18. DISCRETIZATION AND NUMERICAL STABILITY

We leave the mathematicians' ideal world of real and complex numbers to see how the algorithms considererd in the last section can be implemented on a computer. We shall also estimate the effects of numerical errors.

Computer arithmetic

It is not possible to represent an arbitrary real number on a computer. Given a machine base β, precision t, underflow limit L, and overflow limit U, we can represent only the numbers

$$\pm \cdot d_1 \ldots d_t \times \beta^e , \quad 0 \leq d_i < \beta , \quad d_1 \neq 0 , \quad L \leq e \leq U ,$$

together with the number 0. These are known as the floating-point numbers. The value of (β,t,L,U) for Cyber 180 Model 840 is (2, 48, -4096, 4095), while for Cray-1 it is (2, 48, -16384, 8191). An arbitrary real number is 'approximately represented' by its nearest floating-point neighbour if rounded arithmetic is used; in case of a tie, it is rounded away from zero. A complex number is represented by the pair of floating-point representations of its real and imaginary parts. The errors introduced by this approximate representation while performing the arithmetic operations $+$, $-$, $\times$, $/$ are known as the round-off errors. One of the ways of reducing these errors is to carry out certain operations in higher precision, like double (2t) precision or extended (4t) precision. Taking the inner product

$$(18.1) \qquad \underset{\sim}{y}^H \underset{\sim}{x} = x(1)\overline{y(1)} + \ldots + x(n)\overline{y(n)}$$

of two n-vectors $\underset{\sim}{x}$ and $\underset{\sim}{y}$ is one such operation. In the