and where ① is shown in the figure. The second equation, (17.13b), is centered, and F_h^2 is defined, at the grid vertices, as shown by ② in the figure.*

which is the discrete analog of (17.4).

Theorem 17.1. *If (17.16) holds, then the discrete Cauchy-Riemann equations (17.13) with the boundary conditions (17.15) have a unique solution.*

Indeed, the total number of equations (17.13), (17.15) equals the total number of cells and vertices in the grid. The number of discrete unknowns is the number of links. Hence, by a well-known formula of Euler, there is one more equation than unknowns. But the equations are dependent, as we saw in constructing the compatibility condition (17.16). Hence, if (17.16) holds, we can remove an equation and have the same number of equations as unknowns. It is therefore enough to prove the theorem for the homogeneous case $F_1^h \equiv 0$, $F_2^h \equiv 0$, $G^h \equiv 0$. In this case (17.13a) implies the existence of a discrete "stream function" ψ^h, defined at the vertices of the grid, such that $u^h = \partial_y^h \psi^h$, $v^h = -\partial_x^h \psi^h$. The homogeneous (17.13b) yields $\Delta^h \psi^h \equiv 0$, and the homogeneous x (17.15) implies that ψ^h along the boundary vertices is constant. Hence, by the maximum principle, ψ^h is constant everywhere. Thus, in the homogeneous case $u^h \equiv 0$ and $v^h \equiv 0$, which is what we had to show.

17.3 DGS relaxation and its smoothing rate

Most relaxation schemes are based on one-to-one correspondence between equations and unknowns: The basic relaxation step is to satisfy (or over-satisfy, or under-satisfy) one of the discrete equations by changing the corresponding unknown (or satisfy a block of equations by changing the corresponding block of unknowns). Such one-to-one correspondence is not always natural. In our case, it is clear already in the differential equations (17.1) that it would be unnatural to regard (17.1a), say, as the equation corresponding to the unknown u, and (17.1b) as the one corresponding to v. The entire system corresponds to (u, v). In the difference equations it would be impossible to have even a one-to-one correspondence between pairs of equations and pairs of unknowns, since the number of unknowns is one less than the number of equations.

We will therefore use "distributive relaxation", i.e., a relaxation scheme that satisfies each discrete equation in its turn by distributing changes to *several* unknowns, in a natural manner.

To derive a natural distributive scheme we note that neither (17.13a) nor (17.13b) are elliptic equations by themselves. It is their combination together which is elliptic. Hence, in relaxing (17.13a), for example, we should take (17.13b) into account. The simplest way to do it, which is also a special case of the general prescription described in §3.7, is to relax (17.13a) in such a way that equations (17.13b) are not "damaged", i.e., in a way which preserves the residuals of (17.13b). We do this by simultaneously changing four unknowns, in the following way:

Let $(\tilde{u}^h, \tilde{v}^h)$ be the current approximation to (u^h, v^h). Let (x, y) be the cell center where we next wish to relax (17.13a), and let

$$r_1^h = F_1^h - \partial_x^h \tilde{u}^h - \partial_y^h \tilde{v}^h \tag{17.17}$$

be the "dynamic residual" at (x, y). That is, r is the residual at (x, y) just before relaxing there. The relaxation step of (17.13a) at (x, y) is made up of the following four changes:

$$\begin{array}{rcl} \tilde{u}^h(x+\frac{h}{2}, y) & \leftarrow & \tilde{u}^h(x+\frac{h}{2}, y) + \delta \\ \tilde{u}^h(x-\frac{h}{2}, y) & \leftarrow & \tilde{u}^h(x-\frac{h}{2}, y) - \delta \\ \tilde{v}^h(x, y+\frac{h}{2}) & \leftarrow & \tilde{v}^h(x, y+\frac{h}{2}) + \delta \\ \tilde{v}^h(x, y-\frac{h}{2}) & \leftarrow & \tilde{v}^h(x, y-\frac{h}{2}) - \delta, \end{array} \tag{17.18}$$

where

$$\delta = \frac{1}{4} h r_1^h. \tag{17.19}$$

It is easy to check that the distribution of changes (17.18) is such that the residuals

$$r_2^h = F_2^h - \partial_y^h \tilde{u}^h + \partial_x^h \tilde{v}^h \tag{17.20}$$

at all neighboring vertices are not changed, whatever the value of δ. The choice (17.19) for the *ghost unknown* δ is made so that *after* the changes

the residual $r_1^h(x,y)$ will vanish. This is in the manner of the Gauss-Seidel relaxation , where old values are replaced by new values so as to satisfy one difference equation. Such schemes are therefore called *Distributive Gauss-Seidel* (DGS) schemes. In case k of the four values changed in (17.18) are boundary values ($k = 1$ near boundaries, except near corners), then no such change should be introduced in those values, and (17.19) is replaced by

$$\delta = \frac{1}{4-k} h r_1^h. \tag{17.21}$$

The relaxation of (17.13b) is made in a similar manner. If (x,y) is the vertex to be relaxed, the relaxation step will include the changes

$$\begin{array}{rcl} \tilde{u}^h(x,y+\frac{h}{2}) & \leftarrow & \tilde{u}^h(x,y+\frac{h}{2}) + \delta \\ \tilde{u}^h(x,y-\frac{h}{2}) & \leftarrow & \tilde{u}^h(x,y-\frac{h}{2}) - \delta \\ \tilde{v}^h(x+\frac{h}{2},y) & \leftarrow & \tilde{v}^h(x+\frac{h}{2},y) - \delta \\ \tilde{v}^h(x-\frac{h}{2},y) & \leftarrow & \tilde{v}^h(x-\frac{h}{2},y) + \delta, \end{array} \tag{17.22}$$

where

$$\delta = \frac{1}{4} h r_2^h. \tag{17.23}$$

The distribution (17.22) is such that the residuals r_1^h will be preserved, and δ in (17.23) is such that equation (17.13b) at (x,y) will be satisfied by the changed variables.

The above relaxation steps can be taken in various orders. In our programs, each complete relaxation sweep comprised of two passes: The first pass relaxes equation (17.13a) by (17.18)–(17.19), letting (x,y) traverse all cell centers, and the second pass similarly scans all the grid vertices, relaxing (17.13b) by (17.22)–(17.23). within each pass the best ordering of points is Red-Black (RB), although lexicographic ordering was used in the experiments [Din79]. Some operations can be saved by passing first on one color of cells and a matching color of vertices, then on the other cell color, and then on the remaining vertices.

In terms of the general formulation of §3.7 the above scheme is equivalent to introducing the ghost functions w_1^h and w_2^h, where

$$\begin{pmatrix} u^h \\ v^h \end{pmatrix} = \begin{pmatrix} \partial_x^h & \partial_y^h \\ \partial_y^h & -\partial_x^h \end{pmatrix} \begin{pmatrix} w_1^h \\ w_2^h \end{pmatrix} = M^h w^h \tag{17.24}$$

(accidentally $M^h = L^h$), and relaxing the resulting system $\Delta^h w_i^h = F_i^h$, $(i = 1,2)$, Δ^h being the 5-point Laplace operator. Hence the *smoothing factor* of this relaxation is the same as that of Δ^h, that is, for lexicographic ordering $\overline{\mu} = .5$, while in RB ordering $\overline{\mu}_1 = \overline{\mu}_2 = .25$ and $\overline{\mu}_3 = .32$ (cf. (3.2)).

17.4 Multigrid procedures

Assume now we have a sequence of grids (levels) with mesh-sizes $h_1, \ldots, h_M$, where $h_{k+1} = \frac{1}{2} h_k$. The relative position of the different grids is shown in

Fig. 17.2. Instead of F_1^h, F_2^h, G^h, u^h, v^h, $\tilde{u}^h$, $\tilde{v}^h$, r_1^h and r_2^h used above, the discrete functions at the *k-th* level will be denoted by F_1^k, F_2^k, G^k, u^k, v^k, $\tilde{u}^k$, $\tilde{v}^k$, r_1^k and r_2^k, respectively. Similarly, μ_x^k and μ_y^k will stand for $\mu_x^{h_k}$ and $\mu_y^{h_k}$.

The coarse-to-fine interpolation can be of first order, since this is the highest order of derivatives in the Cauchy-Riemann operator (see §4.3). An obvious way of doing such an interpolation (see Fig. 17.2) is

$$I_k^{k+1}\tilde{u}^k(x,y) = \begin{cases} \tilde{u}^k(x, y \pm \frac{1}{2}h_{k+1}) & \text{if } x \text{ is on a coarse-grid line} \\ \mu_x^k I_k^{k+1}\tilde{u}^k(x,y) & \text{otherwise} \end{cases} \tag{17.25}$$

and similarly for $I_k^{k+1}\tilde{v}^k(x,y)$. One can of course use linear interpolations instead.

The Cauchy-Riemann problem is linear. We can therefore make coarse-grid corrections either by the Correction Scheme (CS) or the Full Approximation Scheme (FAS, cf. §8). In the latter case we have to define the fine-to-coarse transfer of solution. We use the following averaging (cf. Fig. 17.2):

$$I_{k+1}^k\tilde{u}^{k+1}(x,y) = \mu_y^{k+1}\tilde{u}^{k+1}(x,y) \tag{17.26a}$$

$$I_{k+1}^k\tilde{v}^{k+1}(x,y) = \mu_x^{k+1}\tilde{v}^{k+1}(x,y). \tag{17.26b}$$

The fine-to-coarse transfer of residuals of the first equation (defined at cell

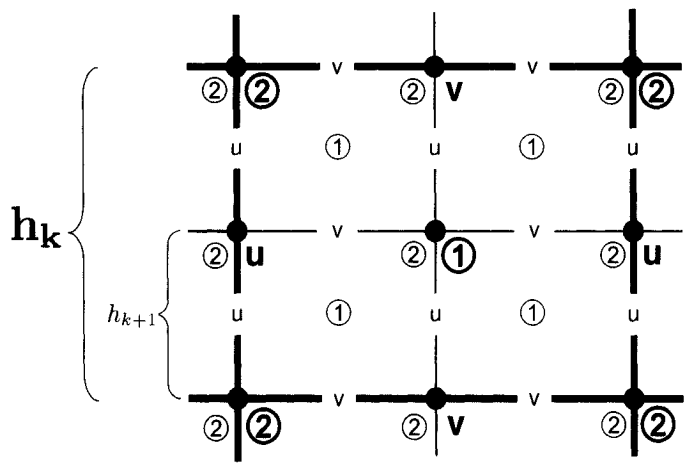

Figure 17.2. ***A coarse-grid cell divided into fine-grid cells.*** *The same notations as in Fig. 17.1, with larger and heavier type being used for the coarse grid and lighter type for the fine grid.*

centers) is also done by averaging:

$$I_{k+1}^k r_1^{k+1} = \mu_x^{k+1}\mu_y^{k+1}r_1^{k+1}. \tag{17.27}$$

When the Correction Scheme is used, (17.27) serves as the right-hand side of equation (17.13a) on the coarser level h_k. In calculating (17.27) using (17.17), observe that some terms are canceled and some of the additions need be made only once for two neighboring coarse-grid cells. It is interesting to note that when FAS is used it is not necessary to calculate (17.27). Transferring $\tilde{u}^{k+1}$ and $\tilde{v}^{k+1}$ by (17.26) and residuals by (17.27), it is easy to see that the FAS coarse-grid equation will read

$$\partial_x^k u^k + \partial_y u^k = \mu_x^{k+1}\mu_y^{k+1}F_1^{k+1}. \tag{17.28}$$

Thus, the coarse-grid equation in this case is not affected at all by the fine-grid solution: If we let $F_1^k = \mu_x^k\mu_y^k F_1^{k+1}$ in the first place, we find that (17.28) is actually identical with (17.13a) for the *k-th* level. In other words, the relative truncation error in (17.13a) vanishes.

Another feature of (17.28) is that if the compatibility condition (17.16) is satisfied on the fine grid, it will automatically be satisfied on the coarse grid too (up to round-off errors, of course). The residuals of (17.13b) can be transferred to the coarse grid either by "injection"

$$I_k^{k+1}r_2^{k+1}(x,y) = r_2^{k+1}(x,y), \tag{17.29}$$

or by the full weighting (4.6).

17.5 Numerical results

Numerical experiments with this algorithm are reported in [Din79]. They show, unsurprisingly, exactly the same convergence as in multigrid solutions for Poisson problems; e.g., a convergence factor of about .55 per RWU (relaxation work unit) when relaxation is performed lexicographically. Hence, although experiments were conducted with a cycling algorithm only, it can be safely predicted that the FMG algorithm (Fig. 1.2), with RB relaxation and $\nu_1 = \nu_2 = 1$, will solve the problem well below truncation errors.

The number of operations in such an algorithm, using the Correction Scheme, is about $40n$, where n is the number of unknowns on the finest grid. Almost all these operations are either additions or shifts (i.e., multiplications by an integer power of 2), and the algorithm is fully parallelizable.

There is a faster way of solving the discrete Cauchy-Riemann equations (17.13): Subtracting from u^h a function u_0^h which satisfies $\partial_x^h u_0^h = F^h$, a new system is obtained in which $F_1^h = O$. The problem can then be rewritten as a Poisson problem for the discrete stream function ψ^h (see §17.2). Solving that Poisson problem by a similar FMG algorithm, together with the operations of subtracting u_0^h and constructing u^h and v^h would require about $17n$ operations (additions and shifts only). The main purpose of this chapter, however, was to study methods for solving elliptic *systems*. The techniques developed for the present simple system guided us in developing the more complicated cases described in the following sections.

17.6 Remark on non-staggered grids

Any staggered-grid formulation can yield a non-staggered one (on a finer grid) simply by overlaying several staggered grids, properly shifted, on top of each other. For example, shifting equations (17.13) by the four shifts $(0,0)$, $(0,h/2)$, $(h/2,0)$ and $(h/2,h/2)$, the four systems together are equivalent to equations (17.5) for grid $h/2$. In fact, various non-staggered formulations appearing in the literature can be shown to be such interlacing of staggered formulations. They are *wasteful* in that the same accuracy is already obtained by just *one* of the interlacing subgrids, for much less work.

Also, the decomposition of the grid into interlacing subgrids locally decoupled from each other introduces *a subtle kind of instability* (typical to quasi-elliptic operators in general): Certain high-frequency modes (those which look like low frequency modes on each subgrid) are magnified in the discrete solution much beyond their size in the differential solution. This may show as large truncation errors in higher Sobolev norms. It can be corrected by averaging. Such an averaging and the multigrid solution of such a quasi-elliptic system are discussed in §18.6.