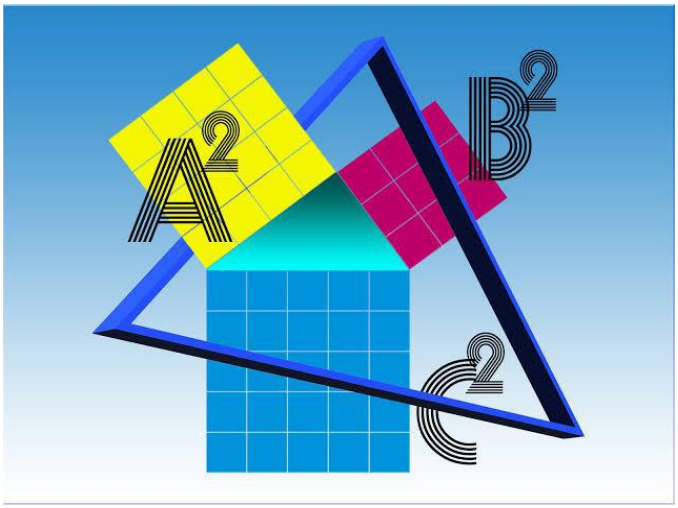
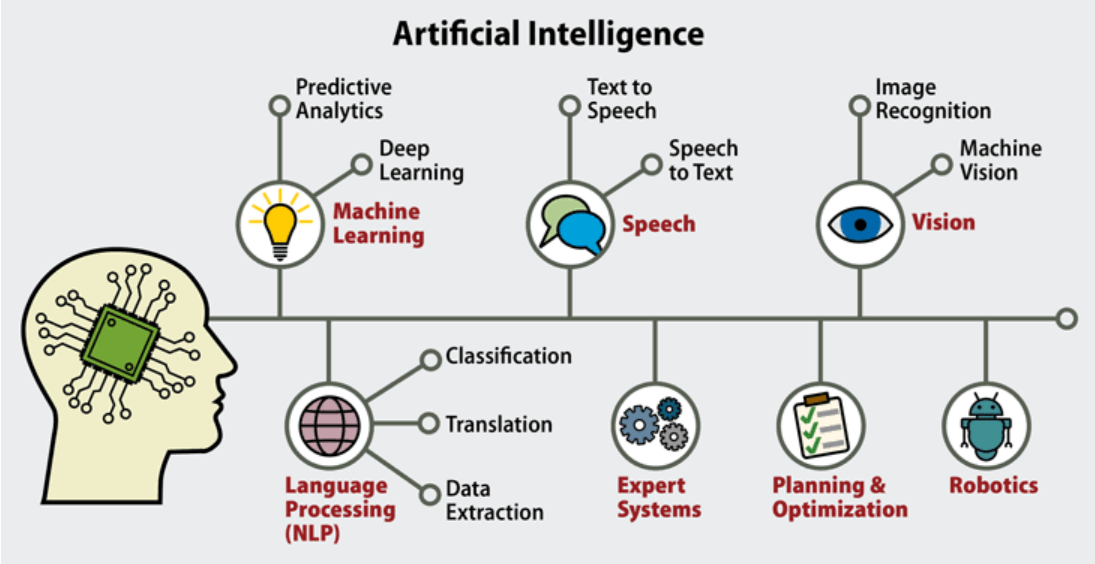
Classic Papers In Applied Mathematics



These papers provide a breadth of information that is generally useful and interesting from a Mathematical Science perspective.



Contents

- Multi-level adaptive solutions to boundary-value problems
- An algorithm for the machine calculation of complex Fourier series
- Variational methods for the solution of problems of equilibrium and vibrations
- On the partial difference equations of mathematical physics
- Integration of stiff equations
- On calculating with B-splines
- Calculating the singular values and pseudo-inverse of a matrix
- A fast algorithm for particle simulations
- Methods of conjugate gradients for solving linear systems
- Unitary triangularization of a nonsymmetric matrix

- Order stars and stability theorems
- A rapidly convergent descent method for minimization
- A new polynomial-time algorithm for linear programming
- On the Movement of Small Particles Suspended in a Stationary Liquid Demanded by the Molecular-Kinetic Theory of Heat
- A Mathematical Theory of Communication
- A New Approach to Linear Filtering and Prediction Problems
- Deterministic Nonperiodic Flow
- Non-Cooperative Games
- A Theory of the Learnable
- Simple Mathematical Models with Very Complicated Dynamics

MATHEMATICS OF COMPUTATION, VOLUME 31, NUMBER 138
APRIL 1977, PAGES 333–390

Multi-Level Adaptive Solutions to Boundary-Value Problems*

By Achi Brandt

Abstract. The boundary-value problem is discretized on several grids (or finite-element spaces) of widely different mesh sizes. Interactions between these levels enable us (i) to solve the possibly nonlinear system of n discrete equations in $O(n)$ operations ($40n$ additions and shifts for Poisson problems); (ii) to conveniently adapt the discretization (the local mesh size, local order of approximation, etc.) to the evolving solution in a nearly optimal way, obtaining "∞-order" approximations and low n, even when singularities are present. General theoretical analysis of the numerical process. Numerical experiments with linear and nonlinear, elliptic and mixed-type (transonic flow) problems—confirm theoretical predictions. Similar techniques for initial-value problems are briefly discussed.

1. Introduction. In most numerical procedures for solving partial differential equations, the analyst first discretizes the problem, choosing approximating algebraic equations on a finite-dimensional approximation space, and then devises a numerical process to (nearly) solve this huge system of discrete equations. Usually, no real interplay is allowed between discretization and solution processes. This results in enormous waste: The discretization process, being unable to predict the proper resolution and the proper order of approximation at each location, produces a mesh which is too fine. The algebraic system thus becomes unnecessarily large in size, while accuracy usually remains rather low, since local smoothness of the solution is not being properly exploited. On the other hand, the solution process fails to take advantage of the fact that the algebraic system to be solved does not stand by itself, but is actually an approximation to continuous equations, and therefore can itself be similarly approximated by other (much simpler) algebraic systems.

The purpose of the work reported here is to study how to intermix discretization and solution processes, thereby making both of them orders-of-magnitude more effective The method to be proposed is not "saturated", that is, accuracy grows indefinitely as computations proceed. The rate of convergence (overall error E as function of computational work W) is in principle of "infinite order", e.g., $E \sim \exp(-\beta^d W)$ for a d-dimensional problem which has a solution with scale-ratios $\geqslant \beta > 0$; or $E \sim \exp(-W^{1/2})$, for problems with arbitrary thin layers (see Section 9).

The basic idea of the Multi-Level Adaptive Techniques (MLAT) is to work not

Received July 22, 1976.

AMS (MOS) subject classifications (1970). Primary 65N05, 65N10, 65N20, 65N30; Secondary 35Jxx, 35M05, 39A10, 76H05, 76G15.

*The research reported here was partly supported by the Israel Commission for Basic Research. Part of the research was conducted at the Thomas J. Watson IBM Research Center, Yorktown Heights, New York. Part of it was also conducted at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, Virginia.

Copyright © 1977, American Mathematical Society

with a single grid, but with a sequence of grids ("levels") of increasing fineness, each of which may be introduced and changed in the process, and constantly interact with each other. For description purposes, it is convenient to regard this technique as composed of two main concepts:

(1) *The Multi-Grid (MG) Method for Solving Discrete Equations.* This method iteratively solves a system of discrete (finite-difference or finite-element) equations on a given grid, by constant interactions with a hierarchy of coarser grids, taking advantage of the relation between different discretizations of the same continuous problem. This method can be viewed in two complementary ways: One is to view the coarser grids as correction grids, accelerating convergence of a relaxation scheme on the finest grid by efficiently liquidating smooth error components. (See general description in Section 2 and algorithm in Section 4.) Another point of view is to regard finer grids as the correction grids, improving accuracy on coarser grids by correcting their forcing terms. The latter is a very useful point of view, making it possible to manipulate accurate solutions on coarser grids, with only infrequent "visits" to pieces of finer levels. (This is the basis for the multi-grid treatment of nonuniform grids; cf. Sections 7.2 and 7.5. The FAS mode for nonlinear problems and the adaptive procedures stem from this viewpoint.) The two seemingly different approaches actually amount to the same algorithm (in the simple case of "coextensive" levels).

The multi-grid process is very efficient: A discrete system of n equations (n points in the finest grid) is solved, to the desired accuracy, in $O(n)$ computer operations. If $\bar{P}$ parallel processors are available, the required number of computer steps is $O(n/\bar{P} + \log n)$. For example, only $40n$ additions and shifts (or $35n$ microseconds CYBER 173 CPU time) are required for solving the 5-point Poisson equation on a grid with n points (see Section 6.3). This efficiency does not depend on the shape of the domain, the form of the boundary conditions, or the mesh-size, and is not sensitive to choice of parameters. The memory area required is essentially only the minimal one, that is, the storage of the problem and the solution. In fact, if the amount of numerical data is small and only few functionals of the solution are wanted, the required memory is only $O(\log n)$, with no need for external memory (see Section 7.5).

Multi-grid algorithms are not difficult to program, if the various grids are suitably organized. We give an example (Appendix B) of a FORTRAN program, showing the typical structure, together with its computer output, showing the typical efficiency. With such an approach, the programming of any new multi-grid problem is basically reduced to the programming of a usual relaxation routine. The same is true for nonlinear problems, where no linearization is needed, due to the FAS (Full Approximation Storage) method introduced in Section 5.

Multi-grid solution times can be predicted in advance—a recipe is given and compared with numerical tests (Section 6). The basic tool is the local mode (Fourier) analysis, applied to the locally linearized-frozen difference equations, ignoring far boundaries. Such an analysis yields a very good approximation to the behavior of the high-frequency error modes, which are exactly the only significant modes in assessing the multi-grid process, since the low-frequency error modes are liquidated by coarse-grid

processing, with negligible amounts of computational work. Thus, mode analysis gives a very realistic prediction of convergence rates per unit work. (For model problems, the analysis can be made rigorous; see Appendix C.) The mode analysis can, therefore, be used to choose suitable relaxation schemes (Section 3) and suitable criteria for switching and interpolating between the grids (Appendix A). Our numerical tests ranged from simple elliptic problems to nonlinear mixed-typed (transonic flow) problems, which included hyperbolic regions and discontinuities (shocks). The results show that, as predicted by the local mode analysis, errors in all these problems are reduced by an order of magnitude (factor 10) expending computational work equivalent to 4 to 5 relaxation sweeps on the finest grid.

(2) *Adaptive Discretization.* Mesh-sizes, orders of approximation and other discretization parameters are treated as spatial variables. Using certain general internal criteria, these variables are controlled in a suboptimal way, adapting themselves to the computed solution. The criteria are devised to obtain maximum overall accuracy for a given amount of calculations; or, equivalently, minimum of calculations for given accuracy. (In practice only near-optimality should of course be attempted, otherwise the required control would become more costly than the actual computations. See Section 8.) The resulting discretization will automatically resolve thin layers (when required), refine meshes near singular points (that otherwise may "contaminate" the whole solution), exploit local smoothness of solutions (in proper scale), etc. (see Section 9).

Multi-grid processing and adaptive discretization can be used independently of each other, but their combination is very fruitful: MG is the only fast (and convenient) method to solve discrete equations on the nonuniform grids typically produced by the adaptive procedure. Its iterative character fits well into the adaptive process. The two ideas use and relate similar concepts, similar data structure, etc. In particular, an efficient and very flexible general way to construct an adaptive grid is as a sequence of uniform subgrids, the same sequence used in the multi-grid process, but where the finer levels may be confined to increasingly smaller subdomains to produce the desired local refinement. In this structure, the difference equations can be defined separately on each of the uniform subgrids, interacting with each other through the multi-grid process Thus, difference equations should only be constructed on equidistant points. This facilitates the employment of high and adaptive orders of approximation. Moreover, the finer, localized subgrids may be defined in terms of suitable local coordinates, facilitating, for example, the use of high-order approximations near pieces of boundary, with all these pieces naturally patched together by the multi-grid process (Section 7).

The presentation in this article is mainly in terms of finite-difference solutions to partial-differential boundary-value problems. The basic ideas, however, are more general applicable to integro-differential problems, functional minimization problems, etc., and to *finite-element* discretizations. The latter is briefly discussed in Sections A.5 and 7.3 and as closing remarks to Sections 8.1 and 8.3. Extensions to *initial-value problems* are discussed in Appendix D. Section 10 presents historical notes and acknowledgements.

Contents of the article:

1. Introduction

2. Multi-grid philosophy
3. Relaxation and its smoothing rate
   3.1. An example
   3.2. General results
   3.3. Acceleration by weighting
4. A multi-grid algorithm (Cycle C) for linear problems
   4.1. Indefinite problems and the size of the coarsest grid
5. The FAS (Full Approximation Storage) algorithm
6. Performance estimates and numerical tests
   6.1. Predictability
   6.2. Multi-grid rates of convergence
   6.3. Overall multi-grid computational work
   6.4. Numerical experiments: Elliptic problems
   6.5. Numerical experiments: Transonic flow problems
7. Nonuniform grids
   7.1. Organizing nonuniform grids
   7.2. The multi-grid algorithm on nonuniform grids
   7.3. Finite-element generalization
   7.4. Local transformations
   7.5. Segmental refinement
8. Adaptive discretization techniques
   8.1. Basic principles
   8.2. Continuation methods
   8.3. Practice of discretization control
   8.4. Generalizations
9. Adaptive discretization: Case studies
   9.1. Uniform-scale problems
   9.2. One-dimensional case
   9.3. Singular perturbation: Boundary layer resolution
   9.4. Singular perturbation without boundary layer resolution
   9.5. Boundary corners
   9.6. Singularities
10. Historical notes and acknowledgements
   Appendices:
A. Interpolation and stopping criteria: Analysis and rules
   A.1. Coarse-grid amplification factors
   A.2. The coarse-to-fine interpolation I_{k-1}^{k}
   A.3. Effective smoothing rate
   A.4. The fine-to-coarse weighting of residuals I_{k}^{k-1} and the coarse-grid operator L^{k-1}
   A.5. Finite-element procedures
   A.6. Criteria for slow convergence rates
   A.7. Convergence criteria on coarser grids

A.8. Convergence on the finest grid
A.9. Partial relaxation sweeps
A.10. Convergence criteria on nonuniform grids
B. Sample multi-grid program and output
C. Rigorous bound to model-problem convergence rate
D. Remarks on initial-value problems
References

2. **Multi-Grid Philosophy.** Suppose we have a set of grids $G^0, G^1, \ldots, G^M$, all approximating the same domain Ω with corresponding mesh-sizes $h_0 > h_1 > \cdots > h_M$. For simplicity one can think of the familiar uniform square grids, with the mesh-size ratio $h_{k+1} : h_k = 1 : 2$. Suppose further that a differential problem of the form

$$LU(x) = F(x) \quad \text{in } \Omega, \qquad \Lambda U(x) = \Phi(x) \quad \text{on the boundary } \partial\Omega, \tag{2.1}$$

is given. On each grid G^k this problem can be approximated by difference equations of the form

$$L^k U^k(x) = F^k(x) \quad \text{for } x \in G^k, \qquad \Lambda^k U^k(x) = \Phi^k(x) \quad \text{for } x \in \partial G^k. \tag{2.2}$$

(See example in Section 3.1.) We are interested in solving this discrete problem on the finest grid, G^M. The main idea is to exploit the fact that the discrete problem on a coarser grid, G^k, say, approximates the same differential problem and hence can be used as a certain approximation to the G^M problem. A simple use of this fact has long been made by various authors (e.g., [14]); namely, they first solved (approximately) the G^k problem, which involves an algebraic system much smaller and thus much easier to solve than the given G^M problem, and then they interpolated their solution from G^k to G^M, using the result as a first approximation in some iterative process for solving the G^M problem. A more advanced technique was to use a still coarser grid in a similar manner when solving the G^k problem, and so on. The next natural step is to ask whether we can exploit the proximity between the G^k and G^M problems not only in generating a good first approximation on G^M, but also in the process of improving the first approximation.

More specifically let u^M be an approximate solution of the G^M problem, and let

$$L^M u^M = F^M - f^M, \qquad \Lambda^M u^M = \Phi^M - \phi^M. \tag{2.3}$$

The discrepancies f^M and ϕ^M are called the residual functions, *or residuals.* Assuming for simplicity that L and Λ are linear (cf. Section 5 for the nonlinear case), the exact discrete solution is $U^M = u^M + V^M$, where the correction V^M satisfies the *residual equations*

$$L^M V^M = f^M, \qquad \Lambda^M V^M = \phi^M. \tag{2.4}$$

Can we solve this equation, to a good first approximation, again by interpolation from solutions on coarser grids? As it is, the answer is generally negative. Not every G^M problem has meaningful approximation on a coarser grid G^k. For instance, if the right-hand side f^M fluctuates rapidly on G^M, with wavelength less than $4h_M$, these fluctua-

tions are not visible on, and therefore cannot be approximate by, coarser grids. Such rapidly-fluctuating residuals f^M are exactly what we get when the approximation u^M has itself been obtained as an interpolation from a coarser-grid solution.

An effective way to damp rapid fluctuations in residuals is by usual relaxation procedures, e.g., the Gauss-Seidel relaxation (see Section 3). At the first few iterations such procedures usually seem to have fast convergence, with residuals (or corrections) rapidly decreasing from one iteration to the next, but soon after the convergence rate levels off and becomes very slow. Closer examination (see Section 3 below) shows that the convergence is fast as long as the residuals have strong fluctuations on the scale of the grid. As soon as the residuals are smoothed out, convergence slows down.

This is then exactly the point where relaxation sweeps should be discontinued and approximate solution of the (smoothed out) residual equations by coarser grids should be employed.

The Multi-Grid (MG) methods are systematic methods of mixing relaxation sweeps with approximate solution of residual equations on coarser grids. The residual equations are in turn also solved by combining relaxation sweeps with corrections through still coarser grids, etc. The coarsest grid G^0 is coarse enough to make the solution of its algebraic system inexpensive compared with, say, one relaxation sweep over the finest grid.

The following sections further explain these ideas. Section 3.1 explains, through a simple example, what is a relaxation sweep and shows that it indeed smooths out the residuals very efficiently. The smoothing rates of general difference systems are summarized in Section 3.2. A full multi-grid algorithm, composed of relaxation sweeps over the various grids with suitable interpolations in between, is then presented in Section 4. An important modification for nonlinear problems is described in Section 5 (and used later as the basic algorithm for nonuniform grids and adaptive procedures). Appendix A supplements these with suitable stopping criteria, details of the interpolation procedures and special techniques (partial relaxation).

3. Relaxation and its Smoothing Rate.

3.1. *An Example.* Suppose, for example, we are interested in solving the partial differential equation

$$LU(x, y) \equiv a\frac{\partial^2 U(x, y)}{\partial x^2} + c\frac{\partial^2 U(x, y)}{\partial y^2} = F(x, y) \tag{3.1}$$

with some suitable boundary conditions. Denoting by U^k and F^k approximations of U and F, respectively, on the grid G^k, the usual second-order discretization of (3.1) is

$$L^k U^k_{\alpha,\beta} \equiv a\frac{U^k_{\alpha+1,\beta} - 2U^k_{\alpha,\beta} + U^k_{\alpha-1,\beta}}{h_k^2} + c\frac{U^k_{\alpha,\beta+1} - 2U^k_{\alpha,\beta} + U^k_{\alpha,\beta-1}}{h_k^2} = F^k_{\alpha,\beta}, \tag{3.2}$$

where

$$U^k_{\alpha,\beta} = U^k(\alpha h_k, \beta h_k), \qquad F^k_{\alpha,\beta} = F^k(\alpha h_k, \beta h_k); \quad \alpha, \beta \text{ integers.}$$

(In the multi-grid context it is important to define the difference equations in this

divided form, without, for example, multiplying throughout by h_k^2, in order to get the proper relative scale at the different levels.) Given an approximation u to U^k, a simple example of a relaxation scheme to improve it is the following.

Gauss-Seidel Relaxation. The points (α, β) of G^k are scanned one by one in some prescribed order; e.g., lexicographic order. At each point the value $u_{\alpha,\beta}$ is replaced by a new value, $\bar{u}_{\alpha,\beta}$, such that Eq. (3.2) at that point is satisfied. That is, $\bar{u}_{\alpha,\beta}$ satisfies

$$a\frac{u_{\alpha+1,\beta} - 2\bar{u}_{\alpha,\beta} + \bar{u}_{\alpha-1,\beta}}{h_k^2} + c\frac{u_{\alpha,\beta+1} - 2\bar{u}_{\alpha,\beta} + \bar{u}_{\alpha,\beta-1}}{h_k^2} = F^k_{\alpha,\beta}, \tag{3.3}$$

where the new values $\bar{u}_{\alpha-1,\beta}$, $\bar{u}_{\alpha,\beta-1}$ are used since, in the lexicographic order, by the time (α, β) is scanned new values have already replaced old values at $(\alpha - 1, \beta)$ and $(\alpha, \beta - 1)$.

A complete pass, scanning in this manner all the points of G^k, is called a (Gauss-Seidel lexicographic) G^k *relaxation sweep.* The new approximation $\bar{u}$ does not satisfy (3.2), and further relaxation sweeps may be required to improve it. An important quantity therefore is the *convergence factor*, μ say, which may be defined by

$$\mu = \|\bar{v}\| / \|v\|, \quad \text{where} \quad v = U^k - u, \quad \bar{v} = U^k - \bar{u}, \tag{3.4}$$

$\|\cdot\|$ being any suitable discrete norm.

The rate of convergence of the above relaxation scheme is asymptotically very slow. That is, except for the first few relaxation sweeps we have $\mu = 1 - O(h_k^2)$. This means that we have to perform $O(h_k^{-2})$ relaxation sweeps to reduce the error order of magnitude.

In the multi-grid method, however, the role of relaxation is not to reduce the error, but to smooth it out; i.e., to reduce the high-frequency components of the error (the lower frequencies being reduced by relaxation sweeps on coarser grids). In fact, since smoothing is basically a local process (high frequencies have short coupling range), we can analyze it in the interior of G^k by (locally) expanding the error in Fourier series. This will allow us to study separately the convergence rate of each Fourier component, and, in particular, the convergence rate of high-frequency components, which is the rate of smoothing.

Thus to study the $\theta = (\theta_1, \theta_2)$ Fourier component of the error functions v and $\bar{v}$ before and after the relaxation sweep, we put

$$v_{\alpha,\beta} = A_\theta e^{i(\theta_1\alpha + \theta_2\beta)} \quad \text{and} \quad \bar{v}_{\alpha,\beta} = \bar{A}_\theta e^{i(\theta_1\alpha + \theta_2\beta)}. \tag{3.5}$$

Subtracting (3.2) from (3.3), we get the relation

$$a(v_{\alpha+1,\beta} - 2\bar{v}_{\alpha,\beta} + \bar{v}_{\alpha-1,\beta}) + c(v_{\alpha,\beta+1} - 2\bar{v}_{\alpha,\beta} + \bar{v}_{\alpha,\beta-1}) = 0, \tag{3.6}$$

from which, by (3.5),

$$(ae^{i\theta_1} + ce^{i\theta_2})A_\theta + (ae^{-i\theta_1} + ce^{-i\theta_2} - 2a - 2c)\bar{A}_\theta = 0.$$

Hence the convergence factor of the θ component is

$$\mu(\theta) = \left|\frac{\bar{A}_\theta}{A_\theta}\right| = \left|\frac{ae^{i\theta_2} + ce^{i\theta_2}}{2a + 2c - ae^{-i\theta_1} - ce^{-i\theta_2}}\right|. \tag{3.7}$$

Define $|\theta| = \max(|\theta_1|, |\theta_2|)$. In domains of diameter $O(1)$ the lowest Fourier components have $|\theta| = O(h_k)$, and their convergence rate therefore is, $\mu(\theta) = 1 - O(h_k^2)$. Here, however, we are interested in the *smoothing factor*, which is defined by

$$\bar{\mu} = \max_{\hat{\rho}\pi \leqslant |\theta| \leqslant \pi} \mu(\theta), \tag{3.8}$$

where $\hat{\rho}$ is the mesh-size ratio and the range $\hat{\rho}\pi \leqslant |\theta| \leqslant \pi$ is the suitable range of high-frequency components, i.e., the range of components that cannot be approximated on the coarser grid, because its mesh-size is $h_{k-1} = h_k/\hat{\rho}$. We will assume here that $\hat{\rho} = ½$, which is the usual ratio (cf. Section 6.2).

Consider first the case $a = c$ (Poisson equation). A simple calculation shows that $\bar{\mu} = \mu(\pi/2, \arccos 4/5) = .5$. This is a very satisfactory rate; it implies that *three relaxation sweeps reduce the high-frequency error-components by almost an order of magnitude.* Similar rates are obtained for general a and c, provided a/c is of moderate size.

The rate of smoothing is less remarkable in the degenerate case $a \ll c$ (or $c \ll a$). For instance,

$$\mu\left(\frac{\pi}{2}, 0\right) = \left[\frac{a^2 + c^2}{a^2 + (c + 2a)^2}\right]^{½}, \tag{3.9}$$

which approaches 1 as $a \longrightarrow 0$. Thus, for problems with such a degeneracy, Gauss-Seidel relaxation is not a suitable smoothing scheme. But other schemes exist. For example,

Line Relaxation. Instead of treating each point (α, β) of G^k separately, one takes simultaneously a line of points at a time, where a line is the set of all points (α, β) in G^k with the same α (a vertical line). All the values $u_{\alpha\beta}$ on such a line are simultaneously replaced by new values $\bar{u}_{\alpha\beta}$ which simultaneously satisfy all the Eqs. (3.2) on that line. (This is easy and inexpensive to do, since the system of equations to be solved for each such line is a tridiagonal, diagonally dominant system. See, e.g., in [17].) As a result, we get the same relation as (3.3) above, except that $u_{\alpha,\beta+1}$ is replaced by $\bar{u}_{\alpha,\beta+1}$. Hence, instead of (3.7) we will get:

$$\mu(\theta) = \left|\frac{a}{2(a + c - c\cos\theta_2) - ae^{-i\theta_1}}\right| \tag{3.10}$$

from which one can derive the smoothing factor

$$\bar{\mu} = \max\left\{5^{-½}, \frac{a}{a + 2c}\right\}, \tag{3.11}$$

which is very satisfactory, even in the degenerate case $a \ll c$.

3.2. *General Results.* The above situation is very general (see [4] and Chapter 3 of [3]): For any uniformly elliptic system of difference equations, it can be shown that few relaxation sweeps are enough to reduce the high-frequency error components by an order of magnitude. The same holds for degenerate elliptic systems, provided a suitable relaxation scheme is selected. A scheme of line-relaxation which alternately uses

all line directions and all sweeping directions is suitable for *any* degenerate case. Moreover, such a scheme is suitable even for nonelliptic systems, provided it is used "*selectively*"; i.e., the entire domain is swept in all directions, but new values are not introduced at points where a local test shows the equation to be nonelliptic and the local "time-like" direction to conflict with the current sweeping direction (a conflict arises when some points are to be relaxed later than neighboring points belonging to their domain of dependence). In hyperbolic regions a selected direction would locally operate similar to marching in the local "time-like" direction, thus leaving no (or very small) residuals.

By employing *local mode analysis* (analysis of Fourier components) similiar to the example above, one can explicitly *calculate the smoothing rate* $|\log \bar{\mu}|^{-1}$ for any given difference equation with any given relaxation scheme. (Usually $\bar{\mu}$ should be calculated numerically; an efficient FORTRAN subroutine exists; typical values are given in Table 1, in Section 6.2.) In this way, one can select the best relaxation scheme from a given set of possibilities. The selection of the difference equation itself may also take this aspect into account. This analysis can also be done for *nonlinear* problems (or linear problems with nonconstant coefficients), by local linearization and coefficients freeze. Such localization is fully justified here, since we are interested only in a local property (the property of smoothing. By constrast, one cannot make similar mode analysis to predict the overall convergence rate μ of a given relaxation scheme, since this is not a local property).

An important feature of the smoothing rate $\bar{\mu}$ is its *insensitivity*. In the above example no relaxation parameters were assumed. We could introduce the usual *relaxation parameter* ω; i.e., replace at each point the old value $u_{\alpha,\beta}$ not with the calculated $\bar{u}_{\alpha,\beta}$, but with $u_{\alpha,\beta} + \omega(\bar{u}_{\alpha,\beta} - u_{\alpha,\beta})$. The mode analysis shows, however, that no $\omega \neq 1$ provides a smoothing rate better than $\omega = 1$. In other cases, $\omega = 1$ is not optimal, but its $\bar{\mu}$ is not significantly larger than the minimal $\bar{\mu}$. In delayed-displacement relaxation schemes a value $\omega < \omega_{\text{critical}} < 1$ should often be used to obtain $\bar{\mu} < 1$, but there is no sensitive dependence on the precise value of ω, and suitable values are easily obtained from the local mode analysis. Generally, smoothing rates of delayed-displacement schemes are somewhat worse than those of immediate-displacement schemes, and the latter should therefore be preferred, except when parallel processing is used. In hyperbolic regions immediate-displacement schemes *should* be used, with $\omega = 1$.

3.3. *Acceleration by Weighting.* The smoothing factor $\bar{\mu}$ may sometimes be further improved by various parameters introduced into the scheme. Since $\bar{\mu}$ is reliably obtained from the local mode analysis, we can choose these parameters to minimize $\bar{\mu}$. For linear problems, such optimal parameters can be determined once and for all, since they do not depend on the shape of the domain. For nonlinear problems precise optimization is expensive and one should prefer the simpler, more robust relaxation schemes, such as SOR.

One general way of parametrization is the *weighting of corrections.* We first calculate, in any relaxation scheme, the required correction $\delta u_\nu = \bar{u}_\nu - u_\nu$ (where $\nu =$

(α, β) or, for a general dimension d, $\nu = (\nu_1, \nu_2, \ldots, \nu_d)$, ν_j integers). Then, instead of introducing these corrections, we actually introduce corrections which are some linear combination of δu_ν at neighboring points. That is, the actual new values are

$$\tilde{u}_\nu = u_\nu + \sum_{\gamma \in \Gamma} \omega_\gamma \delta u_{\nu+\gamma}, \tag{3.12}$$

where the weights ω_γ are the parameters, $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_d)$, γ_j integers, and Γ is a small set near $(0, 0, \ldots, 0)$. For any fixed Γ we can optimize the weights. In case $\Gamma = \{0\}$, ω_0 is the familiar relaxation parameter. Weighting larger Γ is useful in delayed-displacement relaxation schemes. For immediate-displacement schemes one can examine taking weighted averages of old and new values.

Examples. In case of simultaneous displacement (Jacobi) relaxation for the 5-point Poisson equation, the optimal weights for $\Gamma = \{0\}$ is $\omega_{00} = .8$, for which the smoothing factor is $\bar{\mu} = .60$. For the set $\Gamma = \{(\gamma_1, \gamma_2): |\gamma_1| + |\gamma_2| \leqslant 1\}$ the optimal weights are $\omega_{00} = 6\omega_{0,\pm 1} = 6\omega_{\pm 1,0} = 48/41$, yielding $\bar{\mu} = 9/41$. This rate seems very attractive; the smoothing obtained in one sweep equals that obtained by $(\log 9/41)/(\log 1/2) = 2.2$ sweeps of Gauss-Seidel relaxation. Actually, however, each sweep of this weighted-Jacobi relaxation requires nine additions and three multiplications per grid point, whereas each Gauss-Seidel sweep requires only four additions and one multiplication per point, so that the two methods have almost the same convergence rate per operation, Gauss-Seidel being slightly faster. The weighted Jacobi scheme is considerably more efficient than any other simultaneous-displacement scheme, but like any carefully weighted scheme, it is considerably more sensitive to various changes.

The acceleration by weighting can be more significant for higher-order equations. For the 13-point biharmonic operator, Gauss-Seidel relaxation requires twelve additions and three multiplications per grid point and gives $\bar{\mu} = .802$, while weighted Jacobi (with weights $\omega_{00} = 1.552$, $\omega_{0,\pm 1} = \omega_{\pm 1,0} = .353$) requires seventeen additions and five multiplications per point and gives $\bar{\mu} = .549$, which is 2.7 times faster. (The best relaxation sweep for the biharmonic equation $\Delta^2 U = F$ is to write it as the system $\Delta V = F$, $\Delta U = V$ and sweep Gauss-Seidel, alternatively on U and V. Such a double sweep costs eight additions and two multiplications per grid point, and yields $\bar{\mu} = .5$. But a similar procedure is not possible for general fourth-order equations.)

4. A Multi-Grid Algorithm (Cycle C) for Linear Problems. There are several actual algorithms for carrying out the basic multi-grid idea, each with several possible variations. We present here an algorithm (called "Cycle C" in [3]) which is easy to program, generally applicable and never significantly less efficient than the others ("Cycle A" and "Cycle B"). The operation of the algorithm for linear problems is easier to learn, and is therefore described first. In the next section the FAS (Full Approximation Storage) mode of operation, suitable for nonlinear problems and other important generalizations, will be described. A flow-chart of the algorithm is given in Figure 1. (For completeness, we also flowchart, in Figure 2, Cycles A and B.) A sample FORTRAN program of Cycle C, together with a computer output, is given in Appendix B.

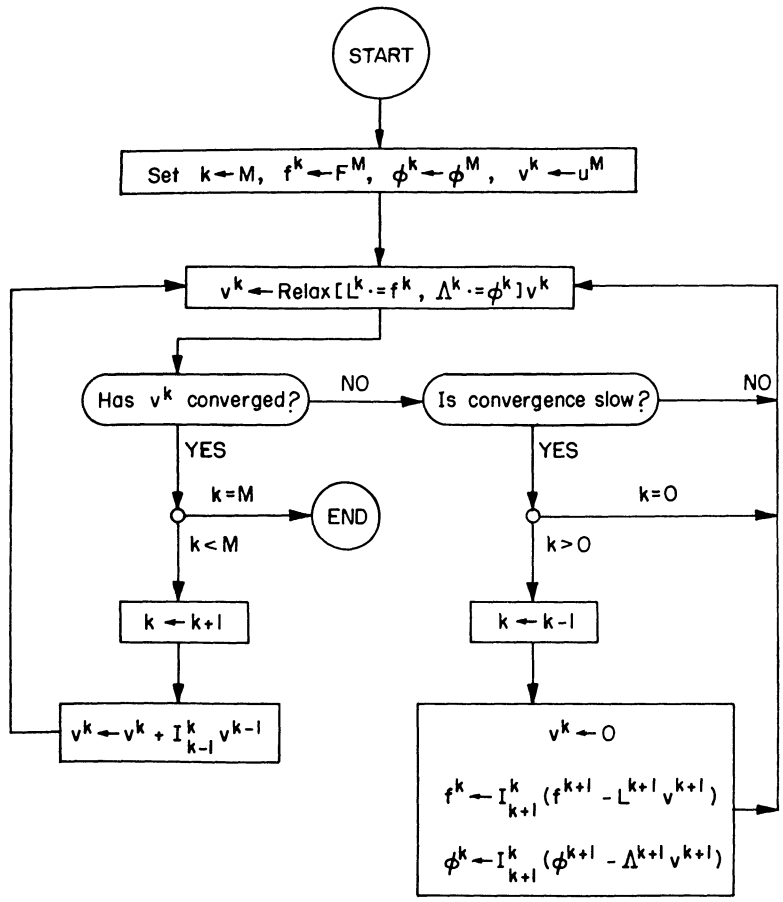


FIGURE 1. *Cycle* C, *Linear Problems*

Cycle C starts with some approximation u_0^M being given on the finest grid G^M. In the linear case one can start with any approximation, but a major part of the computations is saved if u_0^M has smooth residuals (e.g., if u_0^M satisfies the boundary conditions and $L^M u_0^M - F^M$ is smooth. As explained in Section 6, smoothing the residuals involves most of the computational effort). In the nonlinear case, one may have to use a continuation procedure, usually performed on coarser grids (cf. Section 8.2). Even for linear problems, the most efficient algorithm is to obtain u_0^M by interpolating from an approximate solution u^{M-1} calculated on G^{M-1} by a similar algorithm. (Hence the denomination "cycle" for our present algorithm, which would generally serve as the basic step in processes of continuation, refinement and grid adaptation, or as a time step in evolution problems.) For highest efficiency, the interpolation from u^{M-1} to u_0^M should be of sufficiently high order, to exploit all smoothness in u^{M-1}. (Cf. (A.7) in Section A.2, and see also Section 6.3.)

The basic rule in Cycle C is that each v^k (the function defined on the grid G^k; $k = 0, 1, \ldots, M-1$) is designed to serve as a correction for the approximation v^{k+1} previously obtained on the next finer grid G^{k+1}, if and when that approximation

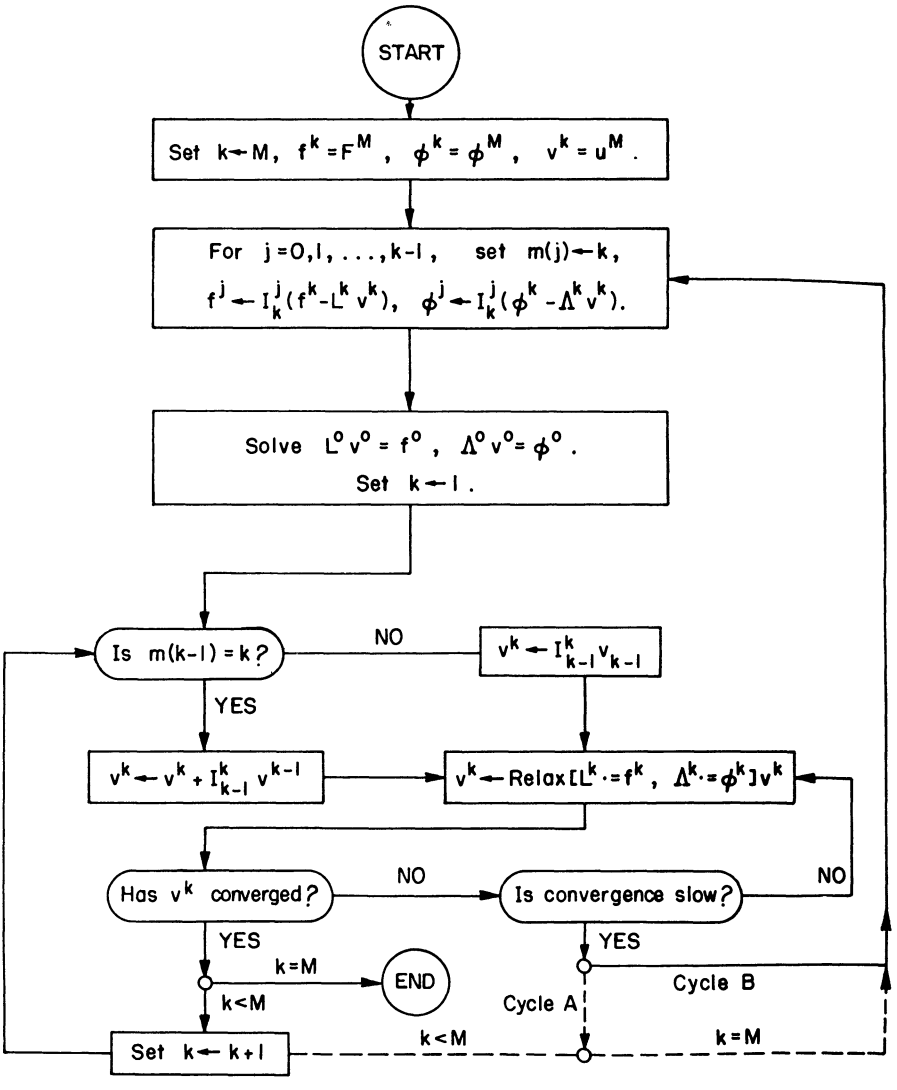


FIGURE 2. *Cycles* A *and* B, *Linear Problems*

actually requires a coarse-grid correction, i.e., if and when relaxation over G^{k+1} exhibits a slow rate of convergence. Thus, the equations to be (approximately) satisfied by v^k are**

$$L^k V^k = f^k, \qquad \Lambda^k V^k = \phi^k, \tag{4.1}$$

where f^k and ϕ^k are the residuals (to the interior equation and the boundary condition, respectively) left by v^{k+1}, that is,

$$f^k = I^k_{k+1}(f^{k+1} - L^{k+1}v^{k+1}), \qquad \phi^k = I^k_{k+1}(\phi^{k+1} - \Lambda^{k+1}v^{k+1}). \tag{4.2}$$

**We denote by V^k the functions in the equations, to distinguish them from their computed approximations v^k. When v^k is changing in the algorithm (causing V^{k-1} to change), V^k remains fixed.

We use the notation I_m^k to represent *interpolation* from G^m to G^k. In case $m > k$, I_m^k may represent a simple transfer of values to the coarser grid G^k from the corresponding points in the finer grid G^m; or instead, it may represent transfer of some weighted averages. In case $k > m$, as in step (e) below, I_m^k is usually a polynomial interpolation of a suitable order (at least the order of the differential equation. See Sections A.2 and A.4 for more details).

The equations on G^k are thus defined in terms of the approximate solution on G^{k+1}. On the finest grid G^M, the equations are the original ones; namely

$$f^M = F^M, \qquad \phi^M = \Phi^M, \qquad v^M = u^M. \tag{4.3}$$

The steps of the algorithm are the following:

(a) Set $k \leftarrow M$ (k is the working level; we start at the finest level), and introduce the given approximation $v^M \leftarrow u_0^M$.

(b) Improve v^k by one relaxation sweep for the difference equations (4.1). Symbolically, we write such a sweep as

$$v^k \leftarrow \text{Relax}[L^k \cdot = f^k, \Lambda^k \cdot = \phi^k]v^k. \tag{4.4}$$

(c) If relaxation has sufficiently converged (the precise criterion is described in Sections A.7 and A.8), go to step (e). If not, and if the convergence rate is still fast (by a criterion given in Section A.6) go back to step (b). If convergence is not obtained and the rate is slow, go to step (d).

(d) If $k = 0$ (meaning slow convergence has taken place at the coarsest grid G^0), go back to step (b) (to continue relaxation nevertheless, since on G^0 relaxation is very inexpensive. If, however, the problem is indefinite, then slow rate of *divergence* may occur, in which case the G^0 problem should be solved directly. This is as inexpensive as relaxation, but requires additional programming. See Section 4.1 below). If $k > 0$, lower k by 1 (to compute correction on the next, coarser level). Compute f^k and ϕ^k on this new level, using (4.2), put $v^k = 0$ as the starting approximation, and go to step (b).

(e) If $k = M$ (convergence has been obtained on the finest level), the algorithm is terminated. If $k < M$ (v^k has converged and is ready to serve as a correction to v^{k+1}), put

$$v^{k+1} \leftarrow v^{k+1} + I_k^{k+1} v^k. \tag{4.5}$$

Then advance k by 1 (to resume computations at the finer level) and go to step (b).

The storage required for this algorithm is only a fraction more than the number of locations, $2n$ say, required to store u^M and F^M on the finest grid. Indeed, for a d-dimensional problem, a storage of roughly $2n/2^d$ locations is required to store v^{M-1} and f^{M-1}, the next level requires $2n/2^{2d}$, etc. The total for all levels is

$$2n(1 + 2^{-d} + 2^{-2d} + \cdots) \leqslant 2n2^d/(2^d - 1). \tag{4.6}$$

(In the FAS version below, a major reduction of storage area is possible through segmental refinement. See Section 7.5.)

4.1. *Indefinite Problems and the Size of the Coarsest Grid.* If, on any grid G^k, the boundary-value problem (4.1) is a nondefinite elliptic problem, with eigenvalues

$$\lambda_1^k \leqslant \lambda_2^k \leqslant \cdots \leqslant \lambda_l^k < 0 < \lambda_{l+1}^k \leqslant \lambda_{l+2}^k \leqslant \cdots, \tag{4.7}$$

and with the corresponding eigenfunctions $V_1^k, V_2^k, \ldots, V_l^k, V_{l+1}^k, \ldots$, then it cannot be solved by straight relaxation. Any relaxation sweep will reduce the error components in the space spanned by $V_{l+1}^k, V_{l+2}^k, \ldots$, but will *magnify* all components in the span of $V_1^k, V_2^k, \ldots, V_l^k$. A multi-grid solution, however, is not seriously affected by this magnification, provided the magnified components are suitably reduced by the coarse-grid corrections. This will usually be the case, since these components are basically of low frequency and are well approximated on coarser grids. But care should be taken regarding the coarsest grid:

On the coarsest grid, an indefinite problem should be solved directly (i.e., not by relaxation of any kind. Semi-iterative solutions, like Newton iterations for nonlinear problems, are, of course, permissible). Furthermore, *this grid should be fine enough to provide rough approximation to the first* $(l+1)$ *eigenfunctions*, so that

$$0 < \lambda_j^{k+1}/\lambda_j^k < \frac{2}{w_k} \qquad (1 \leqslant j \leqslant l+1,\ 1 \leqslant k \leqslant M-1), \tag{4.8}$$

where w_k is an "under-interpolation" factor that should multiply I_k^{k+1} in (4.5) above when the usual value ($w_k = 1$) does not satisfy (4.8). This means that G^0 should contain at least $O(l)$, probably $2l$, points. Also, G^0 should be just fine enough to still have smoothing capability at any finer level G^k. For example, if SOR relaxations with $\omega < \omega_c$ are used, h_0 should satisfy (see [4] or Section 3 in [3])

$$\operatorname{Re}\{B(\theta, h)/b_0(h)\} > 0 \qquad (0 \leqslant h \leqslant h_1), \tag{4.9}$$

where $B(\theta, h)$ is the symbol of L^h (see (A.3) in Appendix A) and $b_0(h)$ is its central coefficient.

Usually, G^0 can still be coarse enough to have the direct solution of its equations still far less expensive than, say, one relaxation sweep over the finest grid, so that the indefinite problem is solved with the same overall efficiency as definite problems.

5. The FAS (Full Approximation Storage) Algorithm. In the FAS mode of the multi-grid algorithms, instead of storing a correction v^k (designed to correct the finer-level approximation u^{k+1}), the idea is to store the full current approximation u^k, which is the sum of the correction v^k and its base approximation u^{k+1}:

$$u^k = I_{k+1}^k u^{k+1} + v^k \quad (k = 0, 1, \ldots, M-1). \tag{5.1}$$

In terms of these full-approximation functions, we can rewrite the correction equations (4.1)–(4.3) as***

$$L^k U^k = \bar{F}^k, \qquad \Lambda^k U^k = \bar{\Phi}^k, \tag{5.2}$$

***Again we distinguish between the notation U^k used to write the equations and the computed approximation u^k. Equation (5.2), for $k < M$, is not equivalent to (2.2), although they both use the notation U^k.

where

$$\begin{aligned} \bar{F}^k &= L^k(I_{k+1}^k u^{k+1}) + I_{k+1}^k(\bar{F}^{k+1} - L^{k+1}u^{k+1}), \\ \bar{\Phi}^k &= \Lambda^k(I_{k+1}^k u^{k+1}) + I_{k+1}^k(\bar{\Phi}^{k+1} - \Lambda^{k+1}u^{k+1}), \quad (k = 0, 1, \ldots, M-1), \end{aligned} \tag{5.3}$$

and where for $k = M$ we have the original problem, i.e.,

$$\bar{F}^M = F^M, \qquad \bar{\Phi}^M = \Phi^M. \tag{5.4}$$

For linear problems, Eqs. (5.2)–(5.4) are exactly equivalent to (4.1)–(4.3). The advantage of the FAS mode is that Eqs. (5.2)–(5.4) apply equally well to *nonlinear* problems. To see this, consider for instance the nonlinear equation $L^M U^M = F^M$ given on the finest grid. Given an approximate solution u^M we can still improve it by relaxation sweeps, with smoothing rates $\bar{\mu}$ (varying over the domain, but still reliably estimated by mode analyses, applied locally to the linearized-frozen equation). As in the linear case, the smoothed-out functions are the residual $f^M = F^M - L^M u^M$ and the correction $U^M - u^M$. Therefore, the equation that can be approximated on coarser grids is the residual equation $L^M U^M - L^M u^M = f^M$. Its coarser-grid approximation is

$$L^{M-1}U^{M-1} - L^{M-1}I_M^{M-1}u^M = I_M^{M-1}f^M, \tag{5.5}$$

which is the same as (5.2) for $k = M - 1$. In interpolating U^{M-1} (or a computed approximation u^{M-1}) back to G^M, we should actually interpolate $U^{M-1} - I_M^{M-1}u^M$, because this is the coarse-grid approximation to the smoothed-out function $U^M - u^M$. Similarly, in interpolating an (approximate) solution u^k of (5.2) to the finer grid G^{k+1}, the polynomial interpolation should operate on the correction. Thus the interpolation is

$$u^{k+1} \leftarrow u^{k+1} + I_k^{k+1}(u^k - I_{k+1}^k u^{k+1}), \tag{5.6}$$

which is equivalent to (4.5). Note that generally,

$$I_k^{k+1} I_{k+1}^k u^{k+1} \neq u^{k+1}.$$

The FAS (Cycle C) algorithm is the same algorithm as in Section 4, with the FAS equations (5.2)–(5.4) replacing (4.1)–(4.3), and with (5.6) replacing (4.5). It is flowcharted in Figure 3.

The FAS mode has several important advantages: It is suitable for general nonlinear problems, with the same procedures (relaxation and interpolation routines) used at all levels. Thus, for example, only one relaxation routine need be written. Moreover, this mode is suitable for *composite grids* (nonuniform grids created by increasingly finer levels being defined on increasingly smaller subdomains; see Section 7.2), which is the basis for grid adaptation on one hand, and segmental refinement (see Section 7.5) on the other hand. Generally speaking, the basic feature of the FAS mode is that the function stored on a coarse grid G^k coincides there with the fine-grid solution: $u^k = I_M^k u^M$. This enables us to manipulate accurate solutions on coarse grids.

The storage required for the FAS algorithm is again given by (4.6). With segmental refinement (Section 7.5) it can be reduced far below that, even to $O(\log n)$.

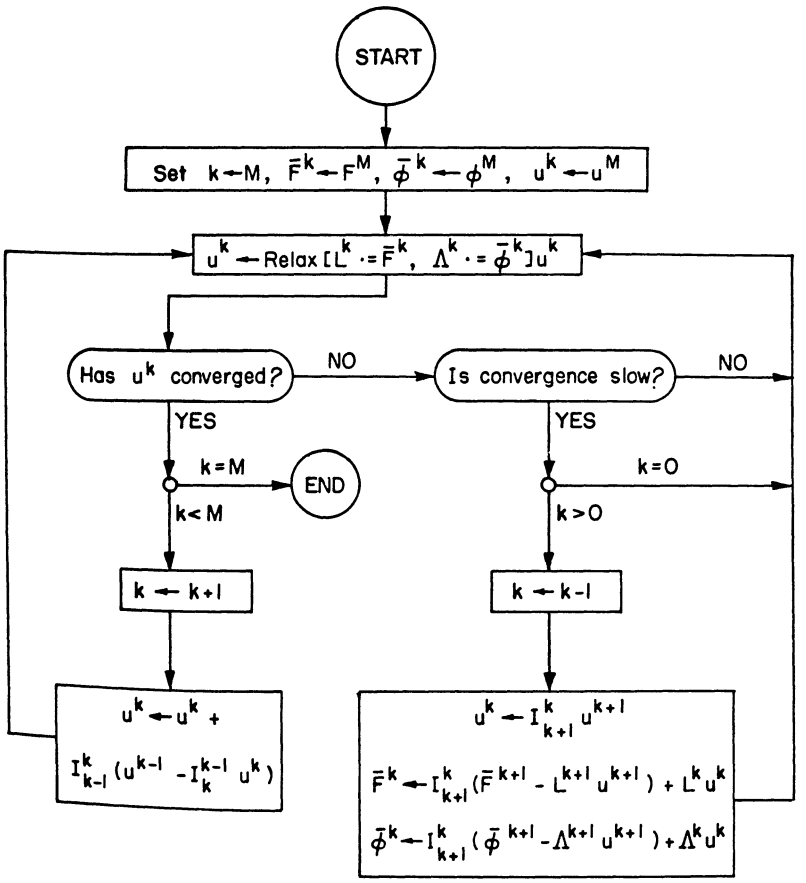


FIGURE 3. *Cycle* C, *Full Approximation Storage*

An important by-product of the FAS mode is a good *estimate for the truncation error*, which is useful in defining natural stopping criteria (see Section A.8) and grid-adaptation criteria (Section 8.3). Indeed, for any $k < m \leqslant M$ it can easily be shown (by induction on m, using (5.2)–(5.3)) that

$$\begin{aligned} \bar{F}^k - I_m^k \bar{F}^m &= L^k(I_m^k u^m) - I_m^k L^m u^m, \\ \bar{\Phi}^k - I_m^k \bar{\Phi}^m &= \Lambda^k(I_m^k u^m) - I_m^k \Lambda^m u^m, \end{aligned} \tag{5.7}$$

which are exactly the G^m approximations to the G^k truncation errors.

A slight disadvantage of the FAS mode is the longer calculation required in computing $\bar{F}^k$, almost twice longer than calculating f^k in the former (Correction Storage) mode. This extra calculation is equivalent to one extra relaxation sweep on G^k, but only for $k < M$, and is about 5% to 10% of the total amount of calculations. Hence, for linear problems on uniform grids, the CS mode is slightly preferable.

6. Performance Estimates and Numerical Tests.

6.1. *Predictability.* An important feature of the multi-grid method is that, although iterative, its total computational work can be predicted in advance by local

mode (Fourier) analysis. Such an analysis, which linearizes and freezes the equations and ignores distant boundaries, gives a very good approximation to the behavior of high-frequency components (since they have short coupling range), but usually fails to approximate the behavior of the lowest frequencies (which interact at long distances). The main point here, however, is that these lowest frequencies may indeed be ignored in the multi-grid work estimates, since their convergence is obtained on coarser grids, where the computational work is negligible. The purpose of the work on the finer grids is only to converge the high frequencies. Thus, the mode-analysis predictions, although not rigorous, are likely to be very realistic. In fact, these predictions are in full agreement with the results of our computational tests. This situation is analogous to the situation in time dependent problems, where precise stability criteria are derived from considering the corresponding linearized-frozen-unbounded problems. (See page 91 in [30].) Rigorous convergence estimates, by contrast, have no use in practice. Except for very simple situations (see Appendix C and Section 10) they yield upper bounds to the computational work which are several orders of magnitude larger than the actual work.

The predictability feature is important since it enables us to *optimize* our procedures (cf. Section 3.3, Appendix A). It is also indispensable in *debugging* multi-grid programs.

6.2. *Multi-Grid Rates of Convergence.* To get a convenient measure of convergence per unit work, we define as our *Work Unit* (WU) the computational work in one relaxation sweep over the finest grid G^M. The number of computer operations in such a unit is roughly wn, where n is the number of points in G^M and w is the number of operations required to compute the residual at each point. (In parallel processing the count should, of course, be different. Also, the work unit should be further specified when comparing different discretization and relaxation schemes.) If the mesh-size ratio is $\hat{\rho} = h_{k+1}/h_k$ and the problem's domain is d-dimensional, then a relaxation sweep over G^{m-j} costs approximately $\hat{\rho}^{dj}$ WU's (assuming the grids are co-extensive, unlike those in Section 7).

Relaxation sweeps make up most of the multi-grid computational work. The only other process that consumes any significant amount of computations is the I_k^{k-1} and I_{k-1}^k interpolations. It is difficult to measure them precisely in WU's, but their total work is always considerably smaller than the total relaxation work. In the example in Appendix B, the interpolation work is about 20% of the relaxation work. Usually the percentage is even lower, since relaxing Poisson problems is particularly inexpensive. To unify our estimates and measurements we will therefore *define the multi-grid convergence factor* $\mathring{\mu}$ *as the factor by which the errors are reduced per one WU of relaxation, ignoring any other computational work* (which is never more than 30% of the total work).

The multi-grid convergence factor may be estimated by a full local mode analysis. The following is a simplified analysis, which gives a good approximation. We assume that a relaxation sweep over any grid G^k affects error components $e^{i\Theta\cdot x}$ only in the range $\pi/h_{k-1} \leqslant |\Theta| \leqslant \pi/h_k$, where

$$(6.1) \qquad \Theta = (\Theta_1, \Theta_2, \ldots, \Theta_d), \qquad \Theta \cdot x = \sum_{j=1}^{d} \Theta_j x_j, \qquad |\Theta| = \max_{1 \leqslant j \leqslant d} |\Theta_j|.$$

(The θ/h_k of Section 3 and Appendix A is denoted here by Θ, to unify the discussion of all levels.) In fact, if a proper interpolation scheme is used (see Section A.2), only components in the range $|\Theta| \leqslant (1 + \epsilon)\pi/h_{k-1}$, say, are affected by interactions with coarser grids. But if proper residual-weighting is also used (to make $\bar{\sigma} = 1$; cf. Section A.4), then the combined action of the coarse-grid correction cycles and the G^k relaxation sweeps yield convergence rates which are slowest at $|\Theta| = \pi/h_{k-1}$ (cf. Appendix A). For such Θ the coarse-grid cycles have neutral effect, since $\bar{\sigma} = 1$, hence the convergence rate is indeed as affected only by the G^k relaxation sweeps.

One relaxation sweep over G^k reduces the error components in the range $\pi/h_{k-1} \leqslant |\Theta| \leqslant \pi/h_k$ by the smoothing factor $\bar{\mu}$. (See Section 3. If the smoothing factor near a boundary is slower than $\bar{\mu}$, which is not the usual case, smoothing may be accelerated there by partial relaxation sweeps—cf. Section A.9.) Thus a multi-grid cycle with s relaxation sweeps on each level reduces all error components by the factor $\bar{\mu}^s$. The amount of Work Units expended in these sweeps is

$$s + s\hat{\rho}^d + s\hat{\rho}^{2d} + \cdots + s\hat{\rho}^{(M-1)d} < s/(1 - \hat{\rho}^d).$$

Hence, the multi-grid convergence factor is

$$(6.2) \qquad \mathring{\mu} = \bar{\mu}^{(1-\hat{\rho}^d)},$$

which is not much bigger than $\bar{\mu}$. In case $\bar{\sigma} > 1$, the effective smoothing factor $\tilde{\mu}$ (see (A.8)) should replace $\bar{\mu}$ in this estimate.

Estimate (6.2) is not rigorous, but is simple to compute and very realistic. In fact, numerical experiments (Sections 6.4–6.5) usually show slightly faster (smaller) factors $\mathring{\mu}$, presumably because the worst combination of Fourier components is not always present.

The theoretical multi-grid convergence factors, for various representative cases, are summarized in Table 1.

Explanations to Table 1. The first column specifies the difference operator and the dimension d. Δ_h denotes the central second-order ($(2d + 1)$-point) approximation, and $\Delta_h^{(4)}$ the fourth-order ($(4d + 1)$-point "star") approximation, to the Laplace operator. Δ_h^2 is the central 13-point approximation to the biharmonic operator. The operators ∂_x, ∂_y, ∂_{xx} and ∂_{yy} are the usual central second-order approximations to the corresponding partial-differential operators. ∂_x^- is the backward approximation. Upstream differencing is assumed for the inertial terms of the Navier-Stokes equations; central differencing for the viscosity terms, forward differencing for the pressure terms, and backward differencing for the continuity equation. Rh is the Reynolds number times the mesh-size.

The second column specifies the relaxation scheme and the relaxation parameter ω. SOR is Successive Over Relaxation, which for $\omega = 1$ is the Gauss-Seidel relaxation. xLSOR (yLSOR) is Line SOR, with lines in the x (y) direction. yLSOR +, yLSOR − and yLSORs indicate, respectively, relaxation marching forward, backward and symmetrically (alternately forward and backward). CSOR means Collective SOR (see

Section 3 in [3]) and the attached ω's are ω_1 for the velocity components and ω_2 for the pressure. "Downstream" means that the flow direction and the relaxation marching direction are the same; "upstream" means that they are opposite. **ADLR** denotes Alternating Direction Line Relaxation (a sweep of x**LSOR** followed by a sweep of y**LSOR**). **SD** is Simultaneous Displacement (Jacobi) relaxation, **WSD** is Weighted Simultaneous Displacement with the optimal weights as specified in Section 3.3 above (and with other weights, to show the sensitivity). **WSD**Δ (for Δ_h^2) is like **WSD**, except that residuals are computed in less operations by making first a special pass that computes $\Delta_h u$. y**LSD** is y-lines relaxation with simultaneous displacement, **ADLSD** is the corresponding alternating-direction (y**LSD** alternating with x**LSD**) scheme.

TABLE 1. *Theoretical smoothing and MG-convergence rates*

L_h	d	Relax. Scheme	ω	$\hat{\rho}$	$\bar{\mu}$	$\overset{o}{\mu}$	$\lvert \ln \overset{o}{\mu} \rvert^{-1}$	add	mult	W_M
Δ_h	1	SOR	1	1:3	.557	.693	2.73	2	1	9.0
				1:2	.477	.668	2.49	3	2	6.9
				2:3	.378	.723	3.08	3	2	7.5
	2	SOR	1	1:3	.667	.697	2.77	4	1	6.8
			.8	1:2	.552	.640	2.24	5	2	4.1
			1		.500	.595	1.92	4	1	3.5
			1.2		.552	.640	2.24	5	2	4.1
			1	2:3	.400	.601	1.96	4	1	2.9
		LSOR	1	1:2	.447	.547	1.66	8	4	3.1
		ADLR	1		.386	.490	1.40	8	4	2.6
			.8		.456	.555	1.70	8	4	3.1
		SD	.8	1:2	.600	.682	2.61	5	2	4.8
		WSD	1.17, .195		.220	.321	0.88	9	3	1.6
			1.40, .203		.506	.600	1.96	9	3	3.6
	3	SOR	1	1:3	.738	.746	3.42	6	1	7.8
				1:2	.567	.608	2.01	6	1	3.7
				2:3	.441	.562	1.73	6	1	2.0
$\Delta_h^{(4)}$	2	SOR	.8	1:2	.581	.665	2.46	9	3	9.1
			1		.534	.625	2.13	8	2	7.9
			1.2		.582	.666	2.46	9	3	9.1
		LSOR	1		.484	.580	1.84	14	7	6.8
	3	SOR	1		.596	.636	2.21	12	2	7.0
$\partial_{xx} + 2\partial_x\partial_y + \partial_{yy}$	2	SOR	1	1:2	. 62	.699	2.79	8	2	5.2
		LSOR,ADLR			.447	.547	1.66	12	5	3.1
Δ_h^2	2	SOR	1	1:2	.802	.847	6.04	12	3	11.1
			1	2:3	.666	.798	4.43	12	3	6.5
		WSD	1.552, .353	1:2	.549	.638	2.22	17	5	4.1
			1.4 , .353		1.03	div.	div.	17	5	div.
		WSDΔ	1.552, .353		.549	.638	2.22	14	4	4.1
NAVIER - STOKES		CSOR								
Rh = 0	2	downstr.	1, .5	1:2	.800	.846	5.98	18	6	11.0
any			1, .5		.800	.846	5.98	33	16	11.0
100			1.1, .5		1.73	div.	div.	33	16	div.
100			.8, .5		.93	.947	18.7	33	16	34.5
10		upstream	1, .5		.884	.912	10.8	33	16	20.0
100			1, .5		.994	.995	220.	33	16	400.
100			.8, .5		.984	.988	83.	33	16	150.
0	3	downstr.	1, .5		.845	.863	6.79	33	8	10.7
any			1, .5		.845	.863	6.79	60	25	10.7
10		upstream	1, .5		.874	.889	8.49	60	25	13.4
100			1, .5		.989	.990	100.	60	25	160.
STOKES' (Rh = 0)		SOR	1,.33		.707					

TABLE 1 (*Continued*). *Here* $d = 2$, $\hat{\rho} = 1 : 2$

L_h	Relax. Scheme	ω	$\bar{\mu}$
$\partial_{xx} + \varepsilon\partial_{yy}$, $\varepsilon \ll 1$	SOR, xLSOR	any	$1 - O(\varepsilon)$
$a\partial_{xx} + c\partial_{yy}$	yLSOR	1	$\max(5^{-1/2}, \frac{a}{a+2c})$
$(q = \min(\frac{a}{c}, \frac{c}{a}))$	ADLR		$5^{-1/4}(1+2q)^{-1/2}$
	SD, yLSD, ADLSD	1	1
	SD	$(2q+2)/(3q+2)$	$(q+2)/(3q+2)$
	yLSD	$(2a+2c)/(2a+3c)$	$(2a+c)/(2a+3c)$
	ADLSD	2/3, 2/3	$\leq 3^{-1/2} = .577$
$\Delta_h - \frac{\eta}{h}\partial_x$	yLSOR	1	$\max\left(\frac{1-\eta}{3-\eta}, \left[\frac{1-\eta+\eta^2/4}{5+\eta+\eta^2/4}\right]^{1/2}\right)$
$\Delta_h - \frac{\eta}{h}\partial_x^-$	yLSOR+	1	$\max\left(\frac{1}{3}, [5+6\eta+2\eta^2]^{-1/2}\right)$
$(\eta > o)$	yLSOR-		$\max(\frac{1}{3}, \lvert\frac{1+\eta}{2+\eta+i}\rvert)$
	yLSORs		$\leq 3^{-1/2} = .577$
Navier - Stokes with large Rh in 2 or 3 dimensions	SOR (pressure corrected by the continuity equation), downstream or up-stream, with any relaxation parameters.		$\geq 1 - \frac{2}{Rh}$

The next columns list $\hat{\rho} = h_k : h_{k+1}$ (see discussion below), the smoothing factor $\bar{\mu}$ as defined by (3.8), and $\mathring{\mu}$, calculated by (6.2). We also list the *multi-grid convergence rate* $|\log \mathring{\mu}|^{-1}$, which is the theoretical number of relaxation Work Units required to reduce the error by the factor e, and W_M, the overall multi-grid computational work (see Section 6.3). To make comparisons of different schemes possible, we also list, for each case, the number of operations per grid point per sweep. This number times n (the number of points in G^M) gives the number of operations in a Work Unit. We list only the basic number of additions and multiplications (counting shifts as multiplications), thus ignoring the operations of transferring information, indexing, etc., which may add up to a significant amount of operations, but which are too computer- and program-dependent to be specified. Also, we assumed that the right-hand sides f^k, including f^M, are stored in the most efficient form (e.g., $h^2 f^M$ is actually stored). Note that the SOR operation count is smaller for $\omega = 1$ (Gauss-Seidel) than for any other ω.

For Navier-Stokes equations with CSOR we found $\bar{\mu}$ to be largest when relaxing upstream, and smallest when relaxing downstream. We also found that in marching alternately back and forth, the worst overall smoothing rate is for flows aligned with the relaxation, i.e., flows for which this relaxation is alternately upstream and downstream. The worst $\bar{\mu}$ is therefore $(\bar{\mu}_u \bar{\mu}_d)^{1/2}$, where $\bar{\mu}_u$ and $\bar{\mu}_d$ are, respectively, the "upstream" and "downstream" values shown in the table.

Numbers in this table were calculated by Allan S. Goodman, at IBM Thomas J. Watson Research Center. A more extensive list is in preparation.

Mesh-Size Ratio Optimization. Examining Table 1, and many other unlisted examples, it is evident that the mesh-size ratio $\hat{\rho} = 1 : 2$ is close to optimal, yielding almost minimal $|\log \mathring{\mu}|^{-1}$ and minimal W_M. This ratio is more convenient and more economical in the interpolation processes (which are ignored in the above calculations) than any other efficient ratio. In practice, therefore, *the ratio* $\hat{\rho} = 1 : 2$ *should always be used*, giving also a very desirable standardization.

6.3. *Overall Multi-Grid Computational Work.* Denote by W_M the computational work (in the above Work Units) required to solve the G^M problem ((2.2), $k = M$) to the level of its truncation errors τ^M (cf. Section A.8). If the problem is first solved on G^{M-1} to the level τ^{M-1}, and if the correct order of interpolation is used to interpolate the solution to G^M (so that unnecessary high-frequencies are not excited, cf. Section A.2, and in particular (A.7) for $i = 1$), then the residuals of this first G^M approximation are $O(\tau^{M-1})$. The computational work required to reduce them to $O(\tau^M)$ is $\log O(\tau^M/\tau^{M-1})/\log \mathring{\mu}$. Hence,

$$W_M = W_{M-1} + \log \frac{\tau^M}{\tau^{M-1}} \Big/ \log \mathring{\mu}. \tag{6.3}$$

Similarly, we can solve the G^{M-j} problem expending work

$$W_{M-j} = W_{M-j-1} + \hat{\rho}^{jd} \log \frac{\tau^{M-j}}{\tau^{M-j-1}} \Big/ \log \mathring{\mu} \tag{6.4}$$

(since a G^{M-j} work unit is $\hat{\rho}^{jd}$ times the G^M unit). If we use p-order approximations, then

$$\tau^k/\tau^{k-1} \leqslant O(h_k^p)/O(h_{k-1}^p) = O(\hat{\rho}^p). \tag{6.5}$$

Hence, using (6.4) for $j = 0, 1, 2, \ldots, M-1$ and neglecting W_0,

$$W_M \leqslant (1 + \hat{\rho}^d + \hat{\rho}^{2d} + \cdots)p \log \hat{\rho}/\log \mathring{\mu}.$$

Or, by (6.2),

$$W_M \leqslant (p \log \hat{\rho})/((1 - \hat{\rho}^d)^2 \log \bar{\mu}). \tag{6.6}$$

(The same $\hat{\rho}$ was assumed in computing the first approximation and in the improvement cycles. This of course is not necessary.)

Typical values of this theoretical W_M are shown in Table 1 above. In actual computations a couple of extra Work Units are always expended in solving a problem, because we cannot make nonintegral numbers of relaxation sweeps or MG cycles, and also because we usually solve to accuracy below the level of the truncation errors.

For 5-point Poisson problems, for example, the following procedure gives a G^M solution with residuals smaller than τ^M. (i) Obtain u^{M-1} on G^{M-1}, with residuals smaller than τ^{M-1}. (ii) Starting with the *cubic* interpolation $u^M \leftarrow I^M_{M-1} u^{M-1}$ (preferably by using the difference operator itself; cf. [7]), make a MG correction cycle such as Cycle C with $\eta = 0$ (i.e., switching to G^{k-1} after two sweeps on G^k), with I^{k-1}_k transfer by injection (cf. Section A.4) and I^k_{k-1} by linear interpolation, and with "convergence" on G^k defined as obtained after the first sweep following a return from G^{k-1}. A precise count shows Step (ii) to require $30n + O(n^{1/2})$ operations, where n is the number of points in G^M. Thus, the total number of operations is

$$\left(1 + \frac{1}{4} + \frac{1}{16} + \cdots\right) 30n + O(n^{1/2}) \leqslant 40n + O(n^{1/2}).$$

Incidentally, none of these operations is a full multiplication: only additions and shifts (multiplications or divisions by 2 or 4) are used. The theoretical W_M for this problem (sixth line in Table 1) amounts to only $17.5n$ operations, since it ignores interpolation work ($10.3n$ operations in the above procedure) and allows nonintegral numbers of sweeps and cycles. In fact, numerical tests showed the above algorithm to yield residuals considerably below the truncation errors. (The only cases in which the residuals approached 50% of the truncation errors were cases with high smoothness, in which the correct MLAT discretization would be different; namely, of higher order. (Cf. Section 8 and the remark following formula (A.7).) If one is interested in still smaller residuals, then another MG correction cycle can be added to the above algorithm. This will require $20n$ more operations and will make the residuals much smaller than (typically 2% of) the truncation errors. For $n \geqslant 500$, a program implementing the above algorithm runs less than $40n$ microseconds on CDC CYBER 173.

6.4. *Numerical Experiments*: *Elliptic Problems.* A typical numerical experiment is shown in Appendix B, including the FORTRAN program and the computer output. The output shows a multi-grid convergence factor

$$\mathring{\mu} = \left(\frac{.009051}{28.1}\right)^{1/12.92} = .537$$

which is close to, and slightly faster than, the theoretical value $\mathring{\mu} = .595$ shown in Table 1.

Many numerical experiments with various elliptic difference equations in various domains were carried out at the Weizmann Institute in 1970–1972, with the collaboration of Y. Shiftan and N. Diner. Some representative results were reported in [2], and many others in [11]. These experiments were made with other variants of the multigrid algorithm (variants A and B), but their convergence factors agree with the same theoretical rates $\mathring{\mu}$. The experiments with equations of the form $aU_{xx} + cU_{yy}$, with $a \gg c$, showed poor convergence rates, since the relaxation scheme used was Gauss-Seidel, and not the appropriate line relaxation (cf. Section 3.1). Some of these rates were better than predicted by the mode analysis, because the grids were not big enough to show the worst behavior. The convergence rates found in the experiments with the biharmonic equation were also rather poor (although nicely bounded, independently of the grid size), again because we used Gauss-Seidel relaxations and injections

instead of the appropriate schemes (cf. Sections 3.3 and A.4). All these points were later clarified by mode analyses, which fully explain all the experimental results. In solving the stationary Navier-Stokes equations, as reported in [2], SOR instead of CSOR was employed (cf. Table 1 above), and an additional over-simplification was done by using, in each multi-grid cycle, values of the nonlinear terms from the previous cycle, instead of using the FAS scheme (Section 5).

Nevertheless, these experiments did clearly demonstrate important features of the multi-grid method: The rate of convergence was essentially insensitive to several factors, including the shape of the domain Ω, the right-hand side F (which has some influence only at the first couple of cycles; cf. Section A.2) and the finest mesh-size h_M (except for mild variations when h_M is large). The experiments indicated that the order I of the interpolations I_{k-1}^k should be the order of the elliptic equation, as shown in Section A.2 below. (Note that in [2] the order was defined as the degree l of the polynomial used in the interpolation, whereas here $I = l + 1$.)

More numerical experiments are now being conducted at the Weizmann Institute in Israel and at IBM Research Center in New York, and will be reported elsewhere. In this article they are briefly mentioned in Sections 6.3, 7.2, A.4, A.6, A.7. We will separately report here only an extreme case of the multi-grid tests—the solution of transonic flow problems.

6.5. *Numerical Experiments: Transonic Flow Problems.* These experiments were started in 1974 at the Weizmann Institute with J. L. Fuchs, and recently conducted at the NASA Langley Research Center in collaboration with Dr. Jerry South while the present author was visiting the Institute for Computer Applications in Science and Engineering (ICASE). They are preliminarily reported in [12], and will be further reported elsewhere. One purpose of this work was to examine the performance of the multi-grid method in a problem that is not only nonlinear, but more significantly, is also of mixed (elliptic-hyperbolic) type and contains discontinuities (shocks).

We considered the transonic small-disturbance equation in conservation form

$$[(K - \bar{K}\phi_x)\phi_x]_x + c\phi_{yy} = 0, \tag{6.7}$$

for the velocity disturbance potential $\phi(x, y)$ outside an airfoil. Here $K = (1 - M_\infty^2)/\tau^{2/3}$, $\bar{K} = \frac{1}{2}(\gamma + 1)M_\infty^2$, M_∞ is the free-stream Mach number, and $\gamma = 1.4$ is the ratio of specific heats. τ is the airfoil thickness ratio, assumed to be small. $c = 1$, unless the y coordinate is stretched. The airfoil, in suitably scaled coordinates, is located at $\{y = 0, |x| \leqslant \frac{1}{2}\}$, and we consider nonlifting flows, so that the problem domain can, by symmetry, be reduced to the half-plane $\{y \geqslant 0\}$, with boundary conditions

$$\phi(x, y) \longrightarrow 0 \quad \text{as} \quad x^2 + y^2 \longrightarrow \infty, \tag{6.8}$$

$$\phi_y(x, 0) = \begin{cases} 0 & \text{for } |x| > \frac{1}{2}, \\ F'(x) & \text{for } |x| < \frac{1}{2}, \end{cases} \tag{6.9}$$

where $\tau F(x)$ is the airfoil thickness function which we took to be parabolic. Equation (6.7) is of hyperbolic or elliptic type depending on whether $K - 2\bar{K}\phi_x$ is negative or positive (supersonic or subsonic).

The difference equations we used were essentially the Murman's conservative scheme ([9]; for a recent account of solution methods, see [8]), where the main idea is to adaptively use upstream differencing in the hyperbolic region and central differencing in the elliptic region, keeping the system conservative. For relaxation we used vertical (y) line relaxation, marching in the stream direction. The multi-grid solution was programmed both in the CS (Section 4) and the FAS (Section 5) modes, with practically the same results. We used cubic interpolation for I_k^{k+1} and injection for I_k^{k-1}.

Local mode analysis of the linearized-frozen difference equations and vertical-forward line relaxation gives the smoothing factor

$$(6.10)\quad \bar{\mu} = \max\left\{\left|\frac{b_+}{b_+ + b_- + ib_-}\right|, \frac{b_+}{2c + b_+}\right\}, \qquad b_\pm(x) = K - 2\bar{K}\phi_x\,(x \pm h/2),$$

at elliptic (subsonic) points, and $\bar{\mu} = 0$ at supersonic points. We were interested in cases where $K < 1$ and $\phi_x \geqslant 0$, and hence, in smooth elliptic regions ($b_+ \approx b_-$) without coordinate stretching we get $\bar{\mu} \approx 1/|2 + i| = 0.45$ and $\mathring{\mu} = \bar{\mu}^{3/4} = 0.55$.

The actual convergence factors, observed in our experiments with moderately supercritical flows ($M_\infty = 0.7$ and $M_\infty = 0.85$, $\tau = 0.1$) on a 64×32 grid, were $\mathring{\mu} = 0.52$ to 0.53, just slightly faster than the theoretical value. (See detailed output in [12]. The work count in [12] is slightly different, counting also the work in the I_{k+1}^k transition.)

For highly supercritical flows ($M_\infty = 0.95$, $\tau = 0.1$) the MG convergence rate deteriorated, although it was still three times faster than solution by line relaxation alone. The worse convergence pattern was caused by a conceptual mistake in our test for slow convergence (cf. Section A.6). For switching to a coarse grid in a transonic problem, it is not enough that slow reduction per relaxation sweep is shown in the residual norm. Slow change per sweep should also be exhibited in the number of supersonic (or subsonic) points. When this extra test was introduced we obtained fast convergence ($\mathring{\mu} \leqslant .73$) even for $\mu = .98$. Further improvement may be obtained by including partial relaxation sweeps (see Section A.9) in a narrow region behind the shock, where $b_+ \gg b_-$ so that $\bar{\mu}$ is close to 1. We had certain difficulties in the convergence on the coarsest grids, which may indicate the need for the residual weighting (A.12).

Coordinate stretching, which transforms the bounded computational domain to the full half-plane, gave difference equations that again exhibited slow multi-grid convergence rates. This, too, is explainable by the mode analysis. For example, in the regions where the y coordinate is highly stretched, c in (6.7) becomes very small, and hence, $\bar{\mu}$ in (6.10) is close to 1. The theoretical remedies: alternating-direction line relaxations and partial relaxation sweeps. The latter was tried in one simple situation (stretching only the x coordinate), and indeed restored the convergence rate of the corresponding unstretched case.

7. Nonuniform Grids. Many problems require very different resolution in different parts of their domains. Special refinement of the grid is required near singular points, in boundary layers, near shocks, and so on. Coarse grids (with higher approxi-

mation order) should be used where the solution is smooth, or in subdomains far from the region where the solution is accurately needed, etc. A general method for locally choosing mesh-sizes and approximation orders is described in Section 8. An important feature of the method is *adaptivity*: the grid may change during the solution process, adapting itself to the evolving solution. In this section, we propose a method of organizing nonuniform grids so that the local refinement is highly flexible. The main idea is to let the sequence of uniform grids $G^0, G^1, \ldots, G^M$ (cf. Section 2) be open-ended and noncoextensive (i.e., finer levels may be introduced on increasingly smaller subdomains to produce higher local refinement, and coarser levels may be introduced on increasingly wider domains to cover unbounded domains), and, furthermore, to let each of the finer levels be defined in terms of suitable local coordinates. The multi-grid FAS process remains practically as before (Section 5), with similar efficiency. Also discussed is a method which employs this grid organization for "segmental refinement", a multi-grid solution process with substantially reduced storage requirement.

7.1. *Organizing Nonuniform Grids.* How are general nonuniform grids organized for actual computations? There are two popular approaches: One, usually used with the *finite element method*, is to keep the entire system very flexible, allowing each grid point to be practically anywhere. This requires a great deal of bookkeeping: grid points' locations and pointers to neighbors need to be stored; sweeping over the grid is complicated; obtaining the coefficients of the difference equations (or the local "stiffness") may require lengthy calculations, especially where the grid is irregular; and these calculations should be repeated each relaxation sweep, or else additional memory areas should be allocated to store the coefficients. Also, it is more difficult to organize a multi-grid solution on a completely general grid (see, however, Sections 7.3 and A.5), and complete generality is not necessary for obtaining any desired refinement pattern.

Another approach for organizing a nonuniform grid is by a *coordinate transformation*, with a uniform grid being used over the *transformed* domain. On such grids, topologically still rectangular, the multi-grid method can be implemented in the usual way, the lines of G^{k-1} being every other line of G^k. Decisions (stopping criteria, residual weighting, relaxation mode and relaxation directions) should be based on the transformed difference equations. Very often, however, coordinate transformation does not offer enough flexibility. A local refinement is not easy to produce, unless it is a one-dimensional refinement, or a tensor product of one-dimensional refinements. The difficulties are enlarged in adaptive procedures, where it should be inexpensive to change *local* mesh-sizes several times in the solution process. Moreover, the transformation usually makes the difference equation much more complicated (requiring additional storage for keeping coefficients, or additional work in recomputing them every sweep), especially when the transformation does become sophisticated (i.e., adaptive, and not merely a product of one-dimensional transformations), and in particular if higher-order approximations should be used in some or all subdomains.

Thus, be it in the original or in some transformed domain, one would like to have a convenient system for local refinements, with minimal bookkeeping and efficient methods for formulating and solving difference equations. The following system is

proposed (and then generalized, in Sections 7.3, 7.4):

A nonuniform grid is a *union of uniform subgrids,* $G^0, G^1, \ldots, G^M$, with corresponding mesh-sizes $h_0, h_1, \ldots, h_M$. Usually $h_k : h_{k+1} = 2 : 1$ and every other grid line of G^{k+1} is a grid line of G^k. Unlike the description in Section 2, however, the subgrids are not necessarily extended over the same domain. The domain of G^{k+1} may be only part of the domain of G^k (but *not* vice versa). Thus we may have different levels of refinement at different subdomains.

For problems on a bounded domain Ω, several of the first (the coarsest) subgrids may extend to the entire domain Ω. That is, they do not serve to produce different levels of refinement; but they are kept in the system for serving in the multi-grid process of solving the difference equations. G^0 *should be coarse enough* to have its system of difference equations relatively inexpensive to solve (i.e., requiring less than $O(\Sigma n_k)$ operations, where n_k is the number of grid points in G^k. But cf. Section 4.1). The finer subgrids typically extend only over certain subdomains of Ω, not necessarily connected. Generally, G^k *is stretched over those subdomains where the desired mesh-size is* h_k *or less.* Thus, very fine levels (e.g., with $M = 20$, so that $h_M = 2^{-20}h_0$) may be introduced, provided they are limited to suitably small subdomains.

Such a system is very flexible, since grid refinement (or coarsening) is done by extending (or contracting) uniform subgrids. There are several possible ways of storing functions on a (possibly disconnected) uniform grid, allowing for easy grid changes. For example, each string (i.e., connected row or column) of function values can be stored separately, at an arbitrary place in one big storing area, with a certain system of pointers leading from one string to the next. The extra storage area needed for these pointers is small compared with the area needed for storing the function values themselves. One such system, with subroutines for creating, changing and interpolating between the grids, is now under construction, and is described in [26].

If the (original or transformed) *problem's domain is unbounded*, we usually put suitable boundary conditions on some finite, "far enough" artificial boundary. In the present system, we do not have to decide in advance where to place the artificial boundary: We can extend (or contract) the coarsest subgrid(s) as the solution evolves. Moreover, we can add *increasingly coarser levels* ($G^{-1}, G^{-2}, \ldots$) to cover increasingly wider domains, if required by the evolving solution. In this way, we may reach computational domains of large diameter R, by adding only $O(\log R)$ grid points (assuming the desired mesh-size, out at distance r, is proportional to r, or larger. This should usually be the case, especially if appropriate higher-order approximations are used at large distances).

There appears to be a certain waste in the proposed system, as one function value may be stored several times, when its grid point belongs to several levels G^k. This is not the case. First, because the amount of such extra storage is small (less than 2^{-d} of the total storage; see (4.6)). Moreover, the stored values are exactly those needed for the multi-grid process of solution: In fact, in that process, the values stored for different levels at the same grid point are not identical; they only converge to the same value as the process proceeds.

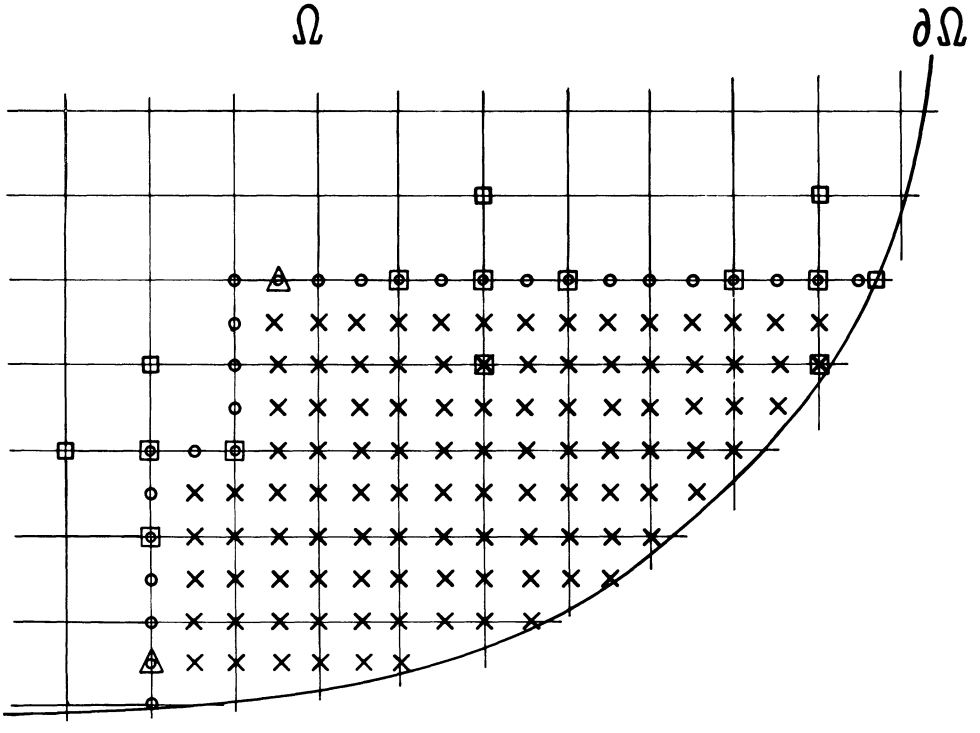


FIGURE 4. *Example of Nonuniform Grid*

A section of the domain Ω and its boundary $\partial\Omega$ is shown, covered with a coarser grid G^k (line intersections) and a finer grid G^{k+1} (crosses and circles). For the case of 5-point (or 9-point "box") difference equations, G^{k+1} inner points are marked with crosses, its outer points with circles. (For convenient interpolation, outer points should lie on G^k lines.) At outer points belonging to G^k, the converged solution satisfies the G^k difference equations, such as the 5-point relations indicated by squares. At other outer points, such as those shown with triangles, the solution is always an interpolation from values at adjacent G^k points. (Note that starting values at outer points should be such that these interpolation relations are satisfied. The FAS interpolation steps will then automatically preserve these relations.)

7.2. *The Multi-Grid Algorithm on Nonuniform Grids.* The following is a description of the modification in the FAS multi-grid algorithm (Section 5) in case of a nonuniform grid with the above structure. The algorithm remains almost the same, except that the difference equations (5.2)–(5.4) are changed to take account of the fact that the levels G^k do not necessarily cover the same domain. Denoting by G^k_m the set of points of G^k which are *inner points* of a finer level G^m (i.e., points where the G^m difference equations are defined†; see Figure 4), the modified form of the difference equations on G^k is

†We use the term "inner", and not "interior", because these points may well be boundary points. Indeed, at boundary points difference equations *are* defined, although they are of a special type, called boundary conditions. The only G^m points where G^m difference equations are not defined are points on or near the *internal* boundary of G^m; i.e., the boundary beyond which the level G^m is not defined, but some coarser levels are. If the grid lines of G^k do not coincide with grid lines of G^m, G^k_m is defined as the set of points of G^k to which proper *interpolation* from inner points of G^m is well defined. For $m > M$, G^k_m is empty.

$$L^k U^k = \bar{F}^k, \qquad \Lambda^k U^k = \bar{\Phi}^k, \tag{7.1}$$

where

$$\bar{F}^k = F^k \quad \text{and} \quad \bar{\Phi}^k = \Phi^k \quad \text{in } G^k - G^k_{k+1} \text{ and for } k = M, \tag{7.2}$$

$$\bar{F}^k = F^k_{k+1} \quad \text{and} \quad \bar{\Phi}^k = \Phi^k_{k+1} \quad \text{in } G^k_{k+1}, \tag{7.3}$$

$$F^k_m = I^k_m(\bar{F}^m - L^m u^m) + L^k(I^k_m u^m), \tag{7.4}$$

$$\Phi^k_m = I^k_m(\bar{\Phi}^m - \Lambda^m u^m) + \Lambda^k(I^k_m u^m). \tag{7.5}$$

F^k and Φ^k, as in Section 2, are the G^k approximation to the original right-hand sides F and Φ, respectively.

Observe that, by (7.2)–(7.3), each intermediate level G^k plays a double role: On the subdomain where the finer subgrid G^{k+1} is not defined, G^k plays the role of the finest grid; and the difference equation there is an approximation to the original differential equation. At the same time, on the subdomain where finer subgrids are present, G^k serves for calculating the coarse-grid correction. These two roles are not confused owing to the FAS mode, in which the correction v^k is only implicitly computed, its equation being actually written in terms of the full approximation u^k. In other words, F^k_m may be regarded as the usual G^k right-hand side (F^k), corrected to achieve G^m accuracy in the G^k solution. Indeed

$$F^k_m - I^k_m \bar{F}^m = L^k(I^k_m u^m) - I^k_m(L^m u^m), \tag{7.6}$$

which is the G^m approximation to the G^k truncation error.

The only other modification required in applying Cycle C to nonuniform grids is in the convergence switching criteria. See Section A.10.

When converged, the solution so obtained satisfies Eqs. (2.2) in the inner part of $G^k - G^k_{k+1}$ ($k = 0, 1, \ldots, M$). On *outer* (i.e., noninner) points the solution *automatically* satisfies either a coarser-grid difference equation (if the point belongs to a coarser grid) or a coarser-grid interpolation relation (see Figure 4). Note that, in this procedure, *difference equations should be defined on uniform grids only.* This is an important advantage. Difference equations on equidistant points are much simpler, more accurate. The basic weights for each term (e.g., the weights (1, −2, 1) for the second-order approximation to $\partial^2/\partial x^2$) can be read from small standard tables; whereas on a general grid those weights should be recomputed (or stored) separately for each point, and they are very complicated for high-order approximations.

Another advantage is that the *relaxation sweeps, too, are on uniform grids only.* This simplifies the sweeping, and is particularly important where symmetric and alternating-direction sweeps of line relaxation are required (cf. Section 3).

Numerical experiments indicate that the typical multi-grid convergence factors, measured by the overall error reduction per work unit and predicted by local mode analysis (cf. Section 6), are retained in multi-grid solutions on nonuniform grids. The work unit, though, is somewhat different: It is the computational work of one sweep on *all* levels, not only on G^M, since here G^M may make up only a small part of the points of the final nonuniform grid.

7.3. *Finite-Element Generaliztion.* The structure and solution process outlined above can be generalized in various ways. An important generalization is to employ piecewise uniform, rather than strictly uniform, levels.

Quite often, especially in problems that use finite-element discretizations, the "basic" partition G^0 (e.g., the coarsest triangulation) of the domain is a nonuniform one, but one which is particularly suitable for the geometry of the problem. Finer levels $G^1, G^2, \ldots,$ are defined as uniform refinements of that basic level; e.g., $h_k = 2^{-k}h_0$; so that h_k is constant within each basic element.

Having defined the levels G^k in this manner, the rest may in principle be as before: The actual composite grid may use only certain, arbitrary portions of each level; i.e., the actual subgrids G^k need not be coextensive, allowing for adaptive refinements. Coarser levels $(G^{-1}, G^{-2}, \ldots)$ may be added if the basic level G^0 is not coarse enough for full-speed multi-grid solution. (Although there is no general algorithm for coarsening a nonuniform G^0, and usually G^0 is coarse enough). Data structures, similar to the uniform case may be used, but should be constructed separately for each basic element (or each set of identical basic elements).

The multi-grid algorithm is the same as in Section 7.2. The discrete equations are thus defined separately for each level. The reproduction of these equations during relaxation is not as convenient as in the strictly uniform case, but still, in the interior of any basic element the equations can readily be read from fixed tables, one table for each set of identical basic elements.

7.4. *Local Transformations.* Another important generalization of the above structure is to subgrids which are defined each in terms of another set of variables. For example, near a boundary or an interface, the most effective local discretizations are made in terms of local coordinates in which the boundary (or interface) is a coordinate line. In particular, with such coordinates it is easy to formulate high-order approximations near the boundary; or to introduce mesh-sizes that are different across and along the interface (or the boundary layer); etc. Usually it is easy to define suitable *local* coordinates, and uniformly discretize them, but it is more difficult to patch together all these local discretizations.

A multi-grid method for patching together a collection of local grids $G_1, G_2, \ldots, G_m$ (each being uniform in its own local coordinates) is to relate them all to a basic grid G_0, which is uniform in the global coordinates and stretches over the entire domain. The relation is essentially as above (Section 7.2); namely, finite-difference equations are separately defined in the inner points of each grid, and the FAS multi-grid process automatically combines them together through its usual interpolation periods.

A Remark: To a given collection of local grids we may have to add intermediate grids to obtain fast multi-grid convergence. That is, if a given local grid G_k is much finer than the basic grid G_0, we have to add increasingly coarser grids, all of them uniform grids in the same local coordinates, such that the coarsest of them has a mesh-size which is (in the global coordinates) nowhere much smaller than the basic mesh-size h_0. Similarly, if the basic global grid G_0 is not coarse enough, the usual multi-grid

sequence of global grids $G^0, G^1, \ldots, G^M = G_0$ should be introduced. Thus, in each set of coordinates we will generally have several grids.

Such a system offers much flexibility. Precise treatment of boundaries and interfaces by the global coordinates is not required. The local coordinates may be changed in the course of computations, e.g., to fit a moving interface. New sets of local coordinates may be introduced (or deleted) as the need arises.

The data structure required for creating, changing and employing such grids is basically again just any data structure suitable for changeable uniform grids. This, however, should be supplemented by tables for the local transformations, such that one can efficiently (i) reproduce the local difference equation, and (ii) interpolate from local to global grid points, and vice versa. See [26].

7.5. *Segmental Refinement.* The multi-grid algorithm for nonuniform grids (Section 7.2) can be useful even in the case of uniform grids, if the computer memory is not sufficiently large to store the finer levels.

"Segmental refinement" is the refinement of one subdomain at a time. To see why and how this is possible, observe that with the FAS mode (Section 5) the full solution u^M is obtained on all grids. But on a coarser grid G^k, the u^M solution satisfies a "corrected" difference equation, with $\overline{F}^k = F_M^k$ replacing F^k. It is therefore not necessary to keep the fine grid, once F_M^k has been computed.

The corrected forcing function F_M^k can be computed by segmental refinement. Refining only one subdomain, one can use the algorithm above (Section 7.2) to obtain a multi-grid solution, including the values of F_M^k in the refined subdomain. Keeping this F_M^k (instead of F^k), one can then discard this refinement, and refine a second subdomain. And so on, through a sequence of subdomains covering the entire domain.

Since subsequent subdomain refinements change the solution everywhere, some further changes are also due in the values of F_M^k on former subdomains. However, at points inner to (and few meshes away from the boundary of) such a former subdomain, these further changes are much smaller than the first correction $F_M^k - F^k$, since they represent changes in the G^k truncation error due to small smooth changes in the solution, while the first correction represents the full G^k truncation error. Thus, if the refinement segments are chosen so that neighboring segments overlap (several mesh intervals into each other), then the further corrections may be ignored. If extra accuracy is desired, another cycle of segmental refinements may be performed. Another way of viewing this technique is to observe that the roll of the finer levels, relative to the coarser ones, is only to liquidate high-frequency error components which cannot be "seen" on the coarser grids. These components have a short (just few mesh-sizes) coupling range, and can therefore be computed at any point by refining only few neighboring meshes.

With this technique one can *operate the multi-grid algorithm almost in its full efficiency, using a storage area which is much smaller than that of the finest grid.* This has been confirmed by preliminary (one-dimensional) numerical tests.

In principle, the required storage area can be reduced to only a constant cube, of J^d locations, on each level (where even $J = 15$ probably offer enough overlap without

substantial reduction in efficiency). Thus, *the overall storage requirement can in principle be reduced to*

$$2J^d\left\{1 + \log \frac{R}{h}\Big/\log J\right\}$$

locations, where h is the finest mesh-size and R is the diameter of the domain. No external memory is needed.

8. Adaptive Discretization Techniques. The previous section described a flexible data structure and solution process which facilitate implementation of variable mesh-sizes h. The difference equations in that process are always defined at inner points of uniform subgrids, which make it easy to employ high and variable approximation orders p. How, then, are mesh-sizes and approximation-orders to be chosen? Should boundary layers, for example, be resolved by the grid? What is their proper resolution? Should we use high-order of approximation at such layers? How to detect such layers automatically? In this section we propose a general framework for automatic selection of h and p in a (nearly) optimal way. In Section 9, we will study some special cases, and show how this proposed system automatically resolves or avoids resolving a thin layer, depending on the alleged goal of the computations.

8.1. *Basic Principles.* We will treat the problem of selecting the discretization parameters h and p (and possibly other parameters, see Section 8.4) as an optimization problem: We will seek to minimize a certain error estimator E, subject to a given amount of computational work W. (Or, equivalently, minimize the work W to obtain a given level E of the error estimator. We will see that the actual control quantity is neither E nor W, but their rate of exchange.) It is important, however, to promptly emphasize that we should not take this optimization too pedantically; it is enough, for instance, to obtain E which is one or two orders of magnitudes larger than the minimum (or, equivalently, to invest work W which is by some fraction more than theoretically needed. Note below that $\log(1/E_{\min})$ is usually proportional to W). Full optimization is not our purpose, is enormously harder and, in fact, is self-defeating, since it requires too much computational work to be invested in controlling h and p. We will aim at having the control work much smaller than the actual numerical work W, using the optimization problem only as a loose directive for sensible discretization.

The Error Estimator E is a functional that estimates, for any given numerical approximation, the overall error in solving the differential boundary-value problem. In principle, such a functional should be furnished whenever a problem is submitted for numerical solution; in practice, it is seldom provided. To have such an estimator depends on having a clear and well-defined idea about *the goal of the computations,* i.e., an idea about what error norm we intend to minimize. Given the goal, even roughly, we can usually formulate E quite easily. We assume that the numerical approximation U^h is in some suitable neighborhood of the true solution (this is a necessary and justifiable assumption; see Section 8.2), so that E can be written as a linear functional

$$E = \int_\Omega G(x)\tau(x)\,dx. \tag{8.1}$$

$\tau(x)$ is a local estimate of the *truncation error*, i.e., the error by which the numerical solution U^h fails to satisfy the differential equation $LU = F$; or, more conveniently, the error by which the differential solution U fails to satisfy the discrete equation $L^h U^h = F^h$. That is,

$$\tau(x) = |LU(x) - L^h U(x)|. \tag{8.2}$$

$G(x)$ is the nonnegative "*error-weighting function*" (or distribution), through which the goal of the computations should be expressed.

The choice of G can be crude. In fact, multiplying G by a constant does not change our optimization problem. Also, we can make large errors, up to one or two orders of magnitudes, in the relative values of G at two different points, since we are content in having E only to that accuracy. What matters is only large changes in G, e.g., near boundaries. For example, if we have a uniformly elliptic problem of order m, and if we are interested in computing good approximations to U and its derivatives up to order l and up to the boundary, then a suitable choice is

$$G(x) = d_x^{m/2-l}, \tag{8.3}$$

where d_x is the distance of x from the boundary. (The formula should be suitably modified near a boundary corner.) This and similar choices of G are easily found by local one-dimensional crude analysis of the relation between a perturbation in the equations and the resulting perturbation in the quantity we wish to approximate. Even though crude, such choice of G would specify our goal much closer than people usually bother to. Moreover, we can change G if we learn that it fails to properly weigh a certain region of the computation; it can serve as a convenient control, conveying our intentions to the numerical discretization and solution.

The Work Functional W. In solving the discrete equations by the multi-grid method, the main overall computational work is the number of Work Units invested in relaxations, times the amount of computations in each Work Unit (see Section 6). If the discretization and relaxation schemes are suitable, the number of Work Units is almost independent of the relaxation parameters h and p. (See e.g., the rate $\mathring{\mu}$ for $\Delta_h^{(4)}$ vs Δ_h in Table 1 above.) Since for our optimization problem we need W only up to a multiplicative constant, we can take into account only the amount of computations in a single Work Unit, i.e., the work in one relaxation sweep over the domain. The local number of grid points per unit volume is $h(x)^{-d}$, and the amount of computation at each grid point is a function $w(p(x))$, where $p(x)$ is the local order of approximation. Hence, we can regard the work functional as being

$$W = \int_\Omega \frac{w(p(x))}{h(x)^d}\, dx. \tag{8.4}$$

Global Optimization Equations. Treating the discretization parameters as spatial variables, $h(x)$ and $p(x)$, the Euler equations of minimizing E for fixed W are

$$\frac{\partial E}{\partial h(x)} + \lambda \frac{\partial W}{\partial h(x)} = 0, \tag{8.5a}$$

$$\frac{\partial E}{\partial p(x)} + \lambda \frac{\partial W}{\partial p(x)} = 0, \tag{8.5b}$$

where λ is a constant (the Lagrange multiplier). It is easily seen that λ is actually the marginal rate of exchange between work and optimal accuracy, i.e.,

$$\lambda = -\frac{dE_{\min}}{dW} = E\,\frac{d \log 1/E}{dW}, \tag{8.6}$$

and the meaning of (8.5) is that we cannot lower E by trading work (e.g., by taking smaller h at one point and larger at another, keeping W constant, or trading a change in h with a change in p).

Equations (8.5) make some essential simplifications in the optimization problem: They regard h and p as defined at all points $x \in \Omega$. Also, h and p are assumed to be continuous variables, whereas in practice they are discrete. (p should be a positive integer, in some schemes a positive even integer. Values of h are restricted by some grid-organization considerations.) These simplifications are crucial for our approach, and they are altogether justified by the fact that we are content in having only an approximate optimum. The practical aspect, of choosing permissible h and p close to the solution of (8.5), is discussed in Section 8.3. One restriction we should, however, take into account in the basic equations, namely, the restriction

$$p_0 \leqslant p(x) \leqslant p_1(x). \tag{8.7}$$

Without such a restriction, the optimization equations may give values of p which cannot be approximated by permissible values. p_0 is usually 1 or (in symmetric schemes) 2. The upper bound p_1 may express the highest feasible order due to round-off errors; or the highest order for which we actually have appropriate (stable) discretization formulae, with special such restriction near boundaries (hence the possible dependence of p_1 on the position x). With this restriction, Euler's equation (8.5b) should be rewritten as

$$\frac{\partial E}{\partial p(x)} + \lambda \frac{\partial W}{\partial p(x)} \begin{cases} \geqslant 0 & \text{if } p(x) = p_0, \\ = 0 & \text{if } p_0 < p(x) < p_1(x), \\ \leqslant 0 & \text{if } p(x) = p_1(x). \end{cases} \tag{8.8}$$

Local Optimization Equations. Substituting (8.1) and (8.4) into (8.5a) and (8.8), we get the following equations at each point $x \in \Omega$:

$$G\frac{\partial \tau}{\partial h} - \frac{\lambda d w(p)}{h^{d+1}} = 0, \tag{8.9a}$$

$$G\frac{\partial \tau}{\partial p} + \frac{\lambda w'(p)}{h^d} \gtreqless 0, \tag{8.9b}$$

where the equality-inequality sign, in (8.9b) and hereinafter, corresponds to the three cases introduced in (8.8). In principle, the pair of equations (8.9) determines, for each $x \in \Omega$, the local optimal values of the pair (h, p), once λ is given.

Thus λ is our global control parameter. Choosing larger λ, we will get an optimized grid with less work and poorer accuracy; lowering λ, we invest more work and get

higher accuracy. For each λ, however, we get (approximately) the highest accuracy for the work invested. In principle λ should be given by whoever submits the problem for numerical solution; i.e., he should tell at what rate of exchange he is willing to invest computational work for additional accuracy (see (8.6)). In practice this is not done, and λ usually serves as a convenient control parameter (see Sections 8.2 and 8.3).

To compute h and p from (8.9) we should know the behavior of τ as a function of h and p. Generally,

$$\tau(x, h, p) \approx t(x, p)h^p, \tag{8.10}$$

where $t(x, p)$ depends on the equations and on the solution. Since it is assumed that all our numerical approximations are in some neighborhood of the solution (see Section 8.2), we may assume that the truncation-error estimates, automatically calculated by the multi-grid processing (see (5.7), for example), give us local estimates for $t(x, p)$. In practice, we never actually solve (8.9), but use these relations to decide upon changes in h and p (see Section 8.3), so that we need to estimate $\tau(x, h, p)$ only for h and p close to the current $h(x)$ and $p(x)$.

In *finite-element formulations* the differential problem is often given as a problem of minimizing a functional $A(U)$. Thus the natural discretization optimization problem is to minimize $A(U^k)$ in a given amount of work W, where the space S^k of approximations U^k is not fixed. This can be translated to practical criteria for adapting S^k. (See end of Section 8.3.)

8.2. *Continuation Methods.* Continuation methods are generally used in numerical solutions of nonlinear boundary value problems. A certain *problem-parameter*, γ say, is introduced, so that instead of a single isolated problem we consider a continuum of problems, one problem $P(\gamma)$ for each value of γ in an interval $\gamma_0 \leqslant \gamma \leqslant \gamma_*$, where $P(\gamma_0)$ is easily solvable (e.g., it is linear), and $P(\gamma_*)$ is the target (the given) problem. The continuation method of solution is to advance γ from γ_0 to γ_* in steps $\delta\gamma$. At each step we use the final solution of the previous step (or extrapolation from several previous steps) as a first approximation in an iterative process for solving $P(\gamma)$. The main purpose of such continuation procedures is to ensure that the approximations we use in the iterative process are always "close enough" to the solution (of the current $P(\gamma)$), so that some desirable properties are maintained. Usually γ is some natural physical parameter (the Reynolds number, the Mach number, etc.) in terms of which either the differential equations or the boundary conditions, or both, are expressed.

The continuation process is not a waste, for several reasons. In many cases, the intermediate problems $P(\gamma)$ are interesting by themselves, since they correspond to a sequence of cases of the same physical problem. More importantly, in solving nonlinear discretized problems the continuation process is not only a method of computing the solution, but also, in effect, the only way to *define* the solution, i.e., the way to select one out of the many solutions of the nonlinear algebraic system. The desired solution is *defined* as the one which is obtained by continuous mapping from $[\gamma_0, \gamma_*]$ to the solution space with a given solution at γ_0 (e.g., the single solution, if $P(\gamma_0)$ is linear). By the continuation process, we keep every intermediate numerical solution in the vicinity of a *physical* solution (to an intermediate problem), hence the target numerical

solution is, hopefully, near the target physical solution, and is not some spurious solution of the algebraic system. Thus, although sometimes we may get away without a continuation process (simply because a starting solution is "close enough", so that the continuation may be done in just one step), in principle a continuation process *must* be present in any numerical solution of nonlinear problems. Moreover, such a process is usually inexpensive, since it can be done with crude accuracy, so that its intermediate steps usually total less computational work than the final step of computing an accurate solution to $P(\gamma_*)$.

A continuation process is necessary, in principle, not only for nonlinear problems but also for linear problems with grid adaptation. In fact, when h or p are themselves unknown, the discrete problem is nonlinear, even if the differential problem is linear.

In our system, a continuation process with crude accuracy and little work is automatically obtained by selecting a large value for the control parameter λ (cf. Section 8.1). Then, in the final step ($\gamma = \gamma_*$), λ is decreased to refine the solution. Thus, the overall process may be viewed as *a multi-grid process of solution, controlled by the two parameters* γ *and* λ.

The most efficient way of changing γ is probably to change it as soon as possible (e.g., when the multi-grid processing exhibits convergence to a crude tolerance), and to control the step-size $\delta\gamma$ by some automatic procedure, so that $\delta\gamma$ is sharply decreased when divergence is sensed (in the multi-grid processing), and slowly increased otherwise.

In changing γ it is advisable to keep the residuals as smooth as possible, since higher-frequency components are more expensive to liquidate (lower components being liquidated on coarser grids). Thus, for example, if a boundary condition should be changed while changing γ, it is advisable to introduce this change into the system at a stage when the algorithm is working at the coarsest level.

γ-Extrapolation. In some cases the given problem ($\gamma = \gamma_*$) is much too difficult to solve, e.g., because the differential solution fluctuates on a scale too fine to be resolved. In such cases one is normally not interested in the details of the solution but rather in a certain functional of the solution. It is sometimes possible in such cases to solve the problem for certain values of γ far from γ_*, and to extrapolate the corresponding functional values to $\gamma = \gamma_*$.

8.3. *Practice of Discretization Control.* The main practical restrictions imposed on the theoretical discretization equations (8.9) are the following: The approximation order p should be a positive integer. In many problems p is also restricted to be even, since odd orders are less efficient. The mesh-size function $h(x)$ should be such that a reasonable grid can be constructed with it. Thus, in the grid structure outlined in Section 7.1, h is restricted to be of the form $h = 2^{-k}h_0$, where k is an integer. Also, in the multi-grid discretization method outlined in Section 7.2, any uniform subgrid truly influences the global solution only if it is large enough, i.e., if at least some of its inner points belong also to coarser grids. These discretization restrictions will actually help us in meeting another practical requirement, namely, the need to keep the control work (computer work invested in testing for and affecting discretization reformulations) small compared with the numerical work (relaxation sweeps and interpolations).

The practical adaptive procedure is proposed to be generally along the following lines:

A. *Testing.* In the multi-grid solution process (possibly incorporating a continuation process), at some natural point we get an estimate of the decrease in the error estimator E introduced by the present discretization parameters. For example, in FAS Cycle C (see its flowchart in Figure 2), at the point where new $\overline{F}^k$ is computed, the quantity

$$-\Delta E = G\,|\overline{F}^k - I_{k+1}^k \overline{F}^{k+1}| \tag{8.11}$$

at each point may serve as a local estimate for the decrease in E per unit volume (cf. (8.1) and (5.7)), owing to the refinement from h_k to h_{k+1}. Each such decrease in E is related to some additional work ΔW (per unit volume). For example, the refinement from h_k to h_{k+1} requires the additional work

$$\Delta W = \frac{w(p)}{h_{k+1}^d} - \frac{w(p)}{h_k^d} \quad \text{(per unit volume).} \tag{8.12}$$

Hence we compute the ratio of exchanging accuracy per work $Q = -\Delta E/\Delta W$. At regions where this ratio is much bigger than λ (the control rate of exchange; cf. Section 8.1) we say that the present parameter (h_{k+1} in the example) is highly profitable, and it is worth trying to further refine the discretization (e.g., introduce there the subgrid G^{k+2} with $h_{k+2} = h_{k+1}/2$). At regions where Q is much smaller than λ, we may coarsen the discretization (abolish the G^{k+1} subgrid).

Extrapolated Tests. More sophisticated tests may be based on assuming the truncation error to have some form of dependence on h and p, such as (8.10) above. Instead of using ΔE and ΔW at the previous change (from h_k to h_{k+1}, in the above example) we can then anticipate the corresponding values $\overline{\Delta E}$ and $\overline{\Delta W}$ at the next change (from h_{k+1} to h_{k+2}), which are the more appropriate values in testing whether to make that next change. Thus, in the above example, assuming (8.10) and $h_{k+2} = h_{k+1}/2 = h_k/4$, we get $\overline{\Delta E} = 2^{-p}\Delta E$, $\overline{\Delta W} = 2^d \Delta W$, and hence

$$\overline{Q} = \frac{-\overline{\Delta E}}{\overline{\Delta W}} = 2^{-p-d} Q = \frac{h_{k+1}^d G|\overline{F}^k - I_{k+1}^k \overline{F}^{k+1}|}{w(p)(2^d - 1)2^p}. \tag{8.13}$$

The extrapolated ratio $\overline{Q}$ is used in testing for grid changes. This may seem risky, since it depends on assuming (8.10). But in fact there is no such risk, because we can see from (8.13) that testing with $\overline{Q}$ is not that much different from testing with Q. (In fact, if p is constant, testing with $\overline{Q}$ is equivalent to testing with Q against another constant λ.) And the test with Q does not presume (8.10); it only assumes that the finer (G^{k+1}) approximation is considerably better than the coarser one, so that their difference roughly corresponds to an added accuracy due to the refinement. Note also that the multi-grid stopping criteria ((A.17) or (A.20) in Appendix A) are precisely such that Q can be reliably computed from the final approximation.

B. *Changing the Discretization.* The desired grid changes are first just recorded (e.g., incidentally to the stage of computing $\overline{F}^k$) and only then they are simultaneously introduced, taking into account some organizational and stabilizational considerations:

A change (e.g., refinement) is introduced only if there is a point where the change is "overdue" (e.g., a point where $\bar{Q} > 10\lambda$). Together with such a point the change is then also introduced at all neighbor (and neighbor of neighbor, etc.) points where the change is "due" (e.g., where $\bar{Q} > 3\lambda$). The changed subgrid (G^{k+2} in the above example) is then augmented as follows: (i) Around each new grid point we add extra points, if necessary, so that the grid point (corresponding to a G^{k+1} point where a refinement was due) becomes an *inner* point (cf. Section 7.2) in the new subgrid (G^{k+2}). (ii) Holes are filled; that is, if, on any grid line, a couple of points are missing in between grid points, then the missing points are added to the grid.

The control work in this system is negligible compared with, say, the work of relaxing over G^{k+1}, because: (i) The tests are made in the transition from G^{k+1} to G^k, which takes place only once per several G^{k+1} relaxation sweeps. (ii) Q is computed and tested only at points of the coarser grid G^k, and at each such point the worl is smaller than the relaxation work per point. (iii) Changing the discretization is itself inexpensive since it is done by extending or contracting uniform grids (cf. Section 7.1), the main work being in interpolating the approximate solution to the new piece of uniform subgrid.

Practical discretization control in finite-element formulations (see closing remark to Section 8.1) is natural: Let $u^k \in S^k$ be the evolving approximate solution, and u^l its projection on a subspace S^l obtained from S^k by (locally) removing some of the parameters (omitting some grid points or lowering the local polynomial degree). It is easy to locally calculate $\Delta A = A(u^l) - A(u^k)$. It is also easy to estimate $\Delta W = W_k - W_l$, where W_j is the work in relaxing over S^j; in a suitable work unit, $W_k - W_l$ may simply equal the number of parameters removed. The removed parameters are "highly profitable" if $Q = \Delta A/\Delta W$ is much larger than λ, in which case it is worth adding more such parameters (more grid points or higher polynomial terms, correspondingly).

8.4. *Generalizations.* In some problems it is not enough to adapt h and p. Some times different increments $h^{(1)}, h^{(2)}, \ldots, h^{(d)}$ should be used at the d different directions, and each $h^{(j)}$ should be separately adapted. Basically the same procedures as above can be used to test and execute, for example, a change from $h^{(j)}$ to $h^{(j)}/2$. More generally, one would like to adapt the local coordinates (cf. Section 7.4), e.g., near discontinuities. Automatic procedures for such adaptation have not been so far developed, but are conceivable.

Other discretization parameters, such as the centering of each term in the difference operator, may be treated adaptively. (In fact, such adaptive discretization is already in use in mixed-type problems, where it was introduced by Murman to obtain stability. See, e.g., [9].) In problems with unbounded domains, the discrete domain may be determined adaptively (with increasingly coarser levels; cf. Section 7.1), using a procedure that decides to extend the domain if the previous extension was highly profitable in terms of $-\Delta E/\Delta W$. In many problems, some terms in the difference operator can altogether be discarded on most levels G^k. In particular, in singularly perturbed problems, the highest-order terms may be kept only on the finest-narrowest levels. Decision can again be made in terms of $-\Delta E/\Delta W$, in an obvious way.

9. Adaptive Discretization: Case Studies. To get a transparent view of the discretization patterns and the accuracy-work relations typical to the adaptive procedures proposed above, we consider now several test cases which are simple enough to be analyzed in closed forms. That is, we consider problems with known solutions and simple behavior of the local truncation errors, and we calculate the discretization functions $h(x)$ and $p(x)$ that would be selected by the local optimization equations (8.9), and the resulting relation between the error estimator E and the computational work W.

9.1. *Uniform-Scale Problems.* A problem is said to have the uniform scale $\eta(x)$ if the local truncation error (8.2) has the behavior

$$\tau(x, h, p) \approx t(x)\left[\frac{h}{\eta(x)}\right]^p \qquad (p_0 \leqslant p \leqslant p_1). \tag{9.1}$$

Such a behavior occurs, for example, when the solution is a trigonometric or exponential function $\exp(\theta \cdot x)$, where θ is either a constant or a slowly varying function (see example in Section 9.2). We will also assume for simplicity that (see (8.4))

$$w(p) = w_0 p^l. \tag{9.2}$$

Usually $l = 1$, since the number of terms in the difference equations, and hence also the amount of computer operations at each grid point, are proportional to p. $l = 2$ is appropriate if we assume that we have to increase the precision of our arithmetic when we increase p. Rescaling W, we can assume that $w_0 = 1$.

Using (9.1)–(9.2) in Eqs. (8.9), we get

$$G\tau = \lambda d p^{l-1} h^{-d}, \tag{9.3a}$$

$$G\tau \log \frac{h}{\eta} + \lambda l p^{l-1} h^{-d} \gtreqless 0. \tag{9.3b}$$

Hence, denoting by $\tilde{p}$ the value of p that satisfies

$$p^{l-1} e^{lp/d} = \lambda^{-1} G t \eta^d e^{-l} d^{-1}, \tag{9.4}$$

we have

$$h = \eta e^{-l/d}, \qquad p = \tilde{p} \quad \text{if } p_0 \leqslant \tilde{p} \leqslant p_1, \tag{9.5a}$$

$$h = (\lambda d p_0^{l-1} \eta^{p_0} t^{-1} G^{-1})^{1/(p_0+d)}, \qquad p = p_0 \quad \text{if } \tilde{p} \leqslant p_0, \tag{9.5b}$$

$$h = (\lambda d p_1^{l-1} \eta^{p_1} t^{-1} G^{-1})^{1/(p_1+d)}, \qquad p = p_1 \quad \text{if } p_1 \leqslant \tilde{p}. \tag{9.5c}$$

Notice that at any point either p or h, but never both, is "adaptive", i.e., dependent of λ. Where p is adaptive ($p_0 \leqslant p = \tilde{p} \leqslant p_1$), h is fixed and each "scale cube" η^d is divided into e^l mesh cells.

Assume now further that the computer precision is unlimited (which is never really the case, but may provide insight), so that $l = 1$ and $p_1 = \infty$. If sufficiently high accuracy is desired, then λ is sufficiently small to have $\tilde{p} > p_0$, so that (9.5a) applies. By (8.1) and (9.3a) this implies

$$E = \lambda d e \int \eta^{-d}\, dx, \tag{9.6}$$

and hence, by (8.6),

$$E = C_0 e^{-W/(ed\int \eta^{-d}\,dx)} = C_0 e^{-c\beta^d W}, \tag{9.7}$$

where β is some average value of the scale $\eta(x)$. In this (idealized) case, E decreases exponentially with W. For realistic W this convergence rate becomes poor when β is very small, as in singularly perturbed problems. In such problems, however, for realistic W (9.5a) no longer applies, and another rate of convergence, independent of β, takes over (see Section 9.3).

9.2. *One-Dimensional Case.* Consider a 2-point boundary-value problem

$$\frac{\eta}{2}\frac{d^2U}{dx^2} + \frac{dU}{dx} = 0 \quad \text{in } 0 < x < 1, \tag{9.8}$$

with constant $\eta > 0$ and with boundary conditions $U(0)$ and $U(1)$ such that the solution is $U = e^{-2x/\eta}$. An elliptic (stable) difference approximation to such an equation can be central for $\eta \geqslant h$ but should be properly directed for $\eta < h$. (The first-order term being the main term, the second-order term should be differenced backward relative to it with approximation order $p' = p - [\log \eta/\log h]$. See [4] and Section 3.2 in [3].) In either case, the truncation error is approximately

$$\tau(x, h, p) = t(x)\left(\frac{h}{\eta}\right)^p, \quad \text{where } t(x) = \frac{1}{2\eta}e^{-2x/\eta}. \tag{9.9}$$

We now choose the error weighting function to be

$$G(x) \equiv 1, \tag{9.10}$$

which would be the choice (see (8.3)) when one is interested in accurate computation of boundary first-order derivatives (corresponding, e.g., to boundary pressure or drag, in some physical models). We again assume no precision limitations, so that $l = 1$ and $p_1 = \infty$. We take $p_0 = 2$ since second-order is no more expensive than first-order approximation. Inserting these into (9.5), we get

$$h = \frac{\eta}{e}, \quad p = \log\frac{1}{2\lambda} - 1 - \frac{2x}{\eta} \quad \text{for } 0 < x \leqslant x_0, \tag{9.11a}$$

$$h = \frac{\eta}{e}e^{2(x-x_0)/(3\eta)}, \quad p = 2 \quad \text{for } x_0 \leqslant x < 1, \tag{9.11b}$$

where

$$x_0 = \frac{\eta}{2}\left(\log\frac{1}{2\lambda} - 3\right). \tag{9.11c}$$

If $x_0 \geqslant 1$, then (9.11a) applies throughout, and hence

$$W = \int_0^1 \frac{p}{h}\,dx = \frac{e}{\eta}\left(\log\frac{1}{2\lambda} - 1 - \frac{1}{\eta}\right), \tag{9.12}$$

$$E = \int_0^1 \tau\,dx = \frac{\lambda e}{\eta} = \frac{1}{2\eta}e^{-\eta W/e - 1/\eta} \tag{9.13}$$

and the condition $x_0 \geqslant 1$ itself becomes, by (9.11c), (9.12),

$$W \geqslant (2 + 1/\eta)e/\eta. \tag{9.14}$$

Thus, if W satisfies (9.14), E converges like (9.13).

9.3. *Singular Perturbation: Boundary-Layer Resolution.* When η is very small, problem (9.8) is singularly perturbed, and its solution has a boundary layer near $x = 0$. The above mesh-size $h = \eta/e$ is too small to be practical. Indeed, in the optimal discretization (9.11), for small η we get small x_0, and an "external region" $x_0 \leqslant x < 1$ is formed where the mesh-size grows exponentially from η/e. The small mesh-size is used only to resolve the boundary layer. In this simplified problem the solution away from the boundary layer (i.e., for $x \gg \eta$) is practically constant, so that indefinitely large h is suitable. Usually h will grow exponentially, as in (9.11b), from $h = \eta/e$ to some definite value suitable for the external region. In the transition region we have $p = 2$, i.e., the minimal order of differencing is used in the region where h changes. This may be useful in practical implementations.

From (9.11) and (9.9) we get for small η

$$W = \int_0^1 \frac{p}{h}\, dx \approx \frac{e}{4}\left(\log \frac{1}{2\lambda}\right)^2, \tag{9.15}$$

$$E = \int_0^1 \tau\, dx = \frac{\lambda e}{2} \log \frac{1}{2\lambda} \approx \left(\frac{e}{4} W\right)^{1/2} e^{-(4W/e)^{1/2}}, \tag{9.16}$$

where the integrals are separately calculated in $(0, x_0)$ and $(x_0, 1)$. Thus, E converges exponentially as a function of $W^{1/2}$ instead of W, but this rate is independent of η and does not deteriorate as $\eta \longrightarrow 0$.

9.4. *Singular Perturbation Without Boundary-Layer Resolution.* To see the effect of choosing different error weighting functions, consider again the above problem (Sections 9.2, 9.3), but with the choice $G(x) = x$. This choice is typical to cases where one is not interested in calculating boundary derivatives of the solution (see (8.3)). We then get

$$\tilde{p} = \log \frac{x}{2\lambda} - 1 - \frac{2x}{\eta} \leqslant \log \frac{\eta}{4\lambda} - 2. \tag{9.17}$$

Therefore, for small η and reasonable λ, $\tilde{p} < 0$ and $p = 2$ for all x. Hence, no resolution of the boundary layer is formed. Indeed, by (9.5b), for very small η (singular-perturbation case)

$$\left(\frac{h}{\eta}\right)^3 = \frac{2\lambda}{x} e^{2x/\eta} \geqslant \frac{4\lambda e}{\eta} \gg 1 \tag{9.18}$$

so that $h \gg \eta$. In the practical situation where the solution in the external region is not constant, the actual mesh-size will be determined by the external regime.

9.5. *Boundary Corners.* Consider the two-dimensional Poisson equation $\Delta U = F$ with smooth F and homogeneous boundary conditions, near a boundary corner with angle π/α, $\frac{1}{2} \leqslant \alpha < 1$. Denoting by r the distance from the corner, at small r the solution U is $O(r^\alpha)$, and so is also the error weighting function G (if accuracy is sought in the solution, but not in its derivatives near the boundary). Hence, $\tau = O(h^p r^{\alpha-p-2})$ and $\partial\tau/\partial h = O(h^{p-1} r^{\alpha-p-2})$. If we fix the order of approximation p, then the optimal mesh-spacing derived from (8.9a) is

$$h = O(\lambda^{1/(p+2)} r^\beta), \qquad \beta = \frac{p + 2 - 2\alpha}{p + 2}. \tag{9.19}$$

Hence, by (8.4) and (8.1) the total work and total error contribution from a region of radius r around the corner are, respectively,

$$W = \int \frac{p}{h^2}\, dx\, dy = O(\lambda^{-2/(p+2)} r^{2-2\beta}),$$

$$E = \int G\tau\, dx\, dy = O(\lambda^{p/(p+2)} r^{2-2\beta}).$$

Hence the relation $E \approx W^{-p/d}$ (the usual relation in d-dimensional smooth problem with pth-order approximation) still holds uniformly. The corner does not "contaminate the global convergence.

In the practical grid organization (Section 8.3) finer levels G^k with increasingly smaller mesh-sizes $h_k = 2^{-k} h_0$ will be introduced near the corner. By (9.19), the level G^k will extend from the corner to a distance $r_k = C\lambda^{2\alpha-p-2} h_k^{1/\beta}$. Since $\beta < 1$, for small h_k we get $h_k > r_k$. This gives us in practice a natural stopping value for the refinement process: The finest mesh-size near the corner is such that $h_k \approx 4r_k$, so that level G^k still has an inner point belonging to G^{k-1}.

9.6. *Singularities.* Like boundary corners, all kinds of other problem singularities when treated adaptively, cause no degradation of the convergence rate (of E as a function of W).

Consider for example the differential equation $LU = F$ where F is smooth except for a jump discontinuity at $x = 0$. Whatever the approximation order p, the system will find $-\Delta E$ (see (8.11)) to be $O(1)$ at all points whose difference equation include values on both sides of the discontinuity. At these points further refinements will, therefore, be introduced as long as $-\Delta E/\Delta W > O(\lambda)$. Thus, around $x = 0$, some fixed number (depending only on p) of mesh-points will be introduced at each level G^k, until a mesh-size $\bar{h} = O(\lambda^{1/d})$ is reached. The total amount of added work is therefore proportional to the number of levels introduced, which is $O(\log \bar{h})$. The error contribution of the discontinuity is $O(\bar{h}^d)$, which is exponentially small in terms of the added work.

This and similar analyses show that the adaptive scheme retains its high-order convergence even when the problem is only piecewise smooth, or has algebraic singularities, etc.

10. Historical Notes and Acknowledgements. Coarse-grid acceleration techniques were recommended and used by several authors, including Southwell [25], [13], [14], Stiefel [15], Fedorenko [5], Ahamed [19], Wachspress [17], de la Vallée Poussin [16] and Settari and Aziz [24]. Southwell called his technique "block" and more generally "group relaxation", described it as "almost essential to practical success", and gave heuristic explanation as well as practical implementation methods based on variational considerations ("the aim being to reduce the total energy by as great an amount as possible"). He also depicted procedures of "advance to a finer net" [14]. Techniques of multiplicative coarse-grid corrections (special cases of which appeared in [14], [19]) were developed by Wachspress [17, Chapter 9], who called them "variational techniques". This work motivated several studies, by Froelich, Wagner, Nakamura and Reed (see a brief survey in [18]) and was applied in nuclear reactor design computations.

All these were two-level methods. The multi-grid idea was introduced by Fedorenko [6], mainly for theoretical purposes. Namely, he rigorously proved that $W(n, \epsilon)$, the number of operations required to reduce the residuals, of a Poisson problem on a rectangular grid with n points, by a factor ϵ, is $O(n|\log \epsilon|)$. Bakhvalov [1] generalized this result to any second-order elliptic operator with continuous coefficients. For large n, this is the best possible result—except for the actual value of the coefficient. The Fedorenko estimate can be written as

$$W(n, .01) \leqslant 210000n + W(10^6, .01),$$

and the Bakhvalov constants are still much larger. For admissible values of n these estimates are therefore far worse than estimates obtained in other methods, and they did not encourage any development of the method. Fedorenko experimented with a two-level algorithm only, and seemed to imply that for practical grid sizes ADI may be more efficient. He did not realize the true practical potential, in both efficiency and programming simplification, of a full, systematic multi-grid approach. (It can be proved that $W(n, .01) \leqslant 168n$, and in practice $W(n, .01) \approx 50n$ is obtainable. See Appendix C.)

The first full multi-grid algorithms and numerical tests were described in [2]. Our original approach was to regard the finer levels as "correcting" the coarser level (cf. Sections 1, 7.2 and 7.5 above). For uniform nonadaptive grids this approach turns out to be equivalent to the one implied by [6], but fundamentally it is different and more powerful, since the process is not confined to preassigned discrete systems.

A systematic multi-grid approach for a restricted class of problems, with somewhat different procedures of relaxation and transfer to coarser grids, is described in [21]. Another similar method was independently developed in [28]. The multi-grid method is also portrayed in [23] and a new rigorous $O(n)$ convergence theorem for finite-element formulations is given in [29].

Adaptive discretization procedures were introduced by several authors. See for example [10], [20], [22] and references in [22]. The present approach is different, not only in its multi-level setting, but also in its basic criteria and procedures. It has common features with procedures for initial-value problems in ordinary differential equations, described in Chapter 5 of [27].

It is my pleasure to acknowledge the help I received from my students and colleagues throughout the work reported here. Yosef Shiftan, Nathan Diner, Yehoshua Fuchs and Dan Ophir in the Weizmann Institute; Jerry South in NASA Langley Research Center; and Will Miranker, Don Quarles, Fred Gustavson and Allan Goodman at IBM Thomas J. Watson Research Center—thank you all. I am also grateful for valuable discussions I had with Olof Widlund, Eugene Wachspress, Antony Jameson, Perry Newman, Jim Ortega, Ivo Babuška and Werner Rheinboldt.

Appendix A. Interpolations and Stopping Criteria: Analysis and Rules. The multi-grid algorithms described above (Sections 4 and 5) need to be supplemented with some rules of interpolation and some stopping criteria. More specifically, for the interpolation I_k^{k-1}, transferring weighted residuals from a fine grid G^k to the next

coarser grid G^{k-1}, we should prescribe the weights, while for I^k_{k-1}, interpolating corrections from G^{k-1} back to G^k, the method and order of interpolation should be prescribed. Stopping criteria should define convergence at the various levels and detect slow convergence rates. Numerical tests show that the parameters to be used are very robust: Full efficiency of the multi-grid algorithm is obtained for stopping parameters that do not depend on the geometry and the mesh-size, and which may change over a wide range (see, e.g., Appendix B), provided the correct *forms* of the stopping criteria are used, and some basic rules of interpolation are observed. To find the correct forms and rules, and to determine the stopping parameters, we have to analyze the Coarse-Grid Correction (CGC) cycle, which consists of interpolating (I^{k-1}_k) the residuals to the coarser grid G^{k-1}, solving the corresponding residual problem on G^{k-1}, and then interpolating (I^k_{k-1}) that solution back as a correction to the G^k approximation.

We can use a local mode analysis (for the linearized, coefficient-frozen difference equations), similar to the example in Section 3. Such an analysis may be inaccurate for the lowest frequency modes, for which the interaction with the boundary is significant. But these lowest modes are of little significance in our considerations, since they are efficiently approximated on the coarsest grids with little computational work, and since care will be taken (i) to choose interpolation schemes that do not convert small low-frequency errors into large high-frequency errors; and (ii) to stop relaxation sweeps before low-frequency error components become so large that they significantly feed the high frequencies (e.g., by boundary and nonlinear interactions). In fact, we will see that the *dominant components* (i.e., the components that are slowest to converge in the combined process of relaxation and coarse-grid corrections) are the Fourier components $e^{i\theta\cdot x/h}$ for which $|\theta|$ is close to $\hat{\rho}\pi$, where (in a general d-dimensional problem)

$$\text{(A.0)} \qquad \theta = (\theta_1, \theta_2, \ldots, \theta_d), \qquad \theta \cdot x = \sum_{j=1}^{d} \theta_j x_j, \qquad |\theta| = \max_{1 \leqslant j \leqslant d} |\theta_j|,$$
$$h = h_k = \hat{\rho} h_{k-1}.$$

These components feed on each other in the interpolation processes between G^k and G^{k-1}, they are slower to converge by relaxation, and in the CGC cycles they may even diverge.

To simplify the discussion we will assume that the mesh-size ratio has its usual value $\hat{\rho} = ½$, which is the only one to be used in practice (cf. Section 6.2).

A.1. *Coarse-Grid Amplification Factors.* For any given set of difference operators L^k and a multi-grid scheme, a local mode analysis of the complete MG cycle can be made (cf. Appendix C), and the various parameters can be optimized. The essential information can, however, be obtained from a much simpler analysis that treat separately the two main processes, relaxation sweeps and CGC cycles. The smoothing rate $\bar{\mu}$ (see Section 3) is the main quantity describing the relaxation sweeps. The CGC local mode analysis is summarized below (for algebraic details see Section 4.5 of [3]).

In the CGC analysis, together with each basic Fourier component $e^{i\theta\cdot x/h}$ $(0 < |\theta| \leqslant \pi/2)$ we should treat all the G^k components that coincide with it on G^{k-1}, i.e., all components $e^{i\theta'\cdot x/h}$ $(0 < |\theta'| \leqslant \pi)$ such that $\theta'_j \equiv \theta_j \pmod{\pi}$ for $j = 1, 2, \ldots, d$. We call such components θ' *harmonics* of θ. We are especially interested in those

harmonics that are not separated from θ by the relaxation sweeps, e.g., the set

$$\Upsilon_\theta = \{\theta' \equiv \theta \pmod{\pi}\colon \mu(\theta') \geqslant \mu(\theta)^2\}.$$

Denote by $|\Upsilon_\theta|$ the number of members in this set. (Usually $|\Upsilon_\theta| = 2^\alpha$, where α is the number of coordinates j for which $|\theta_j| \approx \pi/2$.) In terms of the θ Fourier component and its harmonics, the CGC cycle has two effects:

(i) Assuming the components not in Υ_θ to be comparatively small when the CGC cycle is entered, the set of components in Υ_θ is transformed in the cycle by a certain matrix, whose spectral radius turns out to be

$$\sigma(\theta) = \begin{cases} \sigma_0(\theta) & \text{if } |\Upsilon_\theta| = 1, \\ \max(1, \sigma_0(\theta)) & \text{if } |\Upsilon_\theta| > 1, \end{cases} \tag{A.1}$$

where

$$\sigma_0(\theta) = \left| 1 - \sum_{\theta' \in \Upsilon_\theta} B_k(\theta')\, R(\theta, \theta')\, B_{k-1}(2\theta)^{-1} \rho(\theta') \right|. \tag{A.2}$$

The functions $\rho(\theta')$, $R(\theta, \theta')$ and $B_l(\theta)$ are the "symbols" of I_k^{k-1}, I_{k-1}^k and L^l, respectively, i.e.,

$$\begin{aligned} I_k^{k-1} e^{i\theta'\cdot x/h} &= \rho(\theta') e^{i\theta\cdot x} \qquad \text{(cf. (A.10) below)}, \\ I_{k-1}^k e^{i\theta\cdot x/h} &= \sum_{\theta' \equiv \theta \pmod{\pi}} R(\theta, \theta') e^{i\theta'\cdot x/h}, \\ L^l e^{i\theta\cdot x/h_l} &= B_l(\theta) e^{i\theta\cdot x/h_l} \qquad (l = k,\ k-1). \end{aligned} \tag{A.3}$$

(If L is a *system* of equations, and the right-hand side of (A.2) is therefore a matrix, then $\sigma_0(\theta)$ is meant to be the spectral radius of that matrix.) For small $|\theta|$ we have $|\Upsilon_\theta| = 1$ and hence

$$\sigma(\theta) = \sigma_0(\theta) = 1 - \rho(0) + O(|\theta|^p + |\theta|^I), \tag{A.4}$$

where p is the approximation order of L_k and L_{k-1} (or the minimum of the two) and I is the order of the I_{k-1}^k interpolation ($I = 2$ for linear interpolation, etc.). The principal CGC amplification factor is

$$\bar{\sigma} = \max_{0 \leqslant |\theta| \leqslant \pi/2} \sigma(\theta) = \max(1, \bar{\sigma}_0), \quad \text{where} \quad \bar{\sigma}_0 = \max_{0 \leqslant |\theta| \leqslant \pi/2} \sigma_0(\theta). \tag{A.5}$$

(ii) The CGC cycles also generate new secondary harmonics $\theta'' \notin \Upsilon_\theta$. The rate of generating these, i.e., the ratio of the new θ'' amplitude to the old amplitude of the combined harmonics, turns out to be

$$\sigma_1(\theta'') = |\, B_k(\theta'')\, R(\theta, \theta'')\, B_{k-1}(2\theta)^{-1} \rho(\theta'')| = O(|\theta|^{I-m}), \tag{A.6}$$

where m is the order of the differential equations.

It follows from (A.4), (A.6), that if $\rho(0) = 1$, as it is always chosen to be (cf. Section A.4), and if $I \geqslant m$, then components with small $|\theta|$ are very efficiently reduced in the multi-grid process, together with their harmonics.

A.2. *The Coarse-to-Fine Interpolation* I_{k-1}^{k}. On the other hand, it follows from (A.6) that if $I < m$ then even a small and smooth residual function may produce large high-frequency residuals, and significant amount of computational work will be required to smooth them out. This effect was clearly shown in numerical experiments [2], [11] Hence we have

The Basic Rule: *The order of interpolation should be no less than the order of the differential equations.* ($I \geqslant m$.) In particular, polynomial interpolation should be made with polynomials of degree $\geqslant m - 1$.

Higher interpolation orders ($I > m$) are desired in the initial stages of solving a problem, when the residuals are (locally) smooth. For instance, in regions where the given problem has smoothness of order q (i.e., $F(x) = \Sigma A_\theta e^{i\theta\cdot x/h}$, $A_\theta = O(|\theta|^{-q}h^q)$), in order to ensure that the high-frequency residuals remain $O(h^q)$, *at the ith interpolation from* u^{M-1} *to* u^M *the order should be*

$$I \geqslant m + \max[q - (i-1)p, 0]. \tag{A.7}$$

(In fact, as long as $q > ip$, this interpolation need not be followed by G^k relaxation sweeps, since the low-frequency amplitudes are still dominant. Relaxation would only feed from these low components to high-frequency ones, causing additional work later. Still better, however, instead of this multi-grid mode without intermediate G^k relaxation, is to make a higher-order correction on G^{k-1}.)

Eventually, however, the smoothness of F (which is the original residual function) is completely lost in subsequent residuals and the convergence of components in the dominant range ($|\theta| \approx \pi/2$) becomes our main concern. For these components, *higher interpolation orders* ($I > m$) *is no more effective than the minimal order* ($I = m$). This again was exhibited in numerical experiments [2], [11], which confirmed that the multi-grid efficiency is not improved (except in the $\lceil q/p \rceil$ first cycles) by using $I > m$.

An efficient method to implement high-order interpolations in case of equations of the form $\Delta^m U = F$ is to base the interpolation on suitably-rotated difference approximations. See [14, p. 53] and [7].

A.3. *The Effective Smoothing Rate.* The smoothing factor $\bar{\mu}$ was defined in (3.8) as the largest convergence factor for all components not represented at the coarser level. More relevant, however, is the largest factor among all components for which the coarse-grid correction is not effective, namely,

$$\tilde{\mu} = \max\{\mu(\theta):\ \pi/2 \leqslant |\theta| \leqslant \pi \quad \text{or} \quad \sigma_0(\theta) \geqslant 1\}, \tag{A.8}$$

which we call the "effective smoothing factor". It is clear, on one hand, that no factor smaller than $\tilde{\mu}$ can be generally obtained as a convergence factor per G^k relaxation sweep, no matter how well and how often the G^{k-1} problem is solved. On the other hand, the factor $\tilde{\mu}$ can actually be attained (or approached) by correctly balancing the number of relaxation sweeps in between CGC cycles (see Section A.6). In most cases (all cases examined by us) one can make $\sigma_0(\theta) \leqslant 1$ for all $|\theta| \leqslant \pi/2$ by proper choice of I_k^{k-1} (see Section A.4), and it is therefore justifiable to use $\bar{\mu}$ as the effective factor when relaxation schemes are studied by themselves.

A.4. *The Fine-to-Coarse Weighting of Residuals* (I_k^{k-1}), *and the Coarse-Grid Operator* L^{k-1}. The transfer of the G^k residuals $r^k = f^k - L^k u^k$ to the coarser grid G^{k-1}, to serve there as the right-hand side f^{k-1} (see Section 4, Step (e)) can be made in many ways. Generally, f^{k-1} is defined as some weighted average of the residuals in neighboring G^k points:

$$f^{k-1}(x) = I_k^{k-1} r^k(x) = \sum \rho_\nu r^k(x + \nu h), \tag{A.9}$$

where $\nu = (\nu_1, \nu_2, \ldots, \nu_d)$, ν_j integers, and the summation is over a small set. In terms of these weights, $\rho(\theta)$ in (A.2) is given by

$$\rho(\theta) = \sum \rho_\nu e^{i\theta\cdot\nu}. \tag{A.10}$$

The coarse-grid operator L^{k-1} can also be chosen in many ways, e.g., as some weighted average of the operator L^k in neighboring points.

How are these choices to be made? The main purpose should be to minimize $\bar{\sigma}$, but without investing too much computational work in the weighting. Usually, it is preferable to adjust ρ_ν and not L^{k-1}, because this provides enough control on $\bar{\sigma}$ (cf. (A.2)) and because complicating L^{k-1} adds many more computations and gets increasingly complicated as one advances to still coarser levels. For the programmer, using the same operators at all levels is an important simplification (cf. Appendix B), especially for nonlinear problems. Moreover, if L^{k-1} is derived from L in the same way as L^k, its coefficients will automatically be the suitable weighted averages of the coefficient of L^k. (In case L^k is given and L is not, and if the coefficients vary rapidly, then the weighting (A.12) below should be used for defining the L^{k-1} coefficients.)

It is clear from (A.4) that we should take $\rho(0) = \Sigma\rho_\nu = 1$. There is no a priori restriction, however, on the signs of the weights ρ_ν. The trivial weighting

$$\rho_0 = 1, \qquad \rho_\nu = 0 \quad \text{for } \nu \neq 0; \qquad \rho(\theta) \equiv 1, \tag{A.11}$$

called *injection*, has an important advantage in saving computations, not only because the weighting itself is saved, but mainly because it requires the computation of r^k only at the G^{k-1} points, while other weighting schemes compute r^k at all G^k points, an additional work comparable to one G^k relaxation sweep.

Nontrivial weighting of residuals is *especially important for difference equations with rapidly varying coefficients* (large variations per mesh-width), and, in particular, for nonlinear equations, especially on coarse grids. In the G^k relaxation sweeps for such equations, the slowly converging low frequencies are coupled with high-frequency modes. In the correction function V^k (cf. Sections 2 or 4) these accompanying high-frequency modes are small (i.e., their amplitudes are small relative to the low-frequency amplitudes), and hence V^k can still be well approximated by a function V^{k-1} on the coarser grid G^{k-1}. In the residual r^k, however, these accompanying high-frequency modes may be large (comparable in amplitude to the low-frequency modes) and their distortion of V^{k-1} will be small only if their contribution to $f^{k-1} = I_k^{k-1} r^k$ is also of high frequency (in G^{k-1}). This is indeed the case, except for residual components $\theta = (\theta_1, \ldots, \theta_d)$ where $|\theta| \approx \pi$ and each $|\theta_j|$ is close either to π or to 0, ($j = 1, \ldots, d$). Such a component has on G^{k-1} the low frequency 2θ (mod 2π), and is therefore

disproportionately magnified in V^{k-1}. The weighting of residuals should eliminate exactly these components. A suitable weighting is

$$\text{(A.12)} \qquad \rho_\nu = 2^{-d-|\nu|} \quad \text{for } |\nu| = \max|\nu_i| \leqslant 1, \qquad \rho_\nu = 0 \quad \text{for } |\nu| > 1,$$

for which $\rho(\theta) = \Pi_1^d(1 + \cos\theta_j)/2$, so that components with $\theta_j \approx \pi$ are sharply reduce

Similar weighting is required whenever high frequencies are coupled to low ones. Such coupling may be introduced by the relaxation scheme itself, even in constant-coefficient problems. For example, a Gauss-Seidel relaxation sweep with red-black ordering, starting from a smooth $O(1)$ error function, will generate high-frequency modes with $O(h^m)$ amplitudes in the error and $O(1)$ amplitutes in the residual.

Examples. For symmetric second-order equations, injection should usually be used. For the *5-point Laplace operator*, for example, if we take I_k^{k-1} to be injection, I_{k-1}^k linear interpolation, and L^{k-1} also a 5-point Laplace operator, we get $\bar{\sigma} = \bar{\sigma}_0 = 1$, the minimal possible value. Any weighting is a pure waste, including the "optimal" weighting

$$\text{(A.12a)} \quad \rho_{00} = ½, \qquad \rho_{01} = \rho_{0-1} = \rho_{10} = \rho_{-10} = \frac{1}{8}, \qquad \rho_{\alpha\beta} = 0 \quad \text{for } |\alpha| + |\beta| > 1,$$

which minimized $\bar{\sigma}_0$, giving $\bar{\sigma}_0 = 1/3$, but does not lower $\bar{\sigma}$. Numerical tests (modifying the program of Appendix B) indeed showed no improvement by weighting.

For higher-order equations, nontrivial weighting offers an important advantage. If, for example, L^k and L^{k-1} are 13-point *biharmonic* operators and I_{k-1}^k is cubic interpolation, then $\bar{\sigma} = 3$ for injection, while $\bar{\sigma} = 1$ for the weighting

$$\text{(A.12b)} \quad \rho_{01} = \rho_{0-1} = \rho_{10} = \rho_{-10} = \frac{1}{4}, \qquad \rho_{\alpha\beta} = 0 \quad \text{for } |\alpha| + |\beta| \neq 1.$$

Numerical experiments were conducted with difference equations of the form (3.2), where the coefficients a and c vary wildly, e.g., a and c on G^M are random; or $a = .1$ and $c = 1$ at all points of G^M, except for points coinciding with G^{M-1}, where $a = c = 1$; etc. When, and only when, the algorithm (with proper line relaxation) used the proper residual weighting ((A.12), with similar weighting for transferring a and c to coarser grids, since in these tests a and c were not defined by a differential equation), it was found that even in the worst cases, convergence factors $\mathring{\mu}$ smaller than 0.7 were always obtained. With the weighting (A.11) or (A.12a) cases of divergence were detected.

A.5. *Finite-Element Procedures.* The main difference between finite-element and finite-difference multi-grid procedures is in the interpolation schemes. In the finite element case, interpolation procedures follow automatically from the variational formulation and the definition of the approximation spaces S^k (corresponding to the levels G^k). Usually, S^k is a subspace of S^{k-1}. The coarse-to-fine interpolation is, therefore, simply the identity operation. Also, if the variational problem in S^k is to minimize $A_k(V^k)$, then, for any given approximation v^k, the correction problem in the coarser space S^{k-1} is, simply, to minimize

$$\text{(A.13)} \qquad A_{k-1}(V^{k-1}) = A_k(v^k + V^{k-1}) \qquad (V^{k-1} \in S^{k-1}).$$

Example. Consider the standard example, where S^k is the space of piecewise linear functions on the triangulation G^k and A_k is a Dirichlet integral whose minimiza-

tion is equivalent to the difference equation $\Delta^k V^k = F^k$, Δ^k being the 5-point Laplacian. Computing A_{k-1} by (A.13), it turns out to be equivalent to the equation $\Delta^{k-1} V^{k-1} = I_k^{k-1}(F^k - \Delta^k v^K)$, where I_k^{k-1} has the weights (cf. Section A.4)

$$\rho_{00} = \frac{1}{4}, \qquad \rho_{01} = \rho_{11} = \rho_{10} = \rho_{0-1} = \rho_{-1-1} = \rho_{-10} = \frac{1}{8}.$$

These weights give the same multi-grid convergence rate as injection (and are therefore redundant).

A.6. *Criteria for Slow Convergence Rates.* (A) Relaxation sweeps, say on G^k, should be discontinued, and a switch should be made to a coarse-grid correction, when the rate of convergence becomes slow; e.g., when

$$\text{(A.14)} \qquad \frac{\text{residual norm}}{\text{residual norm a sweep earlier}} \geqslant \eta \equiv \frac{\bar{\sigma}^3 \hat{\mu} + 3\tilde{\mu}}{\bar{\sigma}^3 + 3}.$$

The norm here is a suitable (e.g., L_2, L_∞ or (A.18) below) discrete measure, usually of the "dynamic" residuals, that is, residuals computed incidentally to the relaxation process. $\hat{\mu}$ is the largest amplification factor for Fourier components on the relaxed region. $\hat{\mu} = 1$ can usually be used, except when relaxing on small grids (in partial relaxation sweeps, for example). $\tilde{\mu}$ and $\bar{\sigma}$ are defined in (A.8) and (A.5), respectively. Usually, one can choose the I_k^{k-1} weighting so that $\bar{\sigma} = 1$, in which case $\tilde{\mu} = \bar{\mu}$. In any case, (A.14) is designed to ensure that, on one hand, the CGC cycle is delayed enough to make its $\bar{\sigma}$ magnification small compared with the intermediate reduction by relaxation sweeps. On the other hand, for θ with $\mu(\theta)$ considerably slower than $\tilde{\mu}$, the CGC cycles are still sufficiently frequent to compensate for the slower μ, since their reduction rate $\sigma(\theta)$ decreases rapidly ((A.4) with $\rho(0) = 1$). The stopping rule (A.14) also prevents low error frequencies from dominating the relaxation, thus avoiding significant feeding from low to high frequencies (through boundary and nonlinear interactions).

If the "stopping factor" η varies over the domain of computations (as a result of variations in L, in case of nonlinear or nonconstant-coefficients problems), the largest η should be chosen for the stopping criterion (A.14). If $\log \eta$ changes too much over the domain (which should not happen when a proper relaxation scheme is used), then (A.14) must be checked separately in subdomains, and partial sweeping (see Section A.9) might be used.

An appropriate value of η may also easily be found by direct trial and error. Such a value is typical to the (locally linearized, coefficient-frozen) problem, is independent of either h, Ω or F, and may therefore be found, once and for all, on a moderately coarse grid. In some nonlinear problems the value may need some adjustment as the computations proceed. Whenever the coarse-grid corrections seem to be ineffective, η should be increased, e.g., to $(1 + 3\eta)/4$. Generally, the overall multi-grid convergence rate is not very sensitive to increasing η: At worst, the rate may become η instead of the theoretically best rate $\max_\Omega \tilde{\mu}^{3/4}$ (cf. Section 6.2).

For the *Poisson equation* with Gauss-Seidel relaxation, for example, we have $\bar{\sigma} = 1$, $\tilde{\mu} = \bar{\mu} = .5$, hence $\eta = .625$. The example in Appendix B shows that the optimal MG convergence rate $\tilde{\mu}^{3/4} \approx .595$ is indeed attained. Experimenting with this program gave similar results for any smaller η (the reason being that the minimal num-

ber of two sweeps at each level is good enough in this problem), while for any $\eta \leqslant .95$ the total amount of computational work was no more than twice the work at $\eta = .62$.

(B) Another way to decide upon discontinuation of relaxation is to directly *measure the smoothness of the residuals.* The switch to coarser grids can be made, for instance, when differences between residuals at neighboring points are small compared with the residuals themselves.

A.7. *Convergence Criteria on Coarser Grids.* In the CGC mode analysis above it was assumed that the problem on the coarser grid G^{k-1} was fully solved and then interpolated as a correction to the G^k approximation. In the actual multi-grid algorithm (Section 4) we solve the G^{k-1} problem iteratively, stopping the iterations when some convergence criterion is met. This criterion should roughly detect the situation at which more improvement (per unit work) is obtained by relaxing on the G^k grid (after interpolating) then by further iterating the G^{k-1} problem (before interpolating). A crude mode analysis (similar to Section 4.6.2 in [3]) shows that such a criterion is

$$\|r^{k-1}\| \leqslant \delta \|r^k\|, \qquad \delta = \frac{\bar{\sigma}(1-\mu_k^{2^{-d}})}{\bar{S}(\mu_k^{2^{-d}} - \mu_{k-1})}, \tag{A.15}$$

where d is the dimension, $\bar{\sigma}$ is given by (A.5),

$$\bar{S} = \max_{|\theta| \leqslant \pi/2} \left| \sum_{\theta \in \Upsilon_\theta} B_k(\theta') R(\theta, \theta') B_{k-1}(2\theta)^{-1} \rho(\theta') \right|$$

and $\mu_l = \tilde{\mu}^{(1-2^{-d})}$ on the G^l grid (cf. (A.8)). $\|r^{k-1}\|$ is any norm, such as L_2, L_∞ or (A.18), of the current residuals in the G^{k-1} problem, while $\|r^k\|$ is the corresponding norm in the G^k problem. It is important that these norms are comparable: They shoul be discrete approximations to the same continuum norm. Also, if r^{k-1} are the "dynamic" residuals (i.e., computed incidentally to the last G^{k-1} relaxation sweep, using at each point the latest available values of the relaxed solution), then r^k should be the G^k dynamic residuals, *unlike* the residuals transferred to G^{k-1} (to define f^{k-1}; cf. Section A.4) which must be "static" residuals (i.e., computed over the grid without changing the solution at the same time). If, however, r^k and r^{k-1} are static and dynamic, respectively, the parameter δ in (A.15) should be multiplied by a certain factor $\bar{\beta}$ (see Section 4.6.2 in [3]).

The stopping criterion (A.15) is based on the assumption that error components with $|\theta| = \pi/2$ dominate the process. In the first $\lceil q/p \rceil$ CGC cycles, however, lower components are dominant, and the main consideration is to converge them. Hence, at that initial stage, the G^{k-1} convergence criteria should be

$$\|r^{k-1}\| \leqslant \|\tau^{k-1}\|, \tag{A.16}$$

where τ^{k-1} are the G^{k-1} truncation errors (cf. Section A.8).

The key factor δ can also be found by trial and error. Like η above, it is essentially independent of h, Ω and F, and may, therefore, be found once and for all by tests on moderately coarse grids. Numerical experiments show that the overall multigrid efficiency is not very sensitive to very large variations in δ and, in particular, δ

may be lowered by orders of magnitudes without large changes in the efficiency. For example:

For the 5-point *Poisson equation* with Gauss-Seidel relaxation, injection and linear interpolations, (A.15) yields $\delta = .219$. Numerical experiment (e.g., with the program in Appendix **B**) shows that with any $.001 \leqslant \delta \leqslant .5$ the computational work is no more than 25% above the work with $\delta = .22$, and no more than 100% extra work for any $.0001 \leqslant \delta \leqslant .7$.

A.8. *Convergence on the Finest Grid.* On the finest grid G^M the solution is usually considered converged when the (static) residuals are of the order of the truncation error, in some appropriate norm. One way to estimate the truncation error is to measure them on coarser grids by (5.7), and extrapolate (taking into account that they are $O(h^p)$). Another, related but more straightforward criterion is to detect when the G^M solution has contributed most of its correction to the G^{M-1} solution. In the FAS algorithm the natural place to check is when a new $\bar{F}^{M-1}$ is computed, the convergence test being

$$\|\bar{F}^{M-1} - \bar{F}^{M-1}_{\text{previous}}\| \ll \|\bar{F}^{M-1} - I_M^{M-1} F^M\|. \tag{A.17}$$

The norm here may be any (L_2, L_∞, etc.), but the most relevant one is the discrete version of the norm (cf. Section 8.1)

$$\|f\| = \int G(x)|f(x)|\, dx. \tag{A.18}$$

A.9. *Partial Relaxation Sweeps.* A partial relaxation sweep over G^k is a relaxation sweep that may skip some subdomains of G^k. (Unlike "selective" relaxation sweeps, which in principle pass through all the grid points, although corrections may not be introduced in some of them. Cf. Section 3.2. A partial sweep may be selective, too.)

Partial sweeps are not used much in standard relaxation calculations. Usually, a slow-to-converge subdomain is coupled to other subdomains and therefore cannot be relaxed separately. In the multi-grid process, however, only high-frequency error components are to be reduced by relaxation, and this can be done separately in subdomains: *With regard to high frequencies, subdomains are practically decoupled.* Hence, in the multi-grid process, partial sweeps are potentially very important. In fact, high-frequency amplitudes may vary greatly over the domain, especially if $\bar{\mu}$ and $\bar{\sigma}$ vary much, or if high-frequency error components are introduced at boundaries, making partial sweeping there very desirable.

Partial sweeping may be performed by applying a criterion for slow convergence (Section A.6) separately in subdomains. A subdomain may be excluded from subsequent relaxation sweeps if slow convergence is shown simultaneously on that subdomain and on all neighboring subdomains. Under-relaxation may be used to phase-out the relaxed region (cf. [3], Section 4.6.4). The subdomains may be chosen quite arbitrarily, but each of them should be large enough (at least 4×4) to allow for separate smoothing.

A.10. *Convergence Criteria on Nonuniform Grids.* When G^k and G^{k-1} are not coextensive (i.e., the domain covered by G^k is only part of the G^{k-1} domain; cf. Section 7.2), the convergence criteria (Sections A.7–A.8) should be slightly modified. First, in (A.15), $\|r^k\|$ is not a comparable norm, since it may be measured on a much smaller subdomain. Instead, one can use the test

$$\|r^{k-1}\| < \delta \|r_1^{k-1}\|/\eta, \tag{A.19}$$

where $\|r_1^{k-1}\|$ is the residual norm computed on G^{k-1} at the first relaxation sweep after switching from G^k. The division by η in (A.19) is designed to compensate for the fact that $\|r_1^{k-1}\|$ is computed a sweep later than $\|r^k\|$.

The other modification is in (A.17), where it was assumed that G^M is the finest level everywhere. Generally, the convergence test can be, for example,

$$\|\overline{F}^k - \overline{F}^k_{\text{previous}}\| \ll \|\overline{F}^k - I^k_{k+1}\overline{F}^{k+1}\| \quad \text{for all } k = (0, 1, \ldots, M-1), \tag{A.20}$$

where the norms are taken over G^k_{k+1} (or, more precisely, over $G^k_{k+1} - G^{k+1}_{k+2}$).

Appendix B: Sample Multi-Grid Program and Output. This simple program of Cycle C (written in 1974 by the author at the Weizmann Institute) illustrates multi-grid programming techniques and exhibits the typical behavior of the solution process. For a full description of Cycle C, see Section 4 or the flowchart in Figure 1.

The program solves a Dirichlet problem for the Poisson equation on a rectangle. The same 5-point operator is used on all grids. The I_k^{k-1} residuals transfer is the trivial one (injection), the I_{k-1}^k interpolation is linear. The higher interpolation (A.7) and the special stopping criterion (A.16), recommended for the first $\lceil q/p \rceil$ cycles, are not implemented here.

For each grid G^k we store both v^k and f^k ($k = 1, 2, \ldots, M$). For handling these arrays f^k is also called v^{k+M}. The coarsest grid has $NX0 \times NY0$ intervals of length $H0$ each. Subsequent grids are defined as straight refinements, with mesh-sizes $H(k) = H0/2{**}(k-1)$. The function $F(x, y)$ is the right-hand side of the Poisson equation. The function $G(x, y)$ serves both as the Dirichlet boundary condition (Φ^M) and as the first approximation (u_0^M). The program cycles until the L_2 norm of the residuals on G^M is reduced below TOL, unless the work WU exceeds WMAX. After each relaxation sweep on any grid G^k, a line is printed out showing the level k, the L_2 norm of the ("dynamic") residuals computed in course of this relaxation, and WU which is the accumulated relaxation work (where a sweep on the finest grid is taken as the work unit).

Note the key role of the GRDFN and KEY subroutines. The first is used to define an $M \times N$ array v^k, i.e., to allocate for it space in the general vector Q (where IQ points to the next available location), and to store its parameters. To use array v^k,

CALL KEY(k, IST, M, N, H)

retrieves the grid parameters (dimension $M \times N$ and mesh-size H) and sets the vector IST(i) so that $v_{ij}^k = Q(\text{IST}(i) + j)$. This makes it easy to write one routine for all arrays v^k; see for example, Subroutine PUTZ(k). Or to write the same routines (RELAX, INTADD, RESCAL) for all levels.

CYCLE C

```
      PROGRAM CYCLE C
      EXTERNAL G,F
      CALL MULTIG(3,2,1.,6,.01,30.,G,F)
      STOP
      END

      FUNCTION F(X,Y)                                   Right-hand side of the equation
      F=SIN(3.*(X+Y))
      RETURN
      END

      FUNCTION G(X,Y)                                   Boundary values and first approximation
      G=COS(2.*(X+Y))
      RETURN
      END


      SUBROUTINE MULTIG(NXO,NYO,HO,M,TOL,WMAX,U1,F)
      EXTERNAL U1,F
      DIMENSION EPS(10)                                 Multi-grid algorithm (see Fig. 1)
      DO 1 K=1,M
      K2=2**(K-1)
      CALL GRDFN(K,NXO*K2+1,NYO*K2+1,HO/K2)
    1 CALL GRDFN(K+M,NXO*K2+1,NYO*K2+1,HO/K2)
      EPS(M)=TOL
      K=M
      WU=0
      CALL PUTF(M,U1,0)
      CALL PUTF(2*M,F,2)
    5 ERR=1.E30
    3 ERRP=ERR
      CALL RELAX(K,K+M,ERR)
      WU=WU+4.**(K-M)
      WRITE(6,4)K,ERR,WU
    4 FORMAT(' LEVEL',I2,'   RESIDUAL NORM=',1PE10.3,'   WORK=',0PF7.3)
      IF(ERR.LT.EPS(K))GOTO 2
      IF (WU.GE.WMAX)RETURN
      IF(K.EQ.1.OR. ERR/ERRP.LT. .6)GOTO 3                          η=.6
      CALL RESCAL(K,K+M,K+M-1)
      EPS(K-1)=.3*ERR                                               δ=.3
      K=K-1
      CALL PUTZ(K)
      GOTO 5
    2 IF (K.EQ.M)RETURN
      CALL INTADD(K,K+1)
      K=K+1
      GOTO 5
      END


      SUBROUTINE GRDFN(N,IMAX,JMAX,HH)                  Define an IMAX × JMAX
      COMMON/GRD/NST(20),IMX(20),JMX(20),H(20)              array v^N .
      DATA IQ/1/
      NST(N)=IQ
      IMX(N)=IMAX
      JMX(N)=JMAX
      H(N)=HH
      IQ=IQ+IMAX*JMAX
      RETURN
      END


      SUBROUTINE KEY(K,IST,IMAX,JMAX,HH)
      COMMON/GRD/NST(20),IMX(20),JMX(20),H(20)          Set IST such that
      DIMENSION IST(1)                                   v^k(I,J)= Q(IST(I) + J),
      IMAX=IMX(K)
      JMAX=JMX(K)                                       and set IMAX = IMX(K)
      IS=NST(K)-JMAX-1                                          JMAX = JMX(K)
      DO 1 I=1,IMAX
      IS=IS + JMAX                                               HH = H(K)
    1 IST(I)=IS
      HH=H(K)
      RETURN
      END
```

```
      SUBROUTINE PUTF(K,F,NH)
      COMMON Q(18000),IST(600)
      CALL KEY (K,IST,II,JJ,H)
      H2=H**NH
      DO 1 I=1,II
      DO 1 J=1,JJ
      X=(I-1)*H
      Y=(J-1)*H
    1 Q(IST(I)+J)=F(X,Y)*H2
      RETURN
      END
```

$v^K \leftarrow H(K)^{NH} \cdot F^K$

```
      SUBROUTINE PUTZ(K)
      COMMON Q(18000),IST(200)
      CALL KEY(K,IST,II,JJ,H)
      DO 1 I=1,II
      DO 1 J=1,JJ
    1 Q(IST(I)+J)=0.
      RETURN
      END
```

$v^K \leftarrow 0$

```
      SUBROUTINE RELAX(K,KRHS,ERR)
      COMMON Q(18000),IST(200),IRHS(200)
      CALL KEY(K,IST,II,JJ,H)
      CALL KEY(KRHS,IRHS,II,JJ,H)
      I1=II-1
      J1=JJ-1
      ERR=0.
      DO 1 I=2,I1
      IR=IRHS(I)
      IO=IST(I)
      IM=IST(I-1)
      IP=IST(I-1)
      DO 1 J=2,J1
      A=Q(IR-J)-Q(IO+J+1)-Q(IO+J-1)-Q(IM+J)-Q(IP+J)
      ERR=ERR+(A+4.*Q(IO+J))**2
    1 Q(IO+J)=-.25*A
      ERR=SQRT(ERR)/H
      RETURN
      END
```

A Gauss-Seidel Relaxation sweep on the equation

$$\Delta_h v^K = v^{KRHS}$$

giving

$$ERR = ||\text{residuals}||_{L_2}$$

```
      SUBROUTINE INTADD(KC,KF)
      COMMON Q(18000),ISTC(200),ISTF(200)
      CALL KEY(KC,ISTC,IIC,JJC,HC)
      CALL KEY(KF,ISTF,IIF,JJF,HF)
      DO 1 IC=2,IIC
      IF=2*IC-1
      JF=1
      IFO=ISTF(IF)
      IFM=ISTF(IF-1)
      ICO=ISTC(IC)
      ICM=ISTC(IC-1)
      DO 1 JC=2,JJC
      JF=JF+2
      A=.5*(Q(ICO+JC)+Q(ICO+JC-1))
      AM=.5*(Q(ICM+JC)+Q(ICM+JC-1))
      Q(IFO+JF) = Q(IFO+JF)+Q(ICO+JC)
      Q(IFM+JF) = Q(IFM+JF)+.5*(Q(ICO+JC)+Q(ICM+JC))
      Q(IFO+JF-1)=Q(IFO+JF-1)+A
    1 Q(IFM+JF-1) = Q(IFM+JF-1)+.5*(A+AM)
      RETURN
      END
```

Linear interpolation and addition

$$v^{KF} \leftarrow v^{KF} + I^{KF}_{KC}\, v^{KC}$$

```
      SUBROUTINE RESCAL(KF,KRF,KRC)
      COMMON Q(18000),IUF(200),IRF(200),IRC(200)
      CALL KEY(KF,IUF,IIF,JJF,HF)
      CALL KEY(KRF,IRF,IIF,JJF,HF)
      CALL KEY(KRC,IRC,IIC,JJC,HC)
      IIC1=IIC-1
      JJC1=JJC-1
      DO 1 IC=2,IIC1
      ICR=IRC(IC)
      IF=2*IC-1
```

Residuals injection

$$v^{KRC} \leftarrow I^{coarse}_{fine}\,(v^{KRF} - \Delta_h v^{KF})$$

```
      JF=1
      IFR=IRF(IF)
      IFO=IUF(IF)
      IFM=IUF(IF-1)
      IFP=IUF(IF+1)
      DO 1 JC=2,JJC1
      JF=JF+2
      S=Q(IFO+JF+1)+Q(IFO+JF-1)+Q(IFM+JF)+Q(IFP+JF)
    1 Q(ICR+JC)=4.*(Q(IFR+JF)-S+4.*Q(IFO+JF))
      RETURN
      END
```

```
LEVEL 6   RESIDUAL NORM= 2.814E+01   WORK=  1.000
LEVEL 6   RESIDUAL NORM= 2.764E+01   WORK=  2.000
LEVEL 5   RESIDUAL NORM= 2.659E+01   WORK=  2.250
LEVEL 5   RESIDUAL NORM= 2.555E+01   WORK=  2.500
LEVEL 4   RESIDUAL NORM= 2.317E+01   WORK=  2.563
LEVEL 4   RESIDUAL NORM= 2.095E+01   WORK=  2.625
LEVEL 3   RESIDUAL NORM= 1.649E+01   WORK=  2.641
LEVEL 3   RESIDUAL NORM= 1.285E+01   WORK=  2.656
LEVEL 2   RESIDUAL NORM= 7.626E+00   WORK=  2.660
LEVEL 2   RESIDUAL NORM= 3.840E+00   WORK=  2.664
LEVEL 3   RESIDUAL NORM= 5.058E+00   WORK=  2.680
LEVEL 4   RESIDUAL NORM= 8.006E+00   WORK=  2.742
LEVEL 4   RESIDUAL NORM= 2.545E+00   WORK=  2.805
LEVEL 5   RESIDUAL NORM= 9.736E+00   WORK=  3.055
LEVEL 5   RESIDUAL NORM= 2.464E+00   WORK=  3.305
LEVEL 6   RESIDUAL NORM= 1.064E+01   WORK=  4.305
LEVEL 6   RESIDUAL NORM= 2.442E+00   WORK=  5.305
LEVEL 6   RESIDUAL NORM= 2.399E+00   WORK=  6.305
LEVEL 5   RESIDUAL NORM= 2.351E+00   WORK=  6.555
LEVEL 5   RESIDUAL NORM= 2.303E+00   WORK=  6.805
LEVEL 4   RESIDUAL NORM= 2.173E+00   WORK=  6.867
LEVEL 4   RESIDUAL NORM= 2.043E+00   WORK=  6.930
LEVEL 3   RESIDUAL NORM= 1.739E+00   WORK=  6.945
LEVEL 3   RESIDUAL NORM= 1.453E+00   WORK=  6.961
LEVEL 2   RESIDUAL NORM= 9.889E-01   WORK=  6.965
LEVEL 2   RESIDUAL NORM= 6.183E-01   WORK=  6.969
LEVEL 1   RESIDUAL NORM= 2.760E-01   WORK=  6.970
LEVEL 1   RESIDUAL NORM= 5.170E-02   WORK=  6.971
LEVEL 2   RESIDUAL NORM= 2.292E-01   WORK=  6.975
LEVEL 3   RESIDUAL NORM= 5.465E-01   WORK=  6.990
LEVEL 4   RESIDUAL NORM= 7.710E-01   WORK=  7.053
LEVEL 4   RESIDUAL NORM= 1.163E-01   WORK=  7.115
LEVEL 5   RESIDUAL NORM= 8.657E-01   WORK=  7.365
LEVEL 5   RESIDUAL NORM= 1.058E-01   WORK=  7.615
LEVEL 6   RESIDUAL NORM= 9.059E-01   WORK=  8.615
LEVEL 6   RESIDUAL NORM= 1.052E-01   WORK=  9.615
LEVEL 6   RESIDUAL NORM= 1.012E-01   WORK= 10.615
LEVEL 5   RESIDUAL NORM= 9.759E-02   WORK= 10.865
LEVEL 5   RESIDUAL NORM= 9.452E-02   WORK= 11.115
LEVEL 4   RESIDUAL NORM= 8.710E-02   WORK= 11.178
LEVEL 4   RESIDUAL NORM= 7.960E-02   WORK= 11.240
LEVEL 3   RESIDUAL NORM= 6.389E-02   WORK= 11.256
LEVEL 3   RESIDUAL NORM= 4.931E-02   WORK= 11.271
LEVEL 2   RESIDUAL NORM= 2.916E-02   WORK= 11.275
LEVEL 2   RESIDUAL NORM= 1.622E-02   WORK= 11.279
LEVEL 2   RESIDUAL NORM= 1.017E-02   WORK= 11.283
LEVEL 3   RESIDUAL NORM= 1.949E-02   WORK= 11.299
LEVEL 4   RESIDUAL NORM= 3.128E-02   WORK= 11.361
LEVEL 4   RESIDUAL NORM= 8.843E-03   WORK= 11.424
LEVEL 5   RESIDUAL NORM= 3.710E-02   WORK= 11.674
LEVEL 5   RESIDUAL NORM= 8.486E-03   WORK= 11.924
LEVEL 6   RESIDUAL NORM= 4.007E-02   WORK= 12.924
LEVEL 6   RESIDUAL NORM= 9.051E-03   WORK= 13.924
```

OUTPUT

Error reduction by a factor greater than 10 per cycle.

Each cycle costs 4.3 WU

Insensitivity: Results would be practically the same for any $.005 \leq \delta \leq .5$ or any $0 \leq \eta \leq .65$

To solve on the same domain problems other than Poisson, the only subroutines to be changed are the relaxation routine RELAX and the residual injection routine RESCAL. The latter is always just a slight variation of the first.

For different domains, more general GRDFN and KEY subroutines should be written. A general GRDFN subroutine, in which the domain characteristic function is one of the parameters, has been developed, together with the corresponding KEY

routine. This essentially *reduces the programming of any multi-grid solution to the programming of a usual relaxation routine.* See [26].

Appendix C. Rigorous Bound to Model-Problem Convergence Rate. We consider the model problem: 5-point Poisson equation $\Delta_h U_h = F$ on a $(n_1 + 1) \times (n_2 + 1)$ rectangular grid G^M with Dirichlet boundary conditions. Let $n_j = 2^M N_j$ and let G^k be the $(2^k N_1 + 1) \times (2^k N_2 + 1)$ uniform grid on the same domain, with mesh-size $h_k = 2^{-k} h_0$ $(k = 0, 1, \ldots, M)$. We will estimate the convergence rate and work per one multi-grid cycle C^M.

The cycle C^M is defined inductively as follows: (i) Make r relaxation sweeps on the G^M approximate solution u^M. To facilitate the rigorous Fourier analysis we choose as our relaxation the Weighted Simultaneous Displacement (WSD, or "weighted Jacobi") method with weights $\omega_{00} = \omega_0$, $\omega_{01} = \omega_{0-1} = \omega_{10} = \omega_{-10} = \omega_1$, say (see Section 3.3). (ii) Inject (cf. Section A.4) the residual problem to G^{M-1}. (iii) Get an approximate solution v^{M-1} to this G^{M-1} problem by *two* C^{M-1} cycles, starting from the zero approximation. (iv) Correct $u^M \leftarrow u^M + I^M_{M-1} v^{M-1}$, where I^M_{M-1} is linear interpolation. (v) Make s more relaxation sweeps on G^M.

It is easily calculated that one WSD sweep amplifies the Fourier component $\exp(i\theta \cdot x/h_M)$ of the residual by the factor

$$\mu(\theta) = 1 - (2 - \cos\theta_1 - \cos\theta_2)(.5\omega_0 + \omega_1 \cos\theta_1 + \omega_1 \cos\theta_2).$$

Denote by $A(\theta)$ the amplitude, before the C^M cycle, of the $\theta = (\theta_1, \theta_2)$ component of the residual. Actually present on the grid G^M are only components of the form $\theta = (\alpha_1 \pi/n_1, \alpha_2 \pi/n_2)$ $(\alpha_j = \pm 1, \pm 2, \ldots, \pm(n_j - 1))$, and their amplitudes $A(\theta_1, \theta_2) = -A(\theta_1, -\theta_2) = -A(-\theta_1, \theta_2)$ are real (assuming two of the boundary lines to lie on the axes). Since $\mu(\theta_1, \theta_2) = \mu(\pm\theta_1, \pm\theta_2)$ is real, the r relaxation sweeps operate separately on each residual mode, transforming its amplitude $A(\theta)$ to $A'(\theta) = \mu(\theta)^r A(\theta)$.

For any component $\theta = (\theta_1, \theta_2)$ such that $|\theta| = \max(|\theta_1|, |\theta_2|) \leqslant \pi/2$, denote $\theta^1 = (\theta_1, \theta_2)$, $\theta^2 = (\theta_1 \pm \pi, \theta_2)$, $\theta^3 = (\theta_1, \theta_2 \pm \pi)$, $\theta^4 = (\theta_1 \pm \pi, \theta_2 \pm \pi)$, where each $\pm$ sign is chosen so that $|\theta^l| < \pi$ $(l = 1, 2, 3, 4)$. Of these four "harmonics", only the θ^1 mode appears on G^{M-1}, its amplitude there (in the right-hand side of the G^{M-1} residual problem formed in Step (ii)) being

$$A_\theta = A'(\theta^1) + A'(\theta^2) + A'(\theta^3) + A'(\theta^4). \tag{C.1}$$

Let ϵ_k denote an upper bound to the factors by which any C^k cycle reduces the L_2 norm of the residuals on G^k. In particular, the two C^{M-1} cycles (Step (iii)) are equivalent to solving a G^{M-1} problem with amplitudes a_θ instead of A_θ, where

$$\sum_{|\theta| \leqslant \pi/2} |a_\theta - A_\theta|^2 \leqslant \epsilon^4_{M-1} \sum_{|\theta| \leqslant \pi/2} A^2_\theta. \tag{C.2}$$

Interpolating the computed correction from G^{M-1} to G^M (Step (iv)), and then relaxing on G^M (Step (v)), the new residual amplitudes are easily calculated to be

$$\begin{aligned} \bar{A}(\theta^l) &= \mu(\theta^l)^s [A'(\theta^l) - S(\theta^l) a_\theta] \\ &= \mu(\theta^l)^s [\mu(\theta^l)^r A(\theta^l) - S(\theta^l) A_\theta + S(\theta^l)(A_\theta - a_\theta)] \quad (l = 1, 2, 3, 4), \end{aligned}$$

where

$$S(\theta) = \frac{(1 + \cos\theta_1)(1 + \cos\theta_2)(2 - \cos\theta_1 - \cos\theta_2)}{2 - \cos 2\theta_1 - \cos 2\theta_2}.$$

Hence

$$\text{(C.3)} \qquad \sum_l \bar{A}(\theta^l)^2 \leqslant 2q^2 \sum_l A(\theta^l)^2 + 2\sum_l S(\theta^l)^2 (A_\theta - a_\theta)^2,$$

where q is an upper bound to the L_2 norms of the 4×4 matrices $Q(\theta)$, defined by

$$Q_{lm}(\theta) = \mu(\theta^l)^s\,(\delta_{lm} - S(\theta^l))\,\mu(\theta^m)^r \qquad (1 \leqslant l,\, m \leqslant 4).$$

Denoting $\beta_j = \sin^2\theta_j$, it is easy to check that

$$\text{(C.4)} \qquad \sum_l S(\theta^l)^2 = 2 - \beta_1\beta_2 \frac{2 + \beta_1 + \beta_2}{(\beta_1 + \beta_2)^2} \leqslant 2.$$

Hence, summing (C.3) over the relevant range of θ, using (C.2) and (C.4) and then (C.1), we obtain

$$\begin{aligned} \sum_{|\theta| \leqslant \pi} \bar{A}(\theta)^2 &\leqslant 2q^2 \sum_{|\theta| \leqslant \pi} A(\theta)^2 + 4\epsilon_{M-1}^4 \sum_{|\theta| \leqslant \pi/2} A_\theta^2 \\ &\leqslant (2q^2 + 4\gamma\epsilon_{M-1}^4) \sum_{|\theta| \leqslant \pi} A(\theta)^2, \end{aligned}$$

where γ is any upper bound to all $\Sigma_l \mu(\theta^l)^{2r}$ $(0 < |\theta| \leqslant \pi/2)$. Thus, we have obtained the bound

$$\text{(C.5)} \qquad \epsilon_M^2 = 2q^2 + 4\gamma\epsilon_{M-1}^4.$$

Choosing $r = s = 1$ and $\omega_0 = 1.07$, $\omega_1 = .155$ (to minimize q^2), straightforward computer calculations show that $q^2 \leqslant .015$ and $\gamma \leqslant 1.18$. From (C.5) it now follows, by induction on M, that $\epsilon_M \leqslant .19$.

The number of operations in the C^M cycle is $W_M \leqslant (12r + 12s + 4)n_1 n_2 + 2W_{M-1}$. Hence, by induction on M, $W_M \leqslant (24r + 24s + 8)n_1 n_2$. We thus have in summary

THEOREM. *The above C^M cycle reduces the L_2 error norm by a factor $\leqslant .19$ and costs 56 operations (additions and multiplications) per grid point.*

The theorem can be improved, in a sense, by defining the C^M cycle to consist of $r + M$ relaxation sweeps and only one C^{M-1} cycle, with sufficiently large r.

(Note, however, that employing arbitrarily large r pays only with simultaneous-displacement schemes on rectangular domains, where eigenfunctions of the coarse-grid correction cycle are also eigenfunctions of the relaxation.)

In practice, .1 reduction is obtained in less than 28 operations-per-point. (See

Appendix B. The Gauss-Seidel sweep employed there can be done in 5 operations-per-point. But for every 3 sweeps on G^k the interpolations I_k^{k-1} and I_{k-1}^k are also performed, respectively costing 6 and 10 operations per 4 grid points. Hence, a work unit in Appendix B should be considered as representing $(3 \times 5 + 6/4 + 10/4)/3 = 19/3$ operations per point.) These operations involve only additions and shifts.

Appendix D. Remarks on Initial-Value Problems. The Multi-Level Adaptive Techniques can be used in time-dependent problems. An obvious use could be made for implicit difference approximations, where a multi-grid algorithm (as in Sections 4 or 5) could be employed to solve, at each time step, the equations for the new time values. Usually, however, alternating-direction implicit (ADI) discretizations, when available, offer the same stability and greater efficiency. The multi-grid approach can be used in cases ADI is difficult to apply, such as nonuniform grids (space mesh-sizes $h^{(1)}, \ldots, h^{(d)}$, and possibly also time step $h^{(0)}$, being nonconstant), curved boundaries, certain nonlinear problems, etc. But note that if a nonuniform grid is constructed as a union of uniform subgrids (cf. Section 7.1), then an ADI procedure may sometimes still be applicable.

Whether the solution is advanced in time explicitly, or by ADI or implicit MG procedure, the automatic adaptive discretization techniques described above (Section 8) can be modified to operate with it, to optimize $h^{(0)}, h^{(1)}, \ldots, h^{(d)}$ and other discretization parameters (the approximation orders, both in space and time, the computational boundary for a problem on an unbounded domain, etc.). The main difference is of course that here we have the special coordinate, time, along which we march, usually without iterating.

Each parameter can be adapted locally including $h^{(0)}$, which may vary both in space and in time, provided we impose a structure as in Section 7, in which $h^{(j)}$ is restricted to the values $h_k^{(j)} = 2^{-k} h_0^{(j)}$. This may create time levels for which the boundary is not the natural boundary. The solution on such "internal" boundaries may be defined by extrapolation from past time levels.

Department of Mathematics
Weizmann Institute of Science
Rehovot, Israel

1. N. S. BAKHVALOV (BAHVALOV), "Convergence of a relaxation method with natural constraints on an elliptic operator," Ž. Vyčisl. Mat. i Mat. Fiz., v. 6, 1966, pp. 861–885. (Russian) MR 35 #6378.

2. A. BRANDT, "Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems," *Proc. 3rd Internat. Conf. on Numerical Methods in Fluid Mechanics* (Paris, 1972), Lecture Notes in Physics, vol. 18, Springer-Verlag, Berlin and New York, 1973, pp. 82–89.

3. A. BRANDT, *Multi-Level Adaptive Techniques*, IBM Research Report RC6026, 1976.

4. A. BRANDT, "Elliptic difference operators and smoothing rates." (In preparation.)

5. R. P. FEDORENKO, "A relaxation method for solving elliptic difference equations," Ž. Vyčisl. Mat. i Mat. Fiz., v. 1, 1961, pp. 922–927. (Russian) MR 25 #766.

6. R. P. FEDORENKO, "On the speed of convergence of an iteration process," Ž. Vyčisl. Mat. i Mat. Fiz., v. 4, 1964, pp. 559–564. (Russian) MR 31 #6386.

7. J. M. HYMAN, "Mesh refinement and local inversion of elliptic partial differential equations," *J. Computational Phys.*, v. 23, 1977, pp. 124–134.

8. A. JAMESON, "Numerical solution of nonlinear partial differential equations of mixed type," *Numerical Solution of Partial Differential Equations,* III (SYNSPADE 1975) (Proc. Sympos., Univ. of Maryland, College Park, Md., 1975), Academic Press, New York.

9. E. M. MURMAN, "Analysis of embedded shock waves calculated by relaxation methods," *Proc. AIAA Conf. on Computational Fluid Dynamics* (Palm Springs, Calif., 1973), AIAA, 1973, pp. 27–40.

10. C. E. PEARSON, "On non-linear ordinary differential equations of boundary layer type," *J. Math. Phys.*, v. 47, 1968, pp. 351–358. MR **38** #5400.

11. Y. SHIFTAN, *Multi-Grid Method for Solving Elliptic Difference Equations*, M. Sc. Thesis, Weizmann Institute of Science, Rehovot, Israel, 1972. (Hebrew)

12. J. C. SOUTH, JR. & A. BRANDT, *Application of a Multi-Level Grid Method to Transonic Flow Calculations*, ICASE Report 76-8, NASA Langley Research Center, Hampton, Virginia, 1976.

13. R. V. SOUTHWELL, *Relaxation Methods in Engineering Science*, Oxford Univ. Press, New York, 1940. MR **3**, 152.

14. R. V. SOUTHWELL, *Relaxation Methods in Theoretical Physics*, Clarendon Press, Oxford, 1946. MR **8**, 355.

15. E. L. STIEFEL, "Über einige Methoden der Relaxationsrechnung," *Z. Angew. Math. Phys.*, v. 3, 1952, pp. 1–33. MR **13**, 874; erratum, **13**, p. 1140.

16. F. de la VALLÉE POUSSIN, "An accelerated relaxation algorithm for iterative solution of elliptic equations," *SIAM J. Numer. Anal.*, v. 5, 1968, pp. 340–351. MR **38** #1845.

17. E. L. WACHSPRESS, *Iterative Solution of Elliptic Systems, and Applications to the Neutron Diffusion Equations of Reactor Physics*, Prentice-Hall, Englewood Cliffs, N. J., 1966. MR **38** #2965.

18. E. L. WACHSPRESS, "Variational acceleration of linear iteration," Proc. Army Workshop Watervliet Arsenal, Albany, New York, 1974.

19. S. V. AHAMED, "Accelerated convergence of numerical solution of linear and non-linear vector field problems," *Comput. J.*, v. 8, 1965, pp. 73–76. MR 31 #5343.

20. I. BABUŠKA, W. RHEINBOLDT & C. MESZTENYI, *Self-Adaptive Refinements in the Finite Element Method,* Technical Report TR-375, Computer Science Department, University of Maryland, 1975.

21. P. O. FREDERICKSON, *Fast Approximate Inversion of Large Sparse Linear Systems*, Math. Report 7-75, Lakehead University, Ontario, Canada, 1975.

22. M. LENTINI & V. PEREYRA, *An Adaptive Finite Difference Solver for Nonlinear Two Point Boundary Problems with Mild Boundary Layers*, Report STAN-CS-75-530, Computer Science Department, Stanford University, Stanford, California, 1975.

23. R. A. NICOLAIDES, "On multiple grid and related techniques for solving discrete elliptic systems," *J. Computational Phys.*, v. 19, 1975, pp. 418–431.

24. A. SETTARI & K. AZIZ, "A generalization of the additive correction methods for the iterative solution of matrix equations," *SIAM J. Numer. Anal.*, v. 10, 1973, pp. 506–521. MR **48** #10148.

25. R. V. SOUTHWELL, "Stress calculation in frameworks by the method of systematic relaxation of constraints. I, II," *Proc. Roy. Soc. London Ser. A*, v. 151, 1935, pp. 56–95.

26. A. BRANDT, "Multi-level adaptive solutions to partial differential equations—ideas and software," *Proc. Sympos. on Math. Software* (Math. Research Center, Univ. of Wisconsin, March 1977), Academic Press. (To appear.)

27. C. WILLIAM GEAR, *Numerical Initial Value Problems in Ordinary Differential Equations,* Prentice-Hall, Englewood Cliffs, N. J., 1971. MR **47** #4447.

28. W. HACKBUSH, *Ein Iteratives Verfahren zur Schnellen Auflösung Elliptischer Randwertprobleme,* Math. Inst., Universität zu Köln, Report 76-12 (November 1976). A short English version: "A fast method for solving Poisson's equation in a general region," *Numerische Behandlung von Differentialgleichungen* (R. Bulirsch, R. D. Grigorieff & J. Schröder, Editors), Lecture Notes in Math., Springer-Verlag, Berlin and New York, 1977.

29. R. A. NICOLAIDES, "On the l^2 convergence of an algorithm for solving finite element equations," *Math. Comp.* (To appear.)

30. R. D. RICHTMYER & K. W. MORTON, *Difference Methods for Initial-Value Problems,* 2nd ed., Interscience, New York, 1967. MR 36 #3515.

An Algorithm for the Machine Calculation of Complex Fourier Series

By James W. Cooley and John W. Tukey

An efficient method for the calculation of the interactions of a 2^m factorial experiment was introduced by Yates and is widely known by his name. The generalization to 3^m was given by Box et al. [1]. Good [2] generalized these methods and gave elegant algorithms for which one class of applications is the calculation of Fourier series. In their full generality, Good's methods are applicable to certain problems in which one must multiply an N-vector by an $N \times N$ matrix which can be factored into m sparse matrices, where m is proportional to $\log N$. This results in a procedure requiring a number of operations proportional to $N \log N$ rather than N^2. These methods are applied here to the calculation of complex Fourier series. They are useful in situations where the number of data points is, or can be chosen to be, a highly composite number. The algorithm is here derived and presented in a rather different form. Attention is given to the choice of N. It is also shown how special advantage can be obtained in the use of a binary computer with $N = 2^m$ and how the entire calculation can be performed within the array of N data storage locations used for the given Fourier coefficients.

Consider the problem of calculating the complex Fourier series

$$X(j) = \sum_{k=0}^{N-1} A(k) \cdot W^{jk}, \qquad j = 0, 1, \cdots, N-1, \tag{1}$$

where the given Fourier coefficients $A(k)$ are complex and W is the principal Nth root of unity,

$$W = e^{2\pi i/N}. \tag{2}$$

A straightforward calculation using (1) would require N^2 operations where "operation" means, as it will throughout this note, a complex multiplication followed by a complex addition.

The algorithm described here iterates on the array of given complex Fourier amplitudes and yields the result in less than $2N \log_2 N$ operations without requiring more data storage than is required for the given array A. To derive the algorithm, suppose N is a composite, i.e., $N = r_1 \cdot r_2$. Then let the indices in (1) be expressed

$$\begin{aligned} j &= j_1 r_1 + j_0, \qquad j_0 = 0, 1, \cdots, r_1 - 1, \qquad j_1 = 0, 1, \cdots, r_2 - 1, \\ k &= k_1 r_2 + k_0, \qquad k_0 = 0, 1, \cdots, r_2 - 1, \qquad k_1 = 0, 1, \cdots, r_1 - 1. \end{aligned} \tag{3}$$

Then, one can write

$$X(j_1, j_0) = \sum_{k_0} \sum_{k_1} A(k_1, k_0) \cdot W^{jk_1 r_2} W^{jk_0}. \tag{4}$$

Received August 17, 1964. Research in part at Princeton University under the sponsorship of the Army Research Office (Durham). The authors wish to thank Richard Garwin for his essential role in communication and encouragement.

Since

$$W^{jk_1r_2} = W^{j_0k_1r_2}, \tag{5}$$

the inner sum, over k_1, depends only on j_0 and k_0 and can be defined as a new array,

$$A_1(j_0, k_0) = \sum_{k_1} A(k_1, k_0) \cdot W^{j_0k_1r_2}. \tag{6}$$

The result can then be written

$$X(j_1, j_0) = \sum_{k_0} A_1(j_0, k_0) \cdot W^{(j_1r_1+j_0)k_0}. \tag{7}$$

There are N elements in the array A_1, each requiring r_1 operations, giving a total of Nr_1 operations to obtain A_1. Similarly, it takes Nr_2 operations to calculate X from A_1. Therefore, this two-step algorithm, given by (6) and (7), requires a total of

$$T = N(r_1 + r_2) \tag{8}$$

operations.

It is easy to see how successive applications of the above procedure, starting with its application to (6), give an m-step algorithm requiring

$$T = N(r_1 + r_2 + \cdots + r_m) \tag{9}$$

operations, where

$$N = r_1 \cdot r_2 \cdots r_m. \tag{10}$$

If $r_j = s_jt_j$ with $s_j, t_j > 1$, then $s_j + t_j < r_j$ unless $s_j = t_j = 2$, when $s_j + t_j = r_j$. In general, then, using as many factors as possible provides a minimum to (9), but factors of 2 can be combined in pairs without loss. If we are able to choose N to be highly composite, we may make very real gains. If all r_j are equal to r, then, from (10) we have

$$m = \log_r N \tag{11}$$

and the total number of operations is

$$T(r) = rN \log_r N. \tag{12}$$

If $N = r^m s^n t^p \cdots$, then we find that

$$\begin{aligned} \frac{T}{N} &= m \cdot r + n \cdot s + p \cdot t + \cdots, \\ \log_2 N &= m \cdot \log_2 r + n \cdot \log_2 s + p \cdot \log_2 t + \cdots, \end{aligned} \tag{13}$$

so that

$$\frac{T}{N \log_2 N}$$

is a weighted mean of the quantities

$$\frac{r}{\log_2 r}, \quad \frac{s}{\log_2 s}, \quad \frac{t}{\log_2 t}, \cdots,$$

whose values run as follows

r	$\frac{r}{\log_2 r}$
2	2.00
3	1.88
4	2.00
5	2.15
6	2.31
7	2.49
8	2.67
9	2.82
10	3.01.

The use of $r_j = 3$ is formally most efficient, but the gain is only about 6% over the use of 2 or 4, which have other advantages. If necessary, the use of r_j up to 10 can increase the number of computations by no more than 50%. Accordingly, we can find "highly composite" values of N within a few percent of any given large number.

Whenever possible, the use of $N = r^m$ with $r = 2$ or 4 offers important advantages for computers with binary arithmetic, both in addressing and in multiplication economy.

The algorithm with $r = 2$ is derived by expressing the indices in the form

$$\begin{aligned} j &= j_{m-1}\cdot 2^{m-1} + \cdots + j_1\cdot 2 + j_0\,, \\ k &= k_{m-1}\cdot 2^{m-1} + \cdots + k_1\cdot 2 + k_0\,, \end{aligned} \tag{14}$$

where j_v and k_v are equal to 0 or 1 and are the contents of the respective bit positions in the binary representation of j and k. All arrays will now be written as functions of the bits of their indices. With this convention (1) is written

$$X(j_{m-1}\,,\ \cdots\,,\ j_0) = \sum_{k_0}\sum_{k_1}\cdots\sum_{k_{m-1}} A(k_{m-1}\,,\ \cdots\,,\ k_0)\cdot W^{jk_{m-1}\cdot 2^{m-1}+\cdots+jk_0}, \tag{15}$$

where the sums are over $k_v = 0, 1$. Since

$$W^{jk_{m-1}\cdot 2^{m-1}} = W^{j_0k_{m-1}\cdot 2^{m-1}}, \tag{16}$$

the innermost sum of (15), over k_{m-1}, depends only on j_0, k_{m-2}, $\cdots$, k_0 and can be written

$$A_1(j_0\,,\ k_{m-2}\,,\ \cdots\,,\ k_0) = \sum_{k_{m-1}} A(k_{m-1}\,,\ \cdots\,,\ k_0)\cdot W^{j_0k_{m-1}\cdot 2^{m-1}}. \tag{17}$$

Proceeding to the next innermost sum, over k_{m-2}, and so on, and using

$$W^{j\cdot k_{m-l}\cdot 2^{m-l}} = W^{(j_{l-1}\cdot 2^{l-1}+\cdots+j_0)k_{m-l}\cdot 2^{m-l}}, \tag{18}$$

one obtains successive arrays,

$$\begin{aligned} &A_l(j_0\,,\ \cdots\,,\ j_{l-1}\,,\ k_{m-l-1}\,,\ \cdots\,,\ k_0) \\ &\quad = \sum_{k_{m-l}} A_{l-1}(j_0\,,\ \cdots\,,\ j_{l-2}\,,\ k_{m-l}\,,\ \cdots\,,\ k_0)\cdot W^{(j_{l-1}\cdot 2^{l-1}+\cdots+j_0)\cdot k_{m-l}\cdot 2^{m-l}} \end{aligned} \tag{19}$$

for $l = 1, 2, \cdots, m$.

Writing out the sum this appears as

$$
\begin{aligned}
A_l(j_0, \cdots, j_{l-1}, k_{m-l-1}, \cdots, k_0) \\
&= A_{l-1}(j_0, \cdots, j_{l-2}, 0, k_{m-l-1}, \cdots, k_0) \\
&\quad + (-1)^{j_{l-1}} i^{j_{l-2}} A_{l-1}(j_0, \cdots, j_{l-2}, 1, k_{m-l-1}, \cdots, k_0) \\
&\quad \cdot W^{(j_{l-3}\cdot 2^{l-3}+\cdots+j_0)\cdot 2^{m-l}}, \qquad j_{l-1} = 0, 1.
\end{aligned} \tag{20}
$$

According to the indexing convention, this is stored in a location whose index is

$$j_0 \cdot 2^{m-1} + \cdots + j_{l-1} \cdot 2^{m-l} + k_{m-l-1} \cdot 2^{m-l-1} + \cdots + k_0 . \tag{21}$$

It can be seen in (20) that only the two storage locations with indices having 0 and 1 in the 2^{m-l} bit position are involved in the computation. Parallel computation is permitted since the operation described by (20) can be carried out with all values of $j_0, \cdots, j_{l-2}$, and $k_0, \cdots, k_{m-l-1}$ simultaneously. In some applications* it is convenient to use (20) to express A_l in terms of A_{l-2}, giving what is equivalent to an algorithm with $r = 4$.

The last array calculated gives the desired Fourier sums,

$$X(j_{m-1}, \cdots, j_0) = A_m(j_0, \cdots, j_{m-1}) \tag{22}$$

in such an order that the index of an X must have its binary bits put in reverse order to yield its index in the array A_m.

In some applications, where Fourier sums are to be evaluated twice, the above procedure could be programmed so that no bit-inversion is necessary. For example, consider the solution of the difference equation,

$$aX(j + 1) + bX(j) + cX(j - 1) = F(j). \tag{23}$$

The present method could be first applied to calculate the Fourier amplitudes of $F(j)$ from the formula

$$B(k) = \frac{1}{N} \sum_j F(j) W^{-jk}. \tag{24}$$

The Fourier amplitudes of the solution are, then,

$$A(k) = \frac{B(k)}{aW^k + b + cW^{-k}}. \tag{25}$$

The $B(k)$ and $A(k)$ arrays are in bit-inverted order, but with an obvious modification of (20), $A(k)$ can be used to yield the solution with correct indexing.

A computer program for the IBM 7094 has been written which calculates three-dimensional Fourier sums by the above method. The computing time taken for computing three-dimensional $2^a \times 2^b \times 2^c$ arrays of data points was as follows:

---

* A multiple-processing circuit using this algorithm was designed by R. E. Miller and S. Winograd of the IBM Watson Research Center. In this case $r = 4$ was found to be most practical.

a	b	c	No. Pts.	Time (minutes)
4	4	3	2^{11}	.02
11	0	0	2^{11}	.02
4	4	4	2^{12}	.04
12	0	0	2^{12}	.07
5	4	4	2^{13}	.10
5	5	3	2^{13}	.12
13	0	0	2^{13}	.13

IBM Watson Research Center
Yorktown Heights, New York

Bell Telephone Laboratories,
Murray Hill, New Jersey

Princeton University
Princeton, New Jersey

1. G. E. P. Box, L. R. Connor, W. R. Cousins, O. L. Davies (Ed.), F. R. Hirnsworth & G. P. Silitto, *The Design and Analysis of Industrial Experiments*, Oliver & Boyd, Edinburgh, 1954.

2. I. J. Good, "The interaction algorithm and practical Fourier series," *J. Roy. Statist. Soc. Ser. B.*, v. 20, 1958, p. 361–372; Addendum, v. 22, 1960, p. 372–375. MR **21** #1674; MR **23** #A4231.

VARIATIONAL METHODS FOR THE SOLUTION OF PROBLEMS OF EQUILIBRIUM AND VIBRATIONS

R. COURANT

As Henri Poincaré once remarked, "solution of a mathematical problem" is a phrase of indefinite meaning. Pure mathematicians sometimes are satisfied with showing that the non-existence of a solution implies a logical contradiction, while engineers might consider a numerical result as the only reasonable goal. Such one sided views seem to reflect human limitations rather than objective values. In itself mathematics is an indivisible organism uniting theoretical contemplation and active application.

This address will deal with a topic in which such a synthesis of theoretical and applied mathematics has become particularly convincing. Since Gauss and W. Thompson, the equivalence between boundary value problems of partial differential equations on the one hand and problems of the calculus of variations on the other hand has been a central point in analysis. At first, the theoretical interest in existence proofs dominated and only much later were practical applications envisaged by two physicists, Lord Rayleigh and Walther Ritz; they independently conceived the idea of utilizing this equivalence for numerical calculation of the solutions, by substituting for the variational problems simpler approximating extremum problems in which but a finite number of parameters need be determined. Rayleigh, in his classical work—*Theory of sound*—and in other pubications, was the first to use such a procedure. But only the spectacular success of Walther Ritz and its tragic circumstances caught the general interest. In two publications of 1908 and 1909 [**39**], Ritz, conscious of his imminent death from consumption, gave a masterly account of the theory, and at the same time applied his method to the calculation of the nodal lines of vibrating plates, a problem of classical physics that previously had not been satisfactorily treated.

Thus methods emerged which could not fail to attract engineers and physicists; after all, the minimum principles of mechanics are more suggestive than the differential equations. Great successes in applications were soon followed by further progress in the understanding of the theoretical background, and such progress in turn must result in advantages for the applications.

An address delivered before the meeting of the Society in Washington, D.C., on May 3, 1941, by invitation of the Program Committee; received by the editors June 16, 1942.

The following address will try to convey an idea of this interplay between practical and theoretical points of view. Completeness cannot be attempted; rather some selected topics with which the speaker has been personally concerned will be discussed.

Usually the solution of a difficult problem in analysis proceeds according to a general scheme: The given problem P with the solution S is replaced by a related problem P_n so simple that its solution S_n can be found with comparative ease. Then by improving the approximation P_n to P we may expect, or we may assume, or we may prove, that S_n tends to the desired solution S of P. The essential point in an individual case is to choose the sequence of P_n in a suitable manner.

Here we shall deal mainly with problems of equilibrium and vibrations (boundary value and eigenvalue problems,[1] respectively). They lead to linear self-adjoint differential equations for an unknown function $u(x, y)$:

$$L(u) = f, \tag{1}$$

or

$$L(u) + \lambda u = 0, \tag{2}$$

in a two-dimensional domain of the x, y-plane, or rather to equivalent variational problems for the kinetic and potential energies of the system.

I. The variational problems

We assume our domain B bounded by a piecewise smooth curve C, and we denote the arc length measured along C by s, and differentiation in the direction of the inward normal by $\partial/\partial n$ or by a subscript n.

1. **Quadratic functionals**. Our variational problems refer to quadratic functionals

$$Q(v) = Q(v, v)$$

defined by symmetric bilinear expressions such as

$$D(v, w) = \iint_B (v_x w_x + v_y w_y)\,dx\,dy, \tag{3}$$

$$M(v, w) = \iint_B [\Delta v \Delta w + \alpha(v_{xx} w_{yy} + v_{yy} w_{xx} - 2v_{xy} w_{xy})]\,dx\,dy, \tag{4}$$

[1] Problems of critical loads (buckling) are likewise mathematically formulated as eigenvalue problems.

which occur in the study of membranes and plates, respectively. In (4) the symbol Δ denotes the Laplacian operator, and α a constant. Furthermore, we shall have to use other quadratic integrals defined by bilinear expressions such as

$$H(v, w) = \iint_B vw\,dx\,dy, \tag{5}$$

$$K(v, w) = \int_C vw\,ds, \tag{6}$$

$$R(v, w) = \int_L v_{xx} w_{xx}\,dx, \tag{7}$$

where L is a line $y = \text{const.}$ in B and C is the boundary of B. We then consider functionals such as

$$Q(v, w) = aD(v, w) + bM(v, w) + cK(v, w) + dR(v, w),$$

where a, b, c, d are constants. Always the "*admissible*" *functions* u, v, $w, \cdots$ are restricted by the condition that all the occurring integrands be at least piecewise continuous.

The stable equilibrium of a plate or membrane under an external pressure f is characterized by a variational problem of the type

$$Q(v) + 2\,H(v, f) = \text{minimum}, \tag{8}$$

for the deflection v, whereas vibrations of plates and membranes correspond to the problem of finding stationary values, $\nu^2 = \lambda$, of

$$Q(v)/H(v). \tag{9}$$

The values ν thus defined are the natural frequencies of the system. $Q(v)$ corresponds to the potential energy of the system in the case of equilibrium, while for vibrations $Q(v)$ and $H(v)$ are the "reduced" potential and kinetic energies, respectively.[2] Terms of the form K and R appearing in the expression Q represent *additional energies concentrated along the boundary* C of B or along a line L in B. For example, in the case of *reinforced plates* we would have $Q = M + dR$ if the reinforcement consists of a bar along the line L.

[2] By "reduced" energies we mean the following: If we assume the free system vibrating with a frequency ω (not necessarily a natural frequency), then the deflection u may be represented in the form $u = v \cos \omega t$, where v is a function of position only. The potential energy V may then be written in the form $Q(v) \cos^2 \omega t$ and the kinetic energy T in the form $\omega^2 H(v) \sin^2 \omega t$; the quantities Q and H are termed the reduced energies.

2. **Rigid and natural boundary conditions.** (See [1, 2, 9].) The Euler differential equations (1) or (2) of our variational problems must be supplemented by appropriate boundary conditions. If we focus our attention on the differential equations as such, it is not at all obvious what boundary conditions belong to a particular problem of mechanics. However, from the point of view of the calculus of variations a complete clarification of this delicate question is almost automatically obtained it the mathematical reasoning follows our mechanical intuition. In the first place we observe that in a variational problem (not so in a problem of differential equations) we need not in advance impose boundary conditions in order to single out a specific solution. If the functions admissible in the competition are not subjected to restrictions at the boundary, we speak of a "*free problem.*" In these problems the first variation of the functional will contain terms referring to the boundary, and the vanishing of the first variation will imply not only Euler's differential equation for the domain B but also conditions on the boundary C, which we call "*natural boundary conditions.*" Now the dominant fact is: *appropriate boundary conditions for differential equations are obtained as natural boundary conditions of corresponding variational problems.* In the latter they may, but they need not be prescribed in advance.

There is only the exceptional case, often termed the "simplest case of a variational problem" of *fixed boundary values* of u or derivatives of u or other expressions in u. Here the situation seems somewhat obscured. (The clamped membrane with the boundary condition $u = 0$ and the clamped plate with the boundary conditions $u = u_x = u_y = 0$ belong in this category.) Such fixed or *rigid* or *artificial boundary conditions* must be explicitly stipulated for the variational problem not only for the differential equation. However, we shall recognize them as limiting cases or degenerations of natural conditions.

Physically, rigid conditions correspond to rigid constraints of the system at the boundary C while natural conditions express equilibrium of the system of C if along C partial or full freedom of motion is permitted.

To understand the significance of natural boundary conditions the following observation is essential: The Euler differential equations depend only on the domain integrals or the energies spread over B. But the *natural* boundary conditions are essentially affected by the boundary integrals representing those contributions to the energies which are concentrated along the boundary C. These terms lead to a great variety of possible natural boundary conditions for the same differential equation. In a somewhat different way we may formulate

the fact: Natural boundary conditions, but not Euler's equations, are affected by divergence expressions in the domain integrals.

2a. **Examples. Plates, plane torsion for multiply-connected domains.** As an example we consider a membrane with a free boundary. A pressure f of average value zero may act on the membrane. In this case we have a variational problem for $D(v) + 2H(v, f)$; if ζ denotes an arbitrary variation of v, the variational condition is

$$D(v, \zeta) + H(\zeta, f) = 0.$$

Transforming $D(v, \zeta)$ by Green's formula into a domain integral plus a boundary integral, we find the natural boundary condition

$$\frac{\partial v}{\partial n} = 0,$$

expressing the fact that no force acts on the boundary C. For a plate, free at the boundary, $Q(v) = M(v)$, and the natural boundary conditions appear as the classical Kirchoff conditions:

$$(1 + \alpha)\Delta v = \alpha(v_{xx}x_n^2 + 2v_{xy}x_n y_n + v_{yy}y_n^2),$$

$$\frac{\partial}{\partial n}\Delta v = \alpha(v_{xx}x_n x_s + v_{xy}[x_n y_s + x_s y_n] + v_{yy}y_n y_s),$$

where x_n, y_n and x_s, y_s are, respectively, direction cosines of the inner normal and tangent vector along C.

In $M(v)$ the term $\alpha(v_{xx}v_{yy} - v_{xy}^2) = \alpha\partial(v_x v_{yy})/\partial x - \alpha\partial(v_x v_{xy})/\partial y$ is a divergence expression, very essential for Kirchoff's natural boundary conditions but irrelevant for Euler's differential equation $\Delta\Delta v = 0$ and without consequence for the clamped plate.

Similar remarks apply to *natural "discontinuity conditions"* that arise if energy is concentrated along lines L interior to B, such as in the case of *reinforced plates*. For example, for a rectangular plate clamped at the boundary but reinforced by a bar along the line $y = 0$, the variational problem becomes

$$Q(v) + 2H(v, f) + kR(v) = \iint_B [(\Delta v)^2 + 2\alpha(v_{xx}v_{yy} - v_{xy}^2)]\,dxdy$$

$$+ k\int_L v_{xx}^2 dx + 2\iint_B vf dxdy = \min.$$

with the condition that $v = v_x = v_y = 0$ on C and v, v_x, v_y are continuous in B while the second derivatives of v are at least piecewise continu-

ous in B. If ζ is an arbitrary variation of v satisfying these conditions, we obtain as above not only the Euler equation $\Delta\Delta v + f = 0$, but also the *natural discontinuity condition*

$$[(\Delta v + \alpha v_{xx})]_{-}^{+} = \frac{\partial^4}{\partial x^4} v(x, 0),$$

where the symbol $[g]_{-}^{+}$ means the amount of discontinuity suffered by a function g in crossing the line $y = 0$ from positive to negative values of y. Another example of a somewhat different character is the problem of *torsion of long columns with multiply connected cross sections*. The contour C of the cross section may include a domain B in the x, y-plane from which are removed holds $B_1, B_2, B_3, \cdots$ with contours $C_1, C_2, C_3, \cdots$ and areas $A_1, A_2, A_3, \cdots$. The multiply-connected domain between C and $C_1, C_2, C_3, \cdots$ may be called B^*.

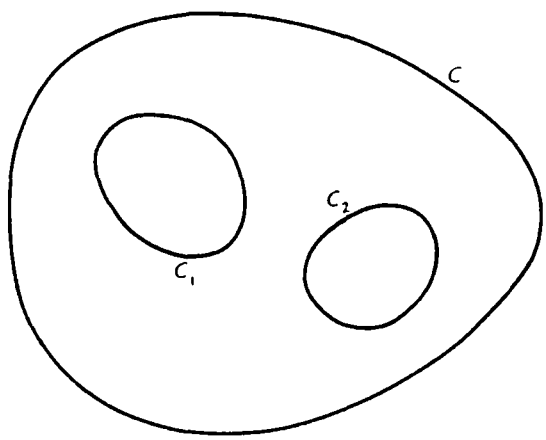


FIG. 1

Then the adequate variational formulation of the torsion problem in proper units is: To find a function $\phi = u$ continuous in $B + C$, having piecewise continuous first derivatives in B, having the boundary values zero on C and constant, but not prescribed values c_i in the holes B_i, such that for the whole domain B

$$D(\phi) = \int\int [(\phi_x^2 + \phi_y^2) + 2\phi]\, dxdy$$

attains its least value d for $\phi = u$. The function u then will give the stresses in the cross section by differentiation.

Note that this problem requires not only to determine u in B^* as a solution of Euler's equation

$$\Delta u = 1$$

with the boundary conditions $u = 0$ on C and $u = \text{const.} = c_i$ on C_i, but also to find the constants c_i. The freedom for these parameters in the functions ϕ admissible in the variational problem must lead to natural conditions from which the c_i are to be characterized.

Indeed, from the variational condition $D(u, \zeta) \times \iint \zeta\, dxdy = 0$ valid for arbitrary ζ that vanishes on C and has constant values in each B: we can only obtain Euler's equation, but likewise immediately the natural boundary conditions

$$\int_{C_i} \frac{\partial u}{\partial n} ds + c_i A_i = 0$$

which connect the unknown boundary constants with the known areas A_i.

Incidentally, for the special choice $\zeta = u$ we obtain (in line with more general results) that the solution, representing a state of equilibrium, satisfies the relation

$$S = D(u) = -\iint u dxdy.$$

The quantity S represents the "*total stiffness*" of the column with respect to torsion.

In the appendix we shall see how this problem, which as an ordinary boundary value problem of a partial differential equation would be rather formidable, can be attacked numerically with success from the point of view developed here.

2b. **Rigid constraints as limiting cases**. If we have additional energy concentrated at the boundary and expressed, for example, by the term $K(v) = \int_C v^2 ds$, then not the Euler equation but the natural boundary conditions will be influenced by these terms. Thus, for the free membrane with

$$Q(v) = D(v) + \gamma K(v)$$

we obtain the natural boundary condition

$$\frac{\partial v}{\partial n} - \gamma v = 0. \tag{10}$$

We observe that as the parameter γ increases indefinitely, that is, as

the elastic restoring forces at the boundary approach a rigid constraint, the condition (10) tends to the boundary condition $v=0$ of the clamped membrane. This is in harmony with the fact that for large positive γ the boundary values of v^2 though free, must in the average be small in order to keep the energy $Q(v)$ in bounds.

Quite generally rigid boundary conditions should be regarded as limiting cases of natural conditions in which a parameter tends to infinity. This corresponds to the physical fact that *rigid constraints are only idealized limiting cases of very large restoring forces.*[3] It may be mentioned that this interpretation of fixed boundary conditions is the key to a more penetrating analysis of the question what we may and what we may not prescribe at the boundary (see [1]).

3. **General conclusions.** The mere formulation of our problem in terms of maxima and minima leads to further important applications. As an example we mention a famous *principle* first formulated by Lord Rayleigh [38]: If a vibrating system whose energies are expressed by quadratic integrals is changed into another system by an increase of masses or by a decrease of elastic forces, then all the natural frequencies of the system can only change toward lower values. This fact was recognized by the speaker (see [9]) as a consequence of the following theorem, which is easily proved: The nth natural frequency of any of our vibration problems is the *highest value of the lowest frequency* of all systems obtained from the given system by imposing $n-1$ constraints; or the nth eigenvalue λ_n is the largest value attained by the minimum $d(w_1, \cdots, w_{n-1})$ of the quotient $Q(v)/H(v)$ if v is subjected to $n-1$ linear conditions of the form

$$H(v, w_i) = 0, \qquad i = 1, \cdots, n-1,$$

the w_i being arbitrarily chosen functions. Here "minimum" refers to a fixed set of $n-1$ functions w_i.

This principle can render useful service for appraising the change in the natural frequencies resulting from changes in the given vibrating system. We recognize immediately that a stiffening of the system by the introduction of new elastic forces (which lead to an increase in potential energy) must produce higher natural frequencies throughout. Likewise, imposing new rigid constraints will have the same effect. Even quantitative results can be obtained by a simple applica-

[3] It might be mentioned that A. Weinstein's method (see §3) is a somewhat different way of presenting rigid boundaries as limiting cases of problems with fewer restrictions.

tion of this principle. The asymptotic behavior of the natural frequencies $\nu_n = \lambda_n^{1/2}$ for large n was found by A. Sommerfeld, H. Weyl, and the speaker. For example, the nth eigenvalue of $\Delta v = \lambda v = 0$ for a free or clamped boundary and domain of area A behaves asymptotically like $4\pi n/A$. (See [9].)

Recently Alexander Weinstein has made an interesting application of this maximum-minimum principle to clamped plates [4]. The eigenvalue problem for a "supported" plate (with $\alpha = 0$) refers to the quadratic energy expressions $Q(v) = \iint_B (\Delta v)^2 dxdy$, $H(v) = \iint_B v^2 dxdy$. It is then required to make the quotient Q/H stationary only under the boundary condition $v = 0$. From this variational problem, we obtain the Euler equation $\Delta\Delta v - \lambda v = 0$ and one natural boundary condition, $\Delta v = 0$; the solutions for this problem are in this case identical with those of the clamped membrane. Now Weinstein, by imposing successively a denumerable number of boundary conditions of the form $\int_C (\partial v/\partial_{ii})\phi_i ds = 0$, $i = 1, 2, 3, \cdots$, where the ϕ_i form a complete system of functions on the boundary C, obtains the problem of the clamped plate as a limiting case. The approximating problems stipulate conditions less restrictive than the limiting problem of a clamped plate, consequently they lead to smaller values of the minima and hence of the maxi-minima, and therefore provide lower bounds for the natural frequencies of the clamped plate. It is remarkable that the approximating problems could be solved explicitly in terms of solutions of the membrane problem [20, 21, 22].

II. Rayleigh-Ritz method

We now discuss the question of attacking a variational problem numerically. In principle, there are many ways in which such a variational problem may suggest approximations by simpler problems. The Rayleigh-Ritz method is only one of them.

1. **The principle and theoretical aspects.** Suppose we seek the minimum d of an integral expression or any other variational expression $I(\phi)$ (for example, our quadratic functionals of the preceding section). We then start with a *minimizing sequence*

$$\phi_1, \phi_2, \phi_3, \cdots, \phi_n, \cdots, \tag{11}$$

that is, a sequence of functions, admissible in our variational problem, for which

$$\lim_{n \to \infty} I(\phi_n) = d, \tag{12}$$

d being the lower bound of the functional $I(\phi)$. The existence of the lower bound d is obvious or may be easily proved in all relevant problems and the *existence* of the minimizing sequence (11) is then a logical consequence.

However, the problem in applications is one, not of the existence, but of the practical *construction* of such a minimizing sequence. Ritz's method is nothing but a recipe for such a construction. A minimizing sequence immediately furnishes an approximation to d (sometimes this is all we wish to know, for example, if we are interested in the natural frequencies of a vibrating system). Moreover, it may be assumed and in many cases it can be proved that the minimizing sequence itself will furnish a good approximation for the function u which actually solves the problem.

Ritz's construction proceeds as follows: We start with an arbitrarily chosen system of "*coordinate functions*"

$$\omega_1, \omega_2, \cdots, \omega_n, \cdots, \tag{13}$$

which should satisfy the two conditions:

(a) Any linear combination

$$\phi_n = c_1\omega_1 + c_2\omega_2 + \cdots + c_n\omega_n \tag{14}$$

of them is admissible in the variational problem.

(b) They should form a complete system of functions in the sense that *any* admissible function ϕ and its relevant derivatives may be approximated with any degree of accuracy by a linear combination of coordinate functions and of their corresponding derivatives, respectively.

If we begin with such a system of coordinate functions, it is clear that for n sufficiently large and for a suitable choice of the coefficients $c_1, c_2, \cdots, c_n$ of (14) we can find admissible functions ϕ_n for which $I(\phi_n)$ differs arbitrarily little from d. In other words, it is possible to find a minimizing sequence $\phi_1, \phi_2, \cdots, \phi_n, \cdots$ as a sequence of linear combinations of the coordinate functions. In order to obtain such a minimizing sequence we choose the c_i in the following manner. We consider any function ϕ_n defined by (14) and substitute it in our variational problem. $I(\phi_n)$ then becomes a function $F(c_1, \cdots, c_n)$ of the n parameters c_i which we may now determine from the ordinary minimum problem of the calculus

$$I(\phi_n) = F(c_1, \cdots, c_n) = \min. \tag{15}$$

In the problems considered here, $I(\phi)$ is a quadratic or bilinear functional, and (15) leads to a system of n linear equations for the

parameters,[4] a system which may be solved by established methods. Thus the minimizing sequence, ϕ_n $(n = 1, 2, \cdots)$ is found. Recalling our general scheme we may identify the construction of the ϕ_n with the problem P_n.

At this point one important remark may be made [1]. While the convergence of $I(\phi_n)$ to d is assured, it is by no means generally true that ϕ_n tends to u, the solution of the original minimum problem, or, even less so, that the derivatives of ϕ_n tend to the corresponding derivatives of u. However, a comparative investigation of different types of variational problems reveals that, generally speaking, the *convergence* of a minimizing sequence $\phi_1, \phi_2, \cdots, \phi_n, \cdots$ and the sequences of the derivatives of the ϕ_n is *improved if the order of the occurring derivatives becomes higher*. On the other hand, there is a tendency toward *worse convergence as the number of independent variables increases*. For example, in the one-dimensional problem of an elastic string $(Q(v) = \int_0^s v'^2 dx)$, the convergence of the ϕ_n to u is assured, the derivatives ϕ_n', however, need not converge to u'. But for the corresponding problem of the bar $(Q(v) = \int_0^s v''^2 dx)$, not only does ϕ_n converge to u but also ϕ_n' to u'. On the other hand, in the membrane problem even the ϕ_n need not converge to u, while in the case of the plate the convergence of ϕ_n to u is assured. The first success attained by Ritz depended largely on his good fortune in attacking the seemingly more difficult problem of the plate rather than that of the membrane.

These facts which are intimately related to more profound questions in the general theory of the variational calculus have suggested the following method of obtaining better convergence in the Rayleigh-Ritz method. Instead of considering the simple variational problem for the corresponding boundary value problem, we modify the former problem without changing the solution of the latter. This is accomplished by adding to the original variational expression terms of higher order which vanish for the actual solution u. For

[4] This suggests the following interpretation and generalization of the procedure whereby reference to a minimum problem need no longer be made: we replace our differential equation $L(u) = 0$ by the condition that $L(u)$ should be orthogonal to n functions of a previously selected complete system of functions. If, then, n tends to infinity, the totality of all these relations will be substituted for the differential equation, and for any fixed n, u may be chosen, for example, as a linear combination (14): $u = \phi_n$. In this general pattern that goes back to Galerkin, there is more freedom left for the choice of the approximations to the solution u. However, in the Rayleigh-Ritz method proper, generally speaking the question of convergence is more easily investigated. (The generalized *view* interpretation is implicitly mentioned in Ritz' papers and was later developed by several authors. See for example [18, 19].)

example, we may formulate the equilibrium problem for a membrane under the external pressure f as follows:

$$I(v) = \iint_B (v_x^2 + v_y^2 + vf)dxdy + \iint_B k(\Delta v - f)^2 dxdy = \min.,$$

where k is an arbitrary positive constant or function. Such additional terms make $I(v)$ more *sensitive* to variations of v without changing the solution. In other words, minimizing sequences attached to such a "sensitized" functional will by force behave better as regards convergence [7].

The practical value of the method of sensitizing the integral by the addition of terms of higher order has not yet been sufficiently explored. Certainly the sensitizing terms will lead to a more complicated system of equations for the c_i. This means that a compromise must be made for a suitable choice of the arbitrary positive function k so that good convergence is assured while the necessary labor is kept within bounds.

2. **Practical viewpoints.** Theoretically the Rayleigh-Ritz method consists merely in the construction of the minimizing sequence. However, the difficulty that challenges the inventive skill of the applied mathematician is to find suitable coordinate functions and to estimate the accuracy of the result. From a practical point of view almost any success depends on the selection of coordinate functions. If these functions are chosen without proper regard for the individuality of the problem the task of computation will become hopeless. A choice should be made so that the system of linear equations for the c_i obtained from (15) will have a preponderance of terms along the diagonal of their matrix, and that the number of terms to be taken into account be kept small. Since only a few of the coordinate functions will enter into the calculation, the theoretical completeness of the ϕ_n is irrelevant. It is important that the initial function be already a fair approximation, and furthermore that the functions ϕ_1, $\phi_2, \ldots$ should be sufficiently different[5] so that increasing the number of terms will lead to an actual improvement of the approximation. Importance should also be attached to the need for making the numerical evaluation of the coefficients of our linear equations practicable.

In many cases the use of polynomials for coordinate functions is most advantageous. Si Luan Wei has shown in his thesis that results

[5] Such a "difference" can be measured. See Courant-Hilbert, *Methoden der mathemateschen Physik*, vol. I, p. 52.

for plates can be obtained by polynomials more readily and more accurately than by the coordinate functions orginally used by Ritz [40].

Recently physicists have chosen coordinate functions according to the following pattern. One starts with a choice of the function ω_1, which is expected to be a fair approximation of the actual solution. Then ω_2 is defined by $\omega_2 = L(\omega_1)$. Finally one chooses $\omega_3 = L(\omega_2) = LL(\omega_1)$, $\omega_4 = L(\omega_3) = LLL(\omega_1)$, and so on. This choice makes the calculation of the matrix elements of our linear equations for the c_i comparatively simple. Since the operator L introduces higher derivatives which may lead to cumbersome complications at boundaries, it is understandable that such a choice of the ω_n seems feasible primarily for infinite domains where no boundary is given and where the natural boundary conditions are equivalent to the finiteness of the variational integrals.

3. **Boundary conditions.** For rigid boundary conditions the approximation by the Rayleigh-Ritz method is comparatively good. Few admissible coordinate functions would in most cases suffice to yield a result near the desired solution. This pleasant feature, however, is offset by the restriction imposed by the rigid boundary conditions which in general precludes the choice of simple coordinate functions.

For free boundaries and natural boundary conditions the choice of coordinate functions is eased considerably since no boundary conditions need be stipulated in advance. As a general rule we might well use, for the ϕ_n, polynomials with undetermined coefficients. For this great advantage, however, we must pay a price, namely, the necessity for using many more terms to secure reasonable accuracy. Therefore it is sometimes preferable if we can satisfy in advance, at least approximately, the natural boundary conditions, by a proper choice of functions. Still, the advantage gained by making the ϕ_n polynomials might be decisive.

In this connection it seems to be of importance that *rigid boundaries can be considered as a limiting case of free boundaries*, as indicated previously in §I, 2. As an example we may consider the following problem for a membrane:

$$D(v) + 2H(v,f) + \gamma K(v) = \iint_B (v_x^2 + v_y^2 + 2vf)\,dx\,dy + \gamma \int_C v^2\,ds = \min.$$

If γ is very large, then the free boundary problem relating to this expression is almost identical to the corresponding problem for a clamped membrane. By letting γ tend to infinity we "*freeze*" the boundary. This suggests an attempt to solve the equilibrium problem of the clamped plate by choosing a large numerical value for γ and then treating the problem as one for a free membrane. However, to obtain reasonable accuracy for the free problem, the larger the quantity γ the more terms will be necessary. Hence the practical application of the method again requires a compromise—γ must be chosen large enough to approximate rigidity but small enough to keep the necessary labor within reasonable bounds. From a theoretical as well as from a practical point of view it would seem worth while to study the preferable choices of these artificial parameters.

4. **Estimates**. A weak point in the Rayleigh-Ritz procedure is that it does not contain a principle for estimating the accuracy of the approximation. This is not the place to give an account of the numerous efforts made to fill this gap at least theoretically. For problems of equilibrium, estimates for the minimum d can be obtained by a method suggested by Castigliano's principle in the theory of elasticity. The mathematical idea is to represent the minimum value d of the given problem as a maximum value of another variational problem (see [**28**]). The margins obtained are quite narrow. But the practical value of this accuracy does not appear great since the value of d in problems of equilibrium is of little interest,[6] whereas we are concerned with the deflection u and its derivatives. In the case of vibrations, where the minimum values, as squares of the frequencies, are of considerable importance, methods for the estimation of accuracy are in general less satisfactory [**29**].

5. **Objections to the Rayleigh-Ritz method**. The vagueness as to the accuracy of the approximation obtained is only one of the objections to the Rayleigh-Ritz method that may be raised. More annoying is that a suitable selection of the coordinate functions is often very difficult and that laborious computations are sometimes necessary. For these reasons, alternative methods must be studied.

III. METHOD OF FINITE DIFFERENCES. GENERAL RANDOM STATISTICAL METHODS

As far as practical experience goes the most important of these

[6] See, however, E. Trefftz, Math. Ann. vol. 108 (1933), p. 595 where the deflection at a point x_0, y_0 is represented as such a minimum d of a modified functional.

methods is that of finite differences. In this well known procedure we replace differential quotients by difference quotients and integral expressions by finite sums defined over a set of net-points in the plane, for example, a quadratic net in the domain B formed by the intersection of the lines $x = \gamma h, y = \mu h (\gamma = 1, 2, \cdots, N; \mu = 1, 2, \cdots, M)$. The simplified problem P_n of difference equations can very often be solved with relative ease. Then if we permit the mesh, $h = 1/n$, of the net to tend to zero with increasing n, not only does P_n tend to P, but the solutions S_n of the difference equations approach the solution S of the original problem exceedingly well. Furthermore, we have the remarkable fact that all relevant difference quotients of first and higher order converge to the corresponding derivatives pertaining to the original problem.

On these grounds the method of finite difference as a general procedure is often preferable to the Rayleigh-Ritz method. The latter might lend itself more readily to the solution of specific problems where suitable analytic expressions are available for coordinate functions. However, in other cases experience points to the superiority of the method of finite differences. One of the underlying reasons is that finite differences are attached directly to the values of the function itself without an interceding medium such as the more or less arbitrary coordinate functions.

If the variational problems contain derivatives not higher than the first order the method of finite difference can be subordinated to the Rayleigh-Ritz method by considering in the competition only functions ϕ which are linear in the meshes of a sub-division of our net into triangles formed by diagonals of the squares of the net. Four such polyhedral functions the integrals become sums expressed by the finite number of values of ϕ in the net-points and the minimum conditions become our difference equations. Such an interpretation suggests a *wide generalization* which provides great flexibility and seems to have considerable practical value. Instead of starting with a quadratic or rectangular net we may consider from the outset any polyhedral surfaces with edges over an *arbitrarily chosen* (preferably triangular) *net*. Our integrals again become finite sums, and the minimum condition will be equations for the values of ϕ in the net-points. While these equations seem less simple than the original difference equations, we gain the enormous advantage of better adaptability to the data of the problem and thus we can often obtain good results with very little numerical calculation. (See appendix.)

This procedure of finite differences was analyzed from a mathematical point of view—and in particular the convergence for the quad-

ratic mesh width $h \to 0$ was proved—first by Philipps in unpublished papers, then by Philipps and Wiener [**14**], and at the same time by Lusternik and by Courant [**5**], and later by Courant, Friedrichs, and Lewy [**8**].

As was already observed by Philipps and Wiener, equations of finite differences can be interpreted by processes of probability—the so-called random walk. This connection has led Courant, R. Lueneburg [**16**], and Petrovsky [**17**] to a more general attack upon our problems, as follows: Instead of simply replacing our integral expressions by finite sums defined over a net, we may replace them in various ways by other simple functionals and thus obtain a greater variety of possible approximate problems P_n. Consider, for example, the nonnegative function $K(x, y; \xi, \eta)$ which is assumed to be symmetric, to be defined and piecewise continuous in the entire plane and to satisfy $\iint_{-\infty}^{+\infty} K(x, y; \xi, \eta)\, d\xi d\eta = 1$. We then focus our attention on integrals of the following form

$$(16) \qquad T(\omega) = \iiiint_{-\infty}^{+\infty} K(x, y; \xi, \eta)\,[\omega(x, y) - \omega(\xi, \eta)]^2\, dx dy d\xi d\eta$$

taken over the entire plane, and investigate the problem P_n,

$$(16a) \qquad T(\omega) = \min.,$$

for which the admissible functions are prescribed in the domain $\bar{B}$, complementary to B, that is, in the portion of the plane outside of B.

For example, if $K = f(r)$, where $r^2 = (x-\xi)^2 + (y-\eta)^2$ and $f(r) \geqq 0$, while $f(r) = 0$ for $r > h$, it is easily verified that, as the parameter h tends to zero, we have $T(\omega) \to D(\omega)$ so that P_n tends to the boundary problem for $\Delta u = 0$. It can be shown generally that the nucleus K may be chosen dependent on a parameter h so that $T(\omega)$ tends to a given quadratic functional which yields as Euler equation any prescribed homogeneous elliptic differential equation.

Now the variational condition for the problem (16a) is no longer a differential equation but a Fredholm integral equation, which is more easily treated:

$$(17) \qquad u(x, y) = \iint_B K(x, y; \xi, \eta) u(\xi, \eta)\, d\xi d\eta + g(x, y),$$

where

$$g(x, y) = \iint_{\bar{B}} K(x, y; \xi, \eta) u(\xi, \eta)\, d\xi d\eta$$

is known because u is prescribed in $\bar{B}$. The solution u_n of (17) for $K = K_n$ converges to the solution u of the problem P. Moreover, and this is the important point for practical applications, it can be proved that the integral equation (17) is solvable by the Neumann method of iteration under very wide conditions for the nucleus K. Thus we may develop a new method for attacking boundary value problems which, however, so far has not been tested in practical applications.

It may be remarked that, generally speaking, the iteration process will converge more slowly if the approximation of P_n to P becomes better. Hence again a compromise for practical purposes is indicated.

In passing, we mention the intimate *connection* of this method *with statistics*. Let us suppose a substance to be distributed with a density $u(x, y)$ over the plane. We now imagine that the substance is redistributed in distinct steps in such a manner that a unit mass concentrated at the point $P(x, y)$ will be spread over the plane with a density $K(x, y; \xi, \eta)$ at the point $Q(\xi, \eta)$. The integral equation (17) then characterizes a state of statistical equilibrium, if the density $u(x, y)$ is prescribed in the complementary domain $\bar{B}$ of B. Of course, if we interpret the integrals as Stieltjes' integrals, we may include in our formulation even problems of finite differences and random walk problems of the classical type. For us here, the main objective is to point out a method that enables us in principle to find approximately solutions of boundary value problems.

IV. Method of gradients

Still another alternative to the Rayleigh-Ritz procedure should be mentioned. This is the method of gradients, which goes back to a paper published by Hadamard in 1907 [**12**]. Highly suggestive as is Hadamard's attempt, difficulties of convergence were encountered. However, recent developments in the theory of conformal mapping and in Plateau's problem throw new light on Hadamard's idea, so that is seems justified now to expect from it not only purely mathematical existence proofs but also a basis for numerical treatment in suitable cases [**13, 10, 11**].

The principle of the method may be understood from the elementary geometric concept of a vector gradient, Let $u = f(x_1, \cdots, x_n)$ be a non-negative function of the n variables x_i, or as we might say of the position vector $X = (x_1, \cdots, x_n)$, and let us seek to determine a vector $X = X_0$ for which u is at least stationary. We then proceed as follows: on the surface $u = f(x)$ we move a point $(x_1, \cdots, x_n, u)$ so that $x_i(t)$ and $u(t)$ become functions of a time-parameter t. Then

the velocity of ascent or descent along the line $X = X(t)$, $u = u(t)$ on the surface is

$$\frac{du}{dt} = \dot{u} = \sum_{i=1}^{n} \dot{x}_i f_{x_i} = \dot{X} \cdot \operatorname{grad} f.$$

We now choose the line along the line along which the motion proceeds so that the descent is as steep as possible (*lines of steepest descent*). This means to make u negative and as large "as possible" in absolute value, for example, by choosing

$$\dot{X} = -\operatorname{grad} f, \tag{18}$$

so that

$$\dot{u} = -(\operatorname{grad} f)^2.$$

Hence the position vector X moves according to the system of ordinary differential equations (18) along the lines of steepest descent with respect to the function f. Under very general assumptions, it is clear that X, starting from an *arbitrary* initial position, will, for $t \to \infty$, approach a position for which $\operatorname{grad} f = 0$, and therefore for which f is stationary and possibly a minimum. However, instead of using the continuous procedure given by the differential equation (18), we may proceed stepwise, correcting a set of approximations x to the solutions of the equations $\operatorname{grad} f = 0$ by corrections proportional to the respective components of $-\operatorname{grad} f$.

This elementary idea can be generalized to variational problems. If we wish to determine a function $u(x, y)$ defined in B and having prescribed boundary values such that u is the solution of a variational problem

$$I(v) = \iint_B F(x, y, v, v_x, v_y)\,dx\,dy = \min., \tag{19}$$

then we interpret the desired function u as the limit for $t \to \infty$ of a function $v(x, y, t)$, whose values may be chosen arbitrarily for $t = 0$ and for all t thereafter are determined in such a way that the expression $I(v)$, considered as a function $I(t)$ of t, decreases as rapidly as possible toward its minimal value. Of course the boundary values of $v(x, y, t)$ *are the same as those for* $u(x, y)$, so that v_t must vanish at the boundary. If we choose $v = v(x, y, t)$, we find

$$\dot{I}(t) = -\iint_B v_t L(v)\,dx\,dy, \tag{20}$$

where $L(v)$ is the Euler expression corresponding to (20). To consider a concrete example, we suppose that

$$I(v) = \int\int_B (v_x^2 + v_y^2)dxdy, \tag{21}$$

so that our minimum problem amounts to determining the equilibrium of a membrane with given boundary deflections $g(s)$. Then $L(v) = -2\Delta v$. Incidentally (20) displays an analogy between the Euler expression and the gradient of a function $f(x_1, \cdots, x_n)$ of n independent variables. The variation or "velocity" of $I(v)$ is expressed as an "inner product" of the velocity of the "independent function" v with the Euler expression $L(v)$, the *gradient of a functional in function space.*

We now assure ourselves of a steady descent or decrease of $I(t)$ by choosing v_t in accordance with the differential equation

$$v_t = -kL(v), \tag{22}$$

where k is a positive arbitrary function of x, y. (21) then becomes

$$I(t) = -\int\int_B k[L(v)]^2 dxdy,$$

and again we can infer that, for $t \to \infty$, $v(x, y, t)$ will tend to the solution $u(x, y)$ of the corresponding boundary value problem $L(u) = 0$.

For the case of the membrane the differential equation (22) becomes

$$v_t = \Delta v, \tag{23}$$

the equation of *heat transfer*. In our interpretation this equation describes a rapid approach to a stationary state along the "lines of steepest descent." While for the equations (23) or (22) the convergence of v for $t \to \infty$ can be proved, serious difficulty arises if we want to replace our continuous process by a process of stepwise corrections as would be required for numerical applications. Each step means a correction proportional to Δv, thus introducing higher and higher derivatives of the initial function v. Another great difficulty is presented by rigid boundary values.[7]

Yet there do exist classes of problems where such difficulties can be overcome if the method is extended properly. First of all we may

[7] Incidentally, if we apply this procedure to a problem for a finite net, it converges very well and is, as a matter of fact, nothing but a natural method of solving a system of linear equations by a method of iteration.

observe that it is not necessary to select the *steepest* descent along the gradient; it suffices to secure a *safe* descent at a suitably fast rate. Furthermore, if we consider problems for which the boundary value problem of the differential equation presents no difficulties for the domain B, but for which a degree of freedom in the boundary values is left, then the problem reduces to one for finding these boundary values, and now all our difficulties disappear. A typical example is the problem of the conformal mapping of a circle B onto a simply-connected domain G, and it may be that the method of gradients opens a path for the attack of the problem of conformal mapping for multiply-connected domains and other problems as well.

This address has emphasized more theoretical aspects. However, some of the general principles described may be helpful in enlarging our equipment for practical purposes.

APPENDIX[8]

NUMERICAL TREATMENT OF THE PLANE TORSION PROBLEM FOR MULTIPLY-CONNECTED DOMAINS

The computation of the stiffness S defined in §I, 2a furnishes an example of independent interest which permits to compare the practical merits of some of the methods described in this address. Numerical calculations were carried out for the cross sections of the following diagrams, a square from which a smaller square is cut out; and a square, from which four squares are cut out. In the first case our quadratic frame was supposed to be bounded by the lines $x = \pm 1$, $y = \pm 1$ and $x = \pm 3/4$, $y = \pm 3/4$. To apply the Rayleigh-Ritz method for the domain as a whole would already be cumbersome because of the boundary conditions for admissible functions ϕ. However, this difficulty disappears if we exploit the symmetry of the domain and the resulting symmetry of the solution; thus we may confine ourselves to considering only one-eighth of the domain B^*, namely the quadrangle $ABCD$. For this polygon any function of the type

$$\phi = a(1 - x)[1 + (x - 3/4)P]$$

where $P(x, y)$ is a polynomial, is admissible, and its substitution in the integral leads to simple linear equations for the cofficients. Thus for the simplest attempt

$$\phi = a(1 - x)$$

which leaves only one constant a to be determined, we find with a

[8] Addition not contained in the original address.

negligible amount of numerical labor $S = .339$ and $c = -.11$. A refined attempt with the function

$$\phi = a(1-x)[1+\alpha(x-3/4)y]$$

yielded $S = .340$ and $c = -.109$ with little more labor.

These results were checked with those obtained by our generalized method of finite differences were arbitrary triangular nets are permitted. The diagrams are self-explanatory. Unknown are the

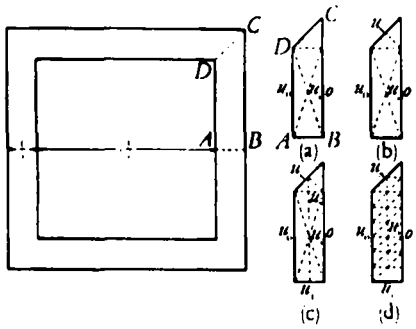


FIG. 2

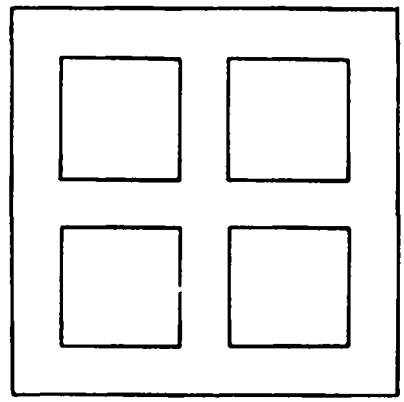


FIG 3

net-point-values u_i, $(c = u_0)$. In the net-triangles our functions were chosen as linear, so that the variational problem results in linear equations for the u_i. The results, easily obtainable, were: case (a) with two unknowns: $S = .344$, $u_0 = -.11$; case (b) with three unknowns: $S = .352$, $u_0 = -.11$; case (c) with five unknowns $S = .353$, $u_0 = -.11$; case (d) with nine unknowns, corresponding to the ordinary difference method $S = .353$, $u_0 = -.11$.

These results show in themselves and by comparison that the generalized method of triangular nets seems to have advantages. It was applied with similar success to the case of a square with four holes, and it is obviously adaptable to any type of domain, much more so than the Rayleigh-Ritz procedure in which the construction of admissible functions would usually offer decisive obstacles.

In a separate publication it will be shown how the method can be extended also to problems of plates and to other problems involving higher derivatives.

Of course, one must not expect good local results from a method

using so few elements. However, it might be expected that a smooth interpolation of the net functions obtained will yield functions which themselves with their derivatives are fairly good approximations to the actual quantities.

REFERENCES

1. R. Courant, *Ueber direkte Methoden bei Variations- und Randwertproblemen*, Jber. Deutschen Math. Verein. vol. 34 (1925) p. 90 ff.; reprinted in Math. Ann. vol. 97 (1927) p. 711 ff.

2. ———, *Ueber direkte Methoden* · · · , Proceedings of the International Congress for Applied Mechanics, Delft, 1925.

3. ———, *Ueber partielle Differenzengleichungen*, Atti del Congresso Internazionale dei Matematici, Bologna, 1928.

4. ———, *Ueber Randwertaufgaben bei partiellen Differenzengleichungen*, Zeitschrift für angewandte Mathematik und Mechanik vol. 6 (1926) p. 1 ff.

5. ———, *Ueber die Theorie der linearen partiellen Differenzengleichungen*, Goettingen Nachrichten, 1925.

6. ———, *Bemerkungen zur numerischen Aufloesung von Randwertproblemen* · · · , Goettingen Nachrichten, 1926, p. 111 ff.

7. ———, *Ueber ein convergenzerzeugendes Prinzip der Variationsrechnung*, Goettingen Nachrichten, 1922.

8. R. Courant, K. Friedrichs, H. Lewy, *Ueber die partiellen Differenzengleichungen der mathematischen Physik*, Math. Ann. vol. 100 (1928) p. 32 ff.

9. R. Courant and D. Hilbert, *Methoden der mathematischen Physik*, vol. I, 2d edition, 1931; vol. II, 1937.

10. R. Courant, *On a method for the solution of boundary-value problems*, Theodore von Kármán Anniversary Volume, 1941.

11. ———, *On the first variation of the Dirchlet-Douglas integral and on the method of gradients*, Proc. Nat. Acad. Sci. U.S.A. vol. 27 (1941) p. 242 ff.

12. J. Hadamard, *Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées*, Mémoires presentés par divers savants éstrangers à l'Académie des Sciences de l'Institut de France (2) vol. 33, 4 (1908).

13. H. Lewy, *Ueber einen Ansatz zur numerischen Loesung von Randwertproblemen*, Goettingen Nachrichten, 1925, p. 118 ff.

14. Philipps and N. Wiener, *Nets and the Dirichlet problem*, Publication of the Massachusetts Institute of Technology, 1925.

15. R. G. D. Richardson, *A new method in boundary value problems for differential equations*, Trans. Amer. Math. Soc. vol. 18 (1917) p. 489 ff.

16. R. Lueneburg, *Das Problem der Irrfahrt ohne Richtungsbeschraenkung* · · · , Math. Ann. vol. 104 (1931) p. 700 ff.

17. Petrovsky, *Ueber das Irrfahrtenproblem*, Math. Ann. vol. 109 (1934) p. 425 ff.

18. W. Feller, *Zur Theorie der Stochastischen Prozesse*, Math. Ann. vol. 113 (1936) p. 113 ff.

19. C. J. Thorne and J. V. Atanasoff, *A functional method for the solution of thin plate problems* · · · , Iowa State College Journal of Science vol. 14 (1940) p. 333 ff.

20. A. Weinstein, *Étude des spectres des équations aux dérivées partielles de la théorié des plaques élastiques*, Mémorial des Sciences Mathématiques, no. 88, 1937.

21. ———, *On a minimal problem in the theory of elasticity*, J. London Math. Soc. vol. 10 (1935) p. 184 ff.

22. ———, *Les vibrations et le calcul des variations*, Portugaliae Mathematica, 1941, p. 36.

23. E. Trefftz, *Ueber Fehlerabschätzungen bei Berechnung von Eigenwerten*, Math. Ann. vol. 108 (1933) p. 595 ff.

24. ———, *Konvergenz und Fehlerabschaetzung beim Ritz'schen Verfahren*, Math. Ann. vol. 100 (1928) p. 503 ff.

25. ———, *Ein Gegenstueck zum Ritz'schen Verfahren*, Verhandlungen, Congress für tecnische Mechanik, Zuerich, 1927, p. 131.

26. H. Cross, *Analysis of continuous frames by distributing fixed-end moments*, Transactions of the American Society of Civil Engineers vol. 96 (1932) pp. 1–10.

27. R. V. Southwell, *Relaxation methods in engineering science. A treatise on approximate computations*, Oxford, 1940.

28. K. Friedrichs, *Ein Verfahren der Variationsrechnung* . . . , Goettingen Nachrichten, 1929, p. 13 ff.

29. K. Friedrichs and G. Horvay, *The finite Stieltjes momentum problem*, Proc. Nat. Acad. Sci. U.S.A. vol. 25 (1939) pp. 528–534.

30. L. Collatz, *Das Differenzenverfahren* . . . , Schriften des Mathematisches Seminars der Universitaet Berlin, vol. 3 (1935).

31. ———, *Ueber das Differenzenverfahren* . . . , Zeitschrift für angewandte Mathematik und Mechanik vol. 16 (1936) p. 239 ff.

32. ———, *Bemerkungen zur Fehlerabschaetzung für das Differenzenverfahren bei partiellen Differentialgleichungen*, Zeitschrift für angewandte Mathematik und Mechanik vol. 13 (1933) p. 56 ff.

33. H. P. Geiringer, *Zur Praxis der Loesung linearer Gleichungen in der Statik*, Zeitschrift für angewandte Mathematik und Mechanik vol. 8 (1928) p. 446 ff.

34. R. von Mises and H. P. Geiringer, *Praktische Verfahren der Gleichungsaufloesung*, Zeitschrift für angewandte Mathematik und Mechanik vol. 9 (1929) pp. 58–77; 152–164.

35. J. P. Den Hartog, *Mechanical vibrations*, 1st edition, 1934, New York.

36. S. Gerschgorin, *Fehlerabschaetzung fuer das Differenzenverfahren zur Loesung partieller Differentialgleichungen*, Zeitschrift für angewandte Mathematik und Mechanik vol. 10 (1930) pp. 373–382.

37. S. Timoshenko, *Vibration problems in engineering*, 2d edition, New York, 1938.

38. Lord Rayleigh, *Theory of sound*, 2d edition, 1894 and 1896, London.

39. W. Ritz, *Ueber eine neue Methode zur Lesung gewisser Variationsprobleme der mathematischen Physik*, J. Reine Angew. Math. vol. 135 (1908); *Theorie der Transversalschwingungen einer quadratischen Platte mit freien Raendern*, Annalen der Physik vol. 38 (1909).

40. Si Luan Wei, *Ueber die eingespannte rechteckige Platte* . . . , dissertation, Goettingen, 1925.

NEW YORK UNIVERSITY

R. Courant*
K. Friedrichs*
H. Lewy†

On the Partial Difference Equations of Mathematical Physics

Editor's note: This paper, which originally appeared in *Mathematische Annalen* 100, 32–74 (1928), is republished by permission of the authors. We are also grateful to the Atomic Energy Commission for permission to republish this translation, which had appeared as AEC Report NYO-7689, and to Phyllis Fox, the translator, who did the work at the AEC Computing Facility at New York University under AEC Contract No. AT(30-1)-1480. Professor Eugene Isaacson had made suggestions on this translation.

Introduction

Problems involving the classical linear partial differential equations of mathematical physics can be reduced to algebraic ones of a very much simpler structure by replacing the differentials by difference quotients on some (say rectilinear) mesh. This paper will undertake an elementary discussion of these algebraic problems, in particular of the behavior of the solution as the mesh width tends to zero. For present purposes we limit ourselves mainly to simple but typical cases, and treat them in such a way that the applicability of the method to more general difference equations and to those with arbitrarily many independent variables is made clear.

Corresponding to the correctly posed problems for partial differential equations we will treat boundary value and eigenvalue problems for elliptic difference equations, and initial value problems for the hyperbolic or parabolic cases. We will show by typical examples that the passage to the limit is indeed possible, i.e., that the solution of the difference equation converges to the solution of the corresponding differential equation; in fact we will find that for elliptic equations in general a difference quotient of arbitrarily high order tends to the corresponding derivative. Nowhere do we assume the existence of the solution to the differential equation problem—on the contrary, we obtain a simple existence proof by using the limiting process.[1] For the case of elliptic equations convergence is obtained independently of the choice of mesh, but we will find that for the case of the initial value problem for hyperbolic equations, convergence is obtained only if the ratio of the mesh widths in different directions satisfies certain inequalities which in turn depend on the position of the characteristics relative to the mesh.

We take as a typical case the boundary value problem of potential theory. Its solution and its relation to the solution of the corresponding difference equation has been extensively treated during the past few years.[2] However in contrast to the present paper, the previous work has involved the use of quite special characteristics of the potential equation so that the applicability of the method used there to other problems has not been immediately evident.

In addition to the main part of the paper, we append an elementary algebraic discussion of the connection of the boundary value problem of elliptic equations with the random walk problem arising in statistics.

[1] Our method of proof may be extended without difficulty to cover boundary value and eigenvalue problems for arbitrary linear elliptic differential equations and initial value problems for arbitrary linear hyperbolic differential equations.

* Now at Courant Institute of Mathematical Sciences, New York University.
† Now at University of California, Berkeley.

[2] J. le Roux, "Sur le problem de Dirichlet", *Journ. de mathém. pur. et appl.* (6) **10**, 189 (1914). R. G. D. Richardson, "A new method in boundary problems for differential equations", *Trans. of the Am. Math. Soc.* **18**, p. 489 ff, (1917). H. B. Philips and N. Wiener, *Nets and the Dirichlet Problem*, Publ. of M.I.T. (1925). Unfortunately these papers were not known by the first of the three authors when he prepared his note "On the theory of partial difference equations," *Gött. Nachr.* 23, X, 1925, on which the present work depends. See also L. Lusternik, "On an application of the direct method in variation calculus," Recueil de la Société Mathém. de Moscou, 1926. G. Bouligand, "Sur le probléme de Dirichlet," *Ann. de la soc. polon. de mathém.* **4**, Cracow (1926). On the meaning of the difference expressions and on further applications of them, see R. Courant, "Über direkte Methoden in der Variationsrechnung," *Math. Ann.* **97**, p. 711, and the references given therein.

I. The elliptic case

Section 1. Preliminary remarks

• 1. *Definitions*

Consider a rectangular array of points in the (x, y)-plane, such that for mesh width $h > 0$ the points of the lattice are given by

$$\left.\begin{aligned} x &= nh \\ y &= mh \end{aligned}\right\} \qquad m, n = 0, \pm 1, \pm 2, \cdots .$$

Let G be a region of the plane bounded by a continuous closed curve which has no double points. Then the corresponding mesh region, G_h—which is uniquely determined for sufficiently small mesh width—consists of all those mesh points lying in G which can be connected to any other given point in G by a connected chain of mesh points. By a connected chain of mesh points we mean a sequence of points such that each point follows in the sequence one of its four neighboring points. We denote as a *boundary point* of G_h a point whose four neighboring points do not all belong to G_h. All other points of G_h we call *interior points*.

We shall consider functions $u, v, \cdots$ of position on the grid, i.e., functions which are defined only for grid points, but we shall denote them as $u(x, y), v(x, y), \cdots$. For their forward and backward difference quotients we employ the following notation,

$$u_x = \frac{1}{h}[u(x + h, y) - u(x, y)],$$

$$u_y = \frac{1}{h}[u(x, y + h) - u(x, y)].$$

$$u_{\bar{x}} = \frac{1}{h}[u(x, y) - u(x - h, y)],$$

$$u_{\bar{y}} = \frac{1}{h}[u(x, y) - u(x, y - h)].$$

In the same way the difference quotients of higher order are formed, e.g.,

$$(u_x)_{\bar{x}} = u_{x\bar{x}} = u_{\bar{x}x}$$
$$= \frac{1}{h^2}[u(x + h, y) - 2u(x, y) + u(x - h, y)], \text{ etc.}$$

• 2. *Difference expressions and Green's function*

In order to study linear difference expressions of second order, we form (using as a model the theory of partial differential equations), a bilinear expression from the forward difference quotients of two functions, u and v,

$$B(u, v) = au_xv_x + bu_xv_y + cu_yv_x + du_yv_y + \alpha u_xv + \beta u_yv + \gamma uv_x + \delta uv_y + guv,$$

where $a = a(x, y), \cdots, \alpha = \alpha(x, y), \cdots, g = g(x, y)$ are functions on the mesh.

From the bilinear expression of first order we derive a difference expression of second order in the following way: we form the sum

$$h^2 \sum_{G_h}\sum B(u, v)$$

over all points of a region G_h in the mesh, where in $B(u, v)$ the difference quotients between boundary points and points not belonging to the mesh are to be set equal to zero. We now transform the sum through partial summation, i.e., we arrange the sum according to v, and split it up into a sum over the set of interior points, G_h' and a sum over the set of boundary points, Γ_h. Thus we obtain:

$$h^2 \sum_{G_h}\sum B(u, v) = -h^2 \sum_{G_h'}\sum vL(u) - h \sum_{\Gamma_h} v\mathfrak{R}(u). \qquad (1)$$

$L(u)$ is a linear difference expression of second order defined for all interior points of G_h:

$$L(u) = (au_x)_{\bar{x}} + (bu_x)_{\bar{y}} + (cu_y)_{\bar{x}} + (du_y)_{\bar{y}} - \alpha u_x - \beta u_y + (\gamma u)_{\bar{x}} + (\delta u)_{\bar{y}} - gu.$$

$\mathfrak{R}(u)$ is, for every boundary point, a linear difference expression whose exact form will not be given here.

If we arrange $\sum_{G_h}\sum B(u, v)$ according to u, we find

$$h^2 \sum_{G_h}\sum B(u, v) = -h^2 \sum_{G_h'}\sum uM(v) - h \sum_{\Gamma_h'} u\mathfrak{S}(v). \qquad (2)$$

$M(v)$ is called the adjoint difference expression of $L(u)$ and has the form

$$M(v) = (av_x)_{\bar{x}} + (bv_y)_{\bar{x}} + (cv_x)_{\bar{y}} + (dv_y)_{\bar{y}} + (\alpha v)_{\bar{x}} + (\beta v)_{\bar{y}} - \gamma v_x - \delta v_y - gv,$$

while $\mathfrak{S}(v)$ is a difference expression corresponding to $\mathfrak{R}(u)$ for the boundary.

Formulas (1) and (2) give

$$h \sum_{G_h'}\sum [vL(u) - uM(v)] + h \sum_{\Gamma_h'} [v\mathfrak{R}(u) - u\mathfrak{S}(v)] = 0. \qquad (3)$$

Formulas (1), (2), and (3) are called *Green's formula.*

The simplest and most important case results if the bilinear form is symmetric, i.e., if the relations $b = c$, $\alpha = \gamma$, $\beta = \delta$ hold. In this case $L(u)$ is identical with $M(u)$—the self-adjoint case—and it can be derived from the quadratic expression

$$B(u, u) = au_x^2 + 2bu_xu_y + du_y^2 + 2\alpha u_xu + 2\beta u_yu + gu^2.$$

In the following we shall limit ourselves mainly to expressions $L(u)$ which are self-adjoint. The character of the difference expression $L(u)$ depends principally on the nature of those terms in the quadratic form $B(u, u)$ which are quadratic in the first difference quotients. We call this part of $B(u, u)$ the *characteristic form*:

$$P(u, u) = au_x^2 + 2bu_xu_y + du_y^2.$$

We call the corresponding difference expression $L(u)$ elliptic or hyperbolic, depending on whether the function $P(u, u)$ of the difference quotients is (positive) definite or indefinite.

The difference expression $\Delta u = u_{x\bar{x}} + u_{y\bar{y}}$ with which we shall concern ourselves in the following paragraph is elliptic, i.e., it comes from the quadratic expression

$$B(u, u) = u_x^2 + u_y^2 \quad \text{or} \quad u_{\bar{x}}^2 + u_{\bar{y}}^2.$$

The corresponding Green's formulas are

$$h^2 \sum_{G_h} \sum (u_x^2 + u_y^2) = -h^2 \sum_{G_h'} \sum u\Delta u - h \sum_{\Gamma_h} u\mathfrak{R}(u) \qquad \text{[Note 3]} \qquad (4)$$

$$h^2 \sum_{G_h'} \sum (v\Delta u - u\Delta v) + h \sum_{\Gamma_h} [v\mathfrak{R}(u) - u\mathfrak{R}(v)] = 0. \qquad (5)$$

The difference expression, $\Delta u = u_{x\bar{x}} + u_{y\bar{y}}$, is obviously the analogue of the differential expression $(\partial^2 u/\partial x^2) + (\partial^2 u/\partial y^2)$ for a function $u(x, y)$ of the continuous variables x and y. Written out explicitly the difference expression is

$$\Delta u = \frac{1}{h^2}[u(x + h, y) + u(x, y + h) + u(x - h, y) + u(x, y - h) - 4u(x, y)].$$

Therefore $(h^2/4)\Delta u$ is the excess of the arithmetic mean of the functional values at the four neighborhood points over the functional value at the point in question.

Completely similar considerations lead to linear difference expressions of the fourth order and corresponding Green's formula, provided one starts from bilinear difference expressions which are formed from the difference quotients of second order. Consider for example the difference expression

$$\Delta\Delta u = u_{xx\bar{x}\bar{x}} + 2u_{x\bar{x}y\bar{y}} + u_{yy\bar{y}\bar{y}}.$$

This corresponds to the quadratic expression

$$B(u, u) = (u_{x\bar{x}} + u_{y\bar{y}})^2 = (\Delta u)^2,$$

provided one orders the sum

$$h^2 \sum_{G_h'} \sum \Delta u \Delta v$$

according to v, or equivalently replaces u by Δu in Eq. (5). One must notice however that in the expression $\Delta\Delta u$, the functional value at a point is connected with the values at its neighboring points and at their neighboring points, and accordingly is defined only for such points of the region G_h as are also interior points of the region G_h' (See Section 5). The entirety of such points we designate as G_h''. We obtain then Green's formula

$$h^2 \sum_{G_h'} \sum \Delta u \cdot \Delta v = h^2 \sum_{G_h''} \sum v \cdot \Delta\Delta u + h \sum_{\Gamma_h + \Gamma_h'} v \cdot \mathfrak{R}(u), \qquad (6)$$

where $\mathfrak{R}(u)$ is a definable linear difference expression for each point of the boundary strip $\Gamma_h + \Gamma_h'$. Γ_h' indicates the set of boundary points of G_h'.

Section 2. Boundary value and eigenvalue problems

• 1. *The theory of boundary value problems*

The boundary value problem for linear elliptic homogeneous difference equations of second order, which corresponds to the classical boundary value problem for partial differential equations, can be formulated in the following way.

Let there be given a self-adjoint elliptic linear difference expression of second order, $L(u)$, in a mesh region, G_h. $L(u)$ results from a quadratic expression $B(u, u)$ which is positive definite in the sense that it cannot vanish if u_x and u_y do not themselves vanish.

A function, u, is to be determined satisfying in G_h the difference equation $L(u) = 0$, and assuming prescribed values at the boundary points.

Under these requirements there will be exactly as many linear equations as there are interior points of the mesh at which the function u is to be determined.[4] Some of these equations which involve only mesh points whose neighbors also lie in the interior of the region are homogeneous; others which involve boundary points of the mesh region are inhomogeneous. If the right-hand side of the inhomogeneous equations is set equal to zero, that is if $u = 0$ on the boundary, then it follows from Green's formula (1), by setting $u = v$ that $B(u, u)$ vanishes, and further, from the definiteness of $B(u, u)$, that u_x and u_y vanish, and hence that u itself vanishes. Thus the difference equation for zero boundary values has the solution $u = 0$, or in other words the solution is uniquely determined since the difference of two solutions with the

[3] The boundary expression $\mathfrak{R}(u)$ may be written as follows: Let $u_0, u_1, \ldots, u_\nu$ be values of the function at a boundary point and at its ν neighboring points ($\nu \leq 3$), then

$$\mathfrak{R}(u) = \frac{1}{h}(u_1 + \cdots + u_\nu - \nu u_0).$$

[4] If the matrix of the linear system of equations corresponding to an arbitrary difference equation of second order, $L(u) = 0$, is transposed, then the transposed set of equations corresponds to the adjoint difference equation $M(v) = 0$. Thus the above self-adjoint system gives rise to a set of linear equations with symmetric coefficients.

same boundary value must vanish. Further, if a linear system of equations with as many unknowns as equations is such that for vanishing right-hand side the unknowns must vanish, then the fundamental theorem of the theory of equations asserts that for an arbitrary right-hand side exactly one solution must exist. In our case there follows at once the existence of a solution of the boundary value problem.

Therefore we see that for elliptic difference equations the uniqueness and existence of the solution of the boundary value problem are related to each other through the fundamental theorem of the theory of linear equations, whereas for partial differential equations both facts must be proved by quite different methods. The basis for this difficulty in the latter case is to be found in the fact that the differential equations are no longer equivalent to a finite number of equations, and so one can no longer depend on the equality of the number of unknowns and equations.

Since the difference expression $\Delta u = 0$ can be derived from the positive definite quadratic expression

$$h^2 \sum_{G_h} \sum (u_x^2 + u_y^2),$$

the boundary value problem for the difference expression is uniquely solvable.

The theory for difference equations of higher order is developed in exactly the same way as that for difference equations of second order; for example one can treat the fourth order difference equation $\Delta\Delta u = 0$. In this case the values of the function must be prescribed on the boundary strip $\Gamma_h + \Gamma_h'$. Evidently here also the difference equation yields just as many linear equations as there are unknown functional values at the points of G_h''. In order to demonstrate the existence and uniqueness of the solution one needs only to show that a solution which has the value zero in the boundary strip $\Gamma_h + \Gamma_h'$ necessarily vanishes identically. To this end we note that the sum over the corresponding quadratic expression

$$h^2 \sum_{G_h''} \sum (\Delta u)^2 \tag{7}$$

for such a function vanishes, as can be seen by transforming the sum according to Green's formula (6). The vanishing of the sum (7) however implies that Δu vanishes at all points of G_h', and according to the above proof this can only happen for vanishing boundary values if the function u assumes the value zero throughout the region. Thus our assertion is proved, and both the existence and uniqueness of the solution to the boundary value problem for the difference equations are guaranteed.[5]

[5] For the actual process of carrying through the solution of the boundary value problem by an iterative method, see among others the treatment: "Über Randwertaufgaben bei partiellen Differenzengleichungen" by R. Courant, *Zeitschr. f. angew. Mathematik u. Mechanik* **6**, 322–325 (1925). Also there is a report by H. Henky, in *Zeitschr. f. angew. Math. u. Mech.* **2**, 58 ff (1922).

• 2. *Relation to the minimum problem*

The above boundary value problem is related to the following minimum problem: among all functions $\varphi(x, y)$ defined in the mesh region G_h and assuming given values at the boundary points, that function $\varphi = u(x, y)$ is to be found for which the sum

$$h^2 \sum_{G_h} \sum B(\varphi, \varphi)$$

over the mesh region assumes the least possible value. We assume that the quadratic difference expression of first order, $B(u, u)$ is positive definite in the sense described in Section 1, Part 2. One can show that the difference equation $L(\varphi) = 0$ results from this minimum requirement on the solution $\varphi = u(x, y)$, where $L(\varphi)$ is the difference expression of second order derived previously from $B(\varphi, \varphi)$. In fact this can be seen either by applying the rules of differential calculus to the sums $h^2 \sum_{G_h} \sum B(\varphi,\varphi)$ as a function of a finite number of values of φ at the grid points, or by employing the usual methods from the calculus of variations.

By way of example, solving the boundary value problem of finding the solution to the equation $\Delta\varphi = 0$ which assumes given boundary values, is equivalent to minimizing the sum $h^2 \sum_{G_h} \sum (\varphi_x^2 + \varphi_y^2)$ over all functions which take on the boundary values.

There is a similar correspondence for the fourth-order difference equations, where we limit ourselves to the example $\Delta\Delta\varphi = 0$. The boundary value problem corresponding to this difference equation is equivalent to the problem of minimizing the sum

$$h^2 \sum_{G_h'} \sum (\Delta\varphi)^2$$

for functions that take on given values on the boundary strip Γ_h'. Besides this sum there are yet other quadratic expressions in the second derivatives which give rise to the equation $\Delta\Delta u = 0$ under the process of being minimized. For example this is true in G_h for the sum

$$h^2 \sum_{G_h'} \sum (u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2).$$

That the minimum problem posed above always has a solution follows from the theorem that a continuous function of a finite number of variables (the functional values of φ at the grid points) always has a minimum if it is bounded from below and if it tends to infinity as soon as any of the independent variables goes to infinity.[6]

• 3. *Green's function*

It is possible to treat the boundary value problem for the inhomogeneous equation, $L(u) = -f$, in much the same way as the homogeneous case, $L(u) = 0$. For the inhomogeneous case it is sufficient to consider only the case of

[6] It can easily be verified that the hypotheses for the application of this theorem are satisfied.

zero boundary conditions, since different boundary conditions can be taken care of by adding a suitable solution of the homogeneous equation. To solve the linear system of equations representing the boundary value problem, $L(u) = -f$, we first choose as the function $f(x, y)$ a function which assumes the value $-1/h^2$ at the point $x = \xi$, $y = \eta$ of the mesh. If $K(x, y; \xi, \eta)$ is the solution (vanishing on the boundary) of this difference equation which depends on the parametric point (ξ, η), then the solution for an arbitrary boundary condition can be represented by the sum

$$u(x, y) = h^2 \sum_{(\xi,\eta) \text{ in } G_h} K(x, y; \xi, \eta) f(\xi, \eta).$$

The function $K(x, y; \xi, \eta)$ which depends on the points (x, y) and (ξ, η) is called the Green's function of the differential expressions $L(u)$. If we call the Green's function for the adjoint expression $M(v)$, $\bar{K}(x, y; \xi, \eta)$, then the equivalence

$$K(\bar{\xi}, \bar{\eta}; \xi, \eta) = \bar{K}(\xi, \eta; \bar{\xi}, \bar{\eta}) \quad \text{holds,}$$

as can be seen to follow immediately from Green's formula (3) when $u = K(x, y; \xi, \eta)$, and $v = \bar{K}(x, y; \bar{\xi}, \bar{\eta})$. For a self-adjoint difference expression the above relation gives the symmetric expression

$$K(\bar{\xi}, \bar{\eta}; \xi, \eta) = K(\xi, \eta; \bar{\xi}, \bar{\eta}).$$

• 4. *Eigenvalue problems*

Self-adjoint difference expressions, $L(u)$, give rise to eigenvalue problems of the following type: find the value of a parameter λ, the eigenvalue, such that in G_h, a solution, the eigenfunction can be found for the difference equation $L(u) + \lambda u = 0$, where u is to be zero on the boundary, Γ_h.

The eigenvalue problem is equivalent to finding the principle axes of the quadratic form $B(u, u)$. Exactly as many eigenvalues and corresponding eigenfunctions exist as there are interior mesh points of the region G_h. The system of eigenfunctions and eigenvalues (and a proof of their existence) is given by the minimum problem:

Among all functions, $\varphi(x, y)$, vanishing on the boundary, and satisfying the orthogonality relation

$$h^2 \sum_{G_h} \sum \varphi u^{(\nu)} = 0, \qquad (\nu = 1, \cdots, m-1)$$

and normalized such that

$$h^2 \sum_{G_h} \sum \varphi^2 = 1,$$

the function, $\varphi = u$, is to be found for which the sum

$$h^2 \sum_{G_h} \sum B(\varphi, \varphi)$$

assumes its minimum value. The value of this minimum is the mth eigenvalue, and the function for which it is assumed is the mth eigenfunction.[7]

Section 3.[8] *Connections with the problem of the random walk*
The theme of the following is related to a question from the theory of probability, namely the problem of the random walk in a bounded region.[9] We consider the lines of a mesh region G_h as paths along which a particle can move from one grid point to a neighboring one. In this net of streets the particle can wander aimlessly, and it can choose at random one of the four directions leading from each intersection of paths of the net—all four directions being equally probable. The walk ends as soon as a boundary point of G_h is reached because here the particle must be absorbed.

We ask:

1) What is the probability $w(P; R)$ that a random walk starting from a point P reaches a particle point R of the boundary?
2) What is the mathematical expectation $v(P; Q)$ that a random walk starting from P reaches a point Q of G_h without touching the boundary?

This probability or mathematical expectation, respectively, will be defined more precisely by the following process. Assume that at the point P there is a unit concentration of matter. Let this matter diffuse into the mesh with constant velocity, traveling say a mesh width in unit time. At each meshpoint let exactly one-fourth of the matter at the point diffuse outwards in each of the four possible directions. The matter which reaches a boundary point is to remain at that point. If the point of origin P is itself a boundary point, then the matter never leaves that point.

We define the probability $w(P; R)$ that a random walk starting from P reaches the boundary point R (without

[7] From the orthogonality condition on the eigenfunctions,

$$h^2 \sum_{G_h} \sum u^{(\nu)} u^{(\mu)} = 0, \qquad (\nu \neq \mu)$$

it follows that each function, $g(x, y)$, which vanishes on the boundary of the mesh can be expanded in terms of the eigenfunctions in the form

$$y = \sum_{\nu=1}^{N} c^{(\nu)} u^{(\nu)},$$

where the coefficients are determined from the equations

$$c^{(\nu)} = h^2 \sum_{G_h} \sum g u^{(\nu)}.$$

In this way in particular the following representation for the Green's functions may be derived,

$$K(x, y; \xi, \eta) = -\sum_{\nu=1}^{N} \frac{u^{(\nu)}(x, y) \cdot u^{(\nu)}(\xi, \eta)}{\lambda^{(\nu)}}.$$

[8] Section 3 is not prerequisite to Section 4.

[9] The present treatment is essentially different from the familiar treatments which can be carried through, say for example in the case of Brownian motion for molecules. The difference lies precisely in the way in which the boundary of the region enters.

previously attaining the boundary), as the amount of matter which accumulates at this boundary point over an infinite amount of time.

We define the probability $E_n(P; Q)$ that the walk starting from the point P reaches the point Q in exactly n steps without touching the boundary by the amount of matter which accumulate in n units of time provided P and Q are both interior points. If either P or Q are boundary points then E_n is set equal to zero.

The value $E_n(P; Q)$ is exactly equal to $1/4^n$ times the number of paths of n steps leading from P to Q without touching the boundary. Thus $E_n(P; Q) = E_n(Q; P)$.

We define the mathematical expectation $v(P; Q)$ that one of the paths considered above leads from P to the point Q by the infinite sum of all of these possibilities,

$$v(P; Q) = \sum_{\nu=0}^{\infty} E_\nu(P; Q) \qquad \text{(Note 10)},$$

i.e., for the interior points P and Q, the sum of all the concentrations which have passed through the point Q at different times. This will be assigned the expected value 1 for a concentration originating at Q.

If the amount arriving at the boundary point R in exactly n steps is designated as $F_n(P; R)$, then the probability $w(P; R)$ is given by the series

$$w(P; R) = \sum_{\nu=0}^{\infty} F_\nu(P; R).$$

All the terms of this series are positive and the partial sum is bounded by one (since the concentration reaching the boundary can be made up of only part of the initial concentration), and therefore the convergence of the series is assured.

Now it is easy to see that the probability $E_n(P; Q)$, that is, the concentration reaching the point Q in exactly n steps tends to zero as n increases. For if at any point Q, from which the boundary point R can be reached in m steps, we have $E_n(P; Q) > \alpha > 0$, then at least $\alpha/4^m$ of the concentration at Q will reach the point R after m steps. However, since the sum $\sum_{\nu=0}^{\infty} F_\nu(P; R)$ converges, the concentration reaching R goes to zero with increasing time, where the value of $E_n(P; Q)$ must itself vanish as time increases; that is, the probability of an infinitely long walk remaining in the interior of the region is zero.

From this it follows that the entire concentration eventually reaches the boundary; or in other words that the sum $\sum_R w(P; R)$ over all the boundary points R is equal to one.

The convergence of the infinite series for the mathematical expectation

$$v(P; Q) = \sum_{\nu=0}^{\infty} E_\nu(P; Q)$$

remains to be shown.

[10] The convergence will be shown below.

To this end we remark that the quantity $E_n(P; Q)$ satisfies the following relations

$$E_{n+1}(P; Q) = \tfrac{1}{4}\{E_n(P; Q_1) + E_n(P; Q_2) + E_n(P; Q_3) + E_n(P; Q_4)\}, \quad [n \geq 1],$$

where Q_1 through Q_4 are the four neighboring points of the interior point Q. That is, the concentration at the point Q at the $n + 1^{\text{st}}$ step consists of $1/4$ times the sum of the concentrations at the four neighboring points at the nth step. If one of the neighbors of Q, for example $Q_1 = R$, is a boundary point then it follows that no concentration flows from this boundary point to Q since the expression $E_n(P; R)$ is zero in this case. Furthermore, for an interior point, $E_0(P; P) = 1$ and of course $E_1(P; Q) = 0$.

From these relationships we find for the partial sum

$$v_n(P; Q) = \sum_{\nu=0}^{n} E_\nu(P; Q)$$

the equation

$$v_{n+1}(P; Q) = \tfrac{1}{4}\{v_n(P; Q_1) + v_n(P; Q_2) + v_n(P; Q_3) + v_n(P; Q_4)\},$$

for $P \neq Q$. For the case of $P = Q$,

$$v_{n+1}(P; P) = 1 + \tfrac{1}{4}\{v_n(P; P_1) + v_n(P; P_2) + v_n(P; P_3) + v_n(P; P_4)\},$$

that is, the expectation that a point goes back into itself consists of the expectation that a nonvanishing path leads from P back again to itself—namely,

$$\tfrac{1}{4}\{v_n(P; P_1) + v_n(P; P_2) + v_n(P; P_3) + v_n(P; P_4)\},$$

together with the expectation unity that expresses the initial position of the concentration itself at this point.

The quantity $v_n(P; Q)$ thus satisfies the following difference equation[11]

$$\Delta v_n(P; Q) = \frac{4}{h^2} E_n(P; Q), \qquad \text{for} \quad P \neq Q,$$

$$\Delta v_n(P; Q) = \frac{4}{h^2} \{E_n(P; Q) - 1\}, \quad \text{for} \quad P = Q.$$

$v_n(P; Q)$ is equal to zero when Q is a boundary point.

[11] This defines the Δ-operation for the variable point Q. This equation can be interpreted as an equation of the heat conduction type. That is, if the function $v_n(P; Q)$ is considered, not as a function of the index n as in our presentation above, but rather as a function of time, t, which is proportional to n, so that $t = n\tau$ and $v_n(P; Q) = v(P; Q; t) = v(t)$, then the above equations can be written in the following form:

$$\Delta v(t) = \frac{4\tau}{h^2} \cdot \frac{v(t + \tau) - v(t)}{\tau} \qquad \text{for} \quad P \neq Q,$$

$$\Delta v(t) = \frac{4\tau}{h^2} \cdot \left(\frac{v(t + \tau) - v(t)}{\tau} - \frac{1}{\tau}\right) \quad \text{for} \quad P = Q.$$

For a similar difference equation which has a parabolic differential equation as its limiting form, see Section 6 of the second half of the paper.

The solution of this boundary value problem for arbitrary right-hand side is uniquely determined as we have explained earlier (Section 2, Part 1), and depends continuously on the right-hand side. Since the variables $E_n(P; Q)$ tend to zero, the solution $v_n(P; Q)$ converges to the solution $v(P; Q)$ of the difference equation

$$\Delta v(P; Q) = 0 \qquad \text{for} \quad P \neq Q$$

$$\Delta v(P; Q) = -\frac{4}{h^2} \quad \text{for} \quad P = Q,$$

with boundary values $v(P; R) = 0$.

Thus we see that the mathematical expectation $v(P; Q)$ exists and is none other than the Green's function for the difference equation $\Delta u = 0$, except for a factor of 4. The symmetry of the Green's function is an immediate consequence of the symmetry of the quantity $E_n(P; Q)$ which was used to define it.

The probability $w(P; R)$ satisfies, with respect to P, the relation

$$w(P; R) = \tfrac{1}{4}\{w(P_1; R) + w(P_2; R) + w(P_3; R) + w(P_4; R)\},$$

and thus the difference equation $\Delta w = 0$. That is, if P_1, P_2, P_3, P_4, are the four neighboring points of the interior point P, then each path from P to R must pass through one of these four directions, and each of the four is equally likely. Furthermore, the probability of going from one boundary point R to another R' is $w(R; R') = 0$ unless the two points R and R' coincide, in which case $w(R; R) = 1$. Thus $w(P; R)$ is that solution of the boundary value problem $\Delta w = 0$ which assumes the value 1 at the boundary point R and the value 0 at all other points of the boundary. Therefore the solution of the boundary value problem for an arbitrary boundary value $u(R)$ has the simple form $u(P) = \sum_R w(P; R)u(R)$, where the sum is to be extended over all the boundary points.[12] If the function $u \equiv 1$ is substituted for u in this expression, then we check the relation $1 = \sum_R w(P; R)$.

The interpretation given above for Green's function as an expectation yields immediately further properties. We mention only the fact that the Green's function decreases if one goes from the region G to a subregion $\bar{G}$ lying within G; that is, the number of possible paths for steps on the mesh leading from one point P to another Q (without touching the boundary), decreases as the region decreases.

Of course, corresponding relationships hold for more than two independent variables. We note only that other elliptic difference equations admit a similar probability interpretation.

[12] Moreover it is easy to show that the probability $w(P; R)$ of reaching the boundary is the boundary expression $\mathfrak{R}(K(P, Q))$, constructed from the Green's function $K(P; Q)$ in terms of Q, where $u(x, y)$ is to be identified with $w(P, Q)$, and $v(x, y)$ with $v(P, Q)$ in Green's formula (5).

If the limit for vanishing mesh width is considered by methods given in the following section, then the Green's function on the mesh goes over to the Green's function of the potential equation except for a numerical factor; a similar relationship holds between the expression $w(P; R)/h$ and the normal derivative of the Green's function at the boundary of the region. In this way, for example, the Green's function for the potential equation could be interpreted as the specific mathematical expectation of going from one point to another,[13] without reaching the boundary.

In going over to the limit of a continuum from the mesh, the influence of the direction in the mesh prescribed for the random walk vanishes. This fact suggests that it would be of some interest to consider limiting cases of more general random walks for which the limitations on the direction of the walk are relaxed. This lies outside of the scope of this presentation, however, and we can only hope to renew the question at some other opportunity.

Section 4. The solution of the differential equation as a limiting form of the solution of the difference equation

• 1. *The boundary value problem of potential theory*

In letting the solution of the difference equation tend to the solution of the corresponding differential equation we shall relinquish the greatest possible degree of generality with regard to the boundary and boundary values in order to demonstrate more clearly the character of our method.[14] Therefore we assume that we are given a simply connected region G with a boundary formed of a finite number of arcs with continuously turning tangents. Let $f(x, y)$ be a given function which is continuous and has continuous partial derivatives of first and second order in a region containing G. If G_h is the mesh region with mesh width h belonging to the region G, then let the boundary value problem for the difference equation $\Delta u = 0$ be solved for the same boundary values which the function $f(x, y)$ assumes on the boundary; let $u_h(x, y)$ be the solution. We shall prove that as $h \to 0$ the function $u_h(x, y)$ converges to a function $u(x, y)$ which satisfies in G the partial differential equation $(\partial^2 u/\partial x^2) + (\partial^2 u/\partial y^2) = 0$ and takes the value of $f(x, y)$ at each of the points of the boundary. We shall show further that for any region lying entirely within G the difference quotients of u_h of arbitrary order tend uniformly towards the corresponding partial derivatives of $u(x, y)$.

In the convergence proof it is convenient to replace the boundary condition $u = f$ by the weaker requirement that

$$\frac{1}{r}\iint_{S_r} (u - f)^2 \, dxdy \to 0 \quad \text{as} \quad r \to 0,$$

[13] Here the *a priori* expectation of reaching a certain area element is understood to be equal to the area of the element.

[14] We mention however that carrying through our method for more general boundaries and boundary values in no way causes any fundamental difficulty.

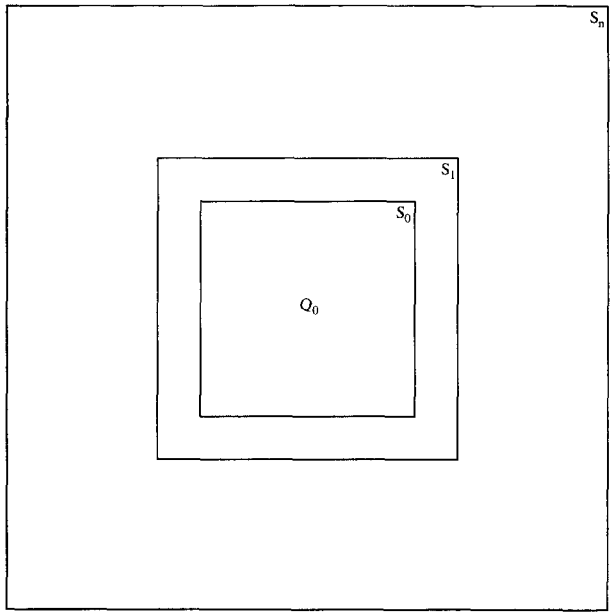


Figure 1

where S_r is that strip of G whose points are at a distance from the boundary smaller than r.[15] The convergence proof depends on the fact that for any subregion G^* lying entirely within G, the function $u_h(x, y)$ and each of its difference quotients is bounded and uniformly continuous as $h \to 0$ in the following sense: For each of these functions, say $w_h(x, y)$, there exists a $\delta(\varepsilon)$ depending only on the subregion and not on h such that

$$|w_h(P) - w_h(P_1)| < \varepsilon$$

provided the mesh points P and P_1 lie in the same subregion of G_h and are separated from each other by a distance less than $\delta(\varepsilon)$.

Once uniform continuity in this sense (equicontinuity) has been established we can in the usual way select from the functions u_h a subsequence of functions which tend uniformly in any subregion G^* to a limit function $u(x, y)$, while the difference quotients of u_h tend uniformly towards the partial derivatives of u. The limit function then possesses derivatives of arbitrarily high order in any proper subregion G^* of G and satisfies $\nabla^2 u = 0$ in this region. If we can show also that u satisfies the boundary condition we can regard it as the solution of our boundary value problem for the region G. Since this solution is uniquely determined, it is clear that not only a partial sequence of the functions u_h, but this sequence of functions itself possesses the required convergence properties.

The uniform continuity (equicontinuity) of our quantities may be established by proving the following lemmas.

1) As $h \to 0$ the sums over the mesh region $h^2 \sum_{G_h} \sum u^2$ and $h^2 \sum_{G_h} \sum (u_x^2 + u_y^2)$ remain bounded.[16]
2) If $w = w_h$ satisfies the difference equation $\Delta w = 0$ at a mesh point of G_h, and if, as $h \to 0$ the sum

$$h^2 \sum_{G_h^*} \sum w^2,$$

extended over a mesh region G_h^* associated with a subregion G^* of G remains bounded, then for any fixed subregion G^{**} lying entirely within G^* the sum

$$h^2 \sum_{G_h^{**}} \sum (w_x^2 + w_y^2)$$

over the mesh region G_h^{**} associated with G^{**} likewise remains bounded as $h \to 0$. Using 1) there follows from this, since all of the difference quotients w of the function u_h again satisfy the difference equation $\Delta w = 0$, that each of the sums $h^2 \sum_{G_h^*} \sum w^2$ is bounded.
3) From the boundedness of these sums there follows finally the boundedness and uniform continuity of all the difference quotients themselves.

• 2. *Proof of the lemmas*

The proof of 1) follows from the fact that the functional values u_h are themselves bounded. For the greatest (or least) value of the function is assumed on the boundary[17] and so is bounded by a prescribed finite value. The boundedness of the sum $h^2 \sum_{G_h} \sum (u_x^2 + u_y^2)$ is an immediate consequence of the minimum property of our mesh function formulated in Part 2 of Section 2 which gives in particular

$$h^2 \sum_{G_h} \sum (u_x^2 + u_y^2) \leq h^2 \sum_{G_h} \sum (f_x^2 + f_y^2),$$

but as $h \to 0$ the sum on the right tends to the integral

$$\iint_G \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 dxdy \right]$$

which, by hypothesis, exists.

[15] The weaker boundary value requirement does in fact provide the unique characterization of the solution, as can be seen from the easily proved theorem: If the boundary condition above is satisfied for $f(x, y) = 0$ for a function satisfying the equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

in the interior of G and if

$$\iint_G \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 \right] dxdy$$

exists, then $u(x, y)$ is identically zero. (See Courant, "Über die Lösungen der Differentialgleichungen der Physik," *Math Ann.* **85**, 296 ff.)

In the case of two independent variables the boundary values are actually attained, as follows from the weaker requirement; but in the case of more variables the corresponding result cannot in general be expected since there can exist exceptional points on the boundary at which the boundary value is not taken on even though a solution exists under the weaker requirement.

[16] Here and in the following we drop the index h from the grid functions.

[17] We note, however, with a view to carrying over the method to other differential equations, that we can relax these conditions. To this end we need only to bring into play the inequality (15) or to use the reasoning of the alternative (see Part 4, Section 4).

To prove 2) we consider the quadratic sum

$$h^2 \sum_{\text{int.}Q_1}\sum (w_x^2 + w_{\bar{x}}^2 + w_y^2 + w_{\bar{y}}^2),$$

where the summation extends over all the interior points of a square Q_1, (see Fig. 1). We denote the values of the function on the boundary S_1 of the square Q_1, by w_1, and those on the boundary S_0, of Q_0, by w_0. Then Green's formula gives

$$h^2 \sum_{\text{int.}Q_1}\sum (w_x^2 + w_{\bar{x}}^2 + w_y^2 + w_{\bar{y}}^2) \tag{8}$$
$$= \sum_{S_1} w^2 - \sum_{S_0} w^2 - \sum_{C_1} w^2$$

where S_1 and S_0 are respectively the boundary of Q_1 and the square boundary of the lattice points Q_0 lying within S_1, while C_1 consists of the four corner points of the boundary of Q_1.

We now consider a sequence of concentric squares $Q_0, Q_1, \cdots, Q_n$ with boundaries $S_0, S_1, \cdots, S_n$, where each boundary is separated from the next by a mesh width. Applying the formula to each of these squares and observing that we have always

$$2h^2 \sum_{Q_0}\sum (w_x^2 + w_y^2)$$
$$\leq h^2 \sum_{Q_k}\sum (w_x^2 + w_{\bar{x}}^2 + w_y^2 + w_{\bar{y}}^2), \qquad (k \geq 1)$$

we obtain

$$2h^2 \sum_{Q_0}\sum (w_x^2 + w_y^2)$$
$$\leq \sum_{S_k} w^2 - \sum_{S_{k-1}} w^2 - \sum_{C_k} w^2 \qquad (1 \leq k < n),$$

where C_k consists of the four corner points of the boundary Q_k.

We strengthen the inequality by neglecting the last nonpositive term on the right and we then add the n inequalities to obtain

$$2nh^2 \sum_{Q_0}\sum (w_x^2 + w_y^2) \leq \sum_{S_n} w^2 - \sum_{S_0} w^2 \leq \sum_{S_n} w^2.$$

Summing this inequality from $n = 1$ to $n = N$ we get

$$N^2h^2 \sum_{Q_0}\sum (w_x^2 + w_y^2) \leq \sum_{Q_N}\sum w^2.$$

Diminishing the mesh width h we can make the squares Q_0 and Q_n converge towards two fixed squares lying within G and having corresponding sides separated by a distance a. In this process $Nh \to a$ and we have, independent of the mesh width

$$h^2 \sum_{Q_0}\sum (w_x^2 + w_y^2) \leq \frac{h^2}{a^2} \sum_{Q_N}\sum w^2. \tag{9}$$

For sufficiently small h this inequality holds of course not only for two squares Q_0 and Q_N but with a change in the constant, a, for any two subregions of G such that one is contained entirely within the other. Thus lemma (2) is proved.[18]

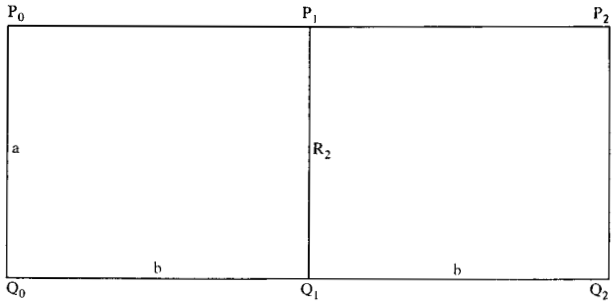


Figure 2

In order to prove the third result, that u_h and all its partial difference quotients w_h remain bounded and uniformly continuous as $h \to 0$, we consider a rectangle R (Fig. 2) with corners P_0, Q_0, P, Q and with sides P_0Q_0 and PQ of length a parallel to the x-axis.

We start with the relation

$$w(Q_0) - w(P_0) = h \sum_{PQ} w_x - h^2 \sum_R\sum w_{xy},$$

and the inequality

$$|w(Q_0) - w(P_0)| \leq h \sum_{PQ} |w_x| + h^2 \sum_R\sum |w_{xy}|, \tag{11}$$

which is a consequence of it.

We then let the side PQ of the rectangle vary between an initial line P_1Q_1, a distance b from P_0Q_0 and a final line P_2Q_2 a distance $2b$ from P_0Q_0, and we sum the corresponding $(b/h) + 1$ inequalities (11). We obtain the estimate

$$|w(P_0) - w(Q_0)| \leq \frac{1}{b+h} h^2 \sum_{R_2}\sum |w_x|$$
$$+ h^2 \sum_{R_2}\sum |w_{xy}|$$

where the summations extend over the entire rectangle, $R_2 = P_0Q_0P_2Q_2$. From Schwarz's inequality it then follows that,

$$|w(P_0) - w(Q_0)| \leq \frac{1}{b}\sqrt{2ab}\sqrt{h^2 \sum_{R_2}\sum w_x^2} \tag{12}$$
$$+ \sqrt{2ab}\sqrt{h^2 \sum_{R_2}\sum w_{xy}^2}.$$

[18] If we do not assume that $\Delta w = 0$, then in place of the inequality (9) we find:

$$h^2 \sum_{G^{**}}\sum (w_x^2 + w_y^2)$$
$$\leq c_1h^2 \sum_{G^*}\sum w^2 + c_2h^2 \sum_{G^*}\sum (\Delta w)^2 \tag{10}$$

for suitable constants c_1 and c_2 independent of h, where G^{**} lies entirely within G^*, and G^* in turn is contained in the interior of G.

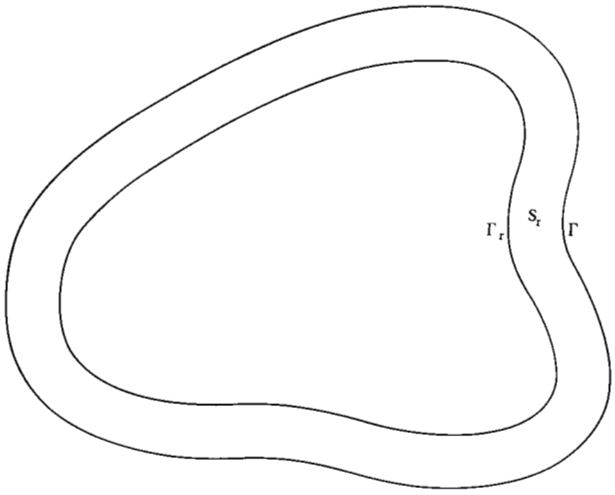


Figure 3

Since, by hypothesis, the sums which occur here multiplied by h^2 remain bounded, it follows that as $a \to 0$ the difference $|w(P_0) - w(Q_0)| \to 0$ independently of the mesh-width, since for each subregion G^* of G the quantity b can be held fixed. Consequently the uniform continuity (equicontinuity) of $w = w_h$ is proved for the x-direction. Similarly it holds for the y-direction and so also for any subregion G^* of G. The boundedness of the function w_h in G^* finally follows from its uniform continuity (equicontinuity) and the boundedness of $h^2 \sum_{G^*} \sum w_h^2$.

By this proof we establish the existence of a subsequence of functions u_h which converge towards a limit function $u(x, y)$ and which do so uniformly together with all their difference quotients, in the sense discussed above for every interior subregion of G. This limit function $u(x, y)$ has throughout G continuous partial derivatives of arbitrary order, and satisfies there the potential equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0.$$

• 3. *The boundary condition*

In order to prove that the solution satisfies the boundary condition formulated above we shall first of all establish the inequality

$$h^2 \sum_{S_{r,h}} \sum v^2 \leq Ar^2h^2 \sum_{S_{r,h}} \sum (v_x^2 + v_y^2) + Brh \sum_{\Gamma_h} v^2 \tag{13}$$

where $S_{r,h}$ is that part of the mesh region G_h which lies inside a boundary strip S_r. This boundary strip S_r consists of all points of G whose distance from the boundary is less than r; it is bounded by Γ and another curve Γ_r (Fig. 3). The constants A and B depend only on the region and not on the function v nor the mesh width h.

In order to prove the above inequality, we divide the boundary, Γ, of G into a finite number of pieces for which the angle of the tangent with one of the x- or y-axes is greater than some positive value (say 30°). Let γ, for instance, be a piece of Γ which is sufficiently steep (in the above sense) relative to the x-axis (see Fig. 4). Lines parallel to the x-axis intersecting γ will cut a section γ_r from the neighboring curve Γ_r, and will define together with γ and γ_r a piece s_r of the boundary strip S_r. We use the symbol $s_{r,h}$ to denote the portion of G_h contained in s_r and denote the associated portion of the boundary Γ_h by γ_h.

We now imagine a parallel to the x-axis to be drawn through a mesh point P_h of $s_{r,h}$. Let it meet the boundary γ_h in a point $\bar{P}_h$. The portion of this line which lies in $s_{r,h}$ we call $p_{r,h}$. Its length is certainly smaller than cr, where the constant c depends only on the smallest angle of inclination of a tangent γ to the x-axis.

Between the values of v at P_h and $\bar{P}_h$ we have the relation

$$v(P_h) = r(\bar{P}_h) \pm h \sum_{P_h\bar{P}_h} v_x.$$

Squaring both sides and applying Schwarz's inequality, we obtain

$$v(P_h)^2 \leq 2v(\bar{P}_h)^2 + 2cr \cdot h \sum_{p_{r,h}} v_x^2.$$

Summing with respect to P_h in the x-direction, we get

$$h \sum_{p_r} v^2 \leq 2crv(\bar{P}_h)^2 + 2c^2r^2h \sum_{p_r} v_x^2.$$

Summing again in the y-direction we obtain the relation

$$h \sum_{S_{r,h}} \sum v^2 \leq 2cr \sum_{\Gamma_h} v(\bar{P}_h)^2 + 2c^2r^2h \sum_{S_{r,h}} \sum v_x^2. \tag{14}$$

Writing down the inequalities associated with the other portions of Γ and adding all the inequalities together we obtain the desired inequality (13).[19]

We next set $v_h = u_h - f_h$ so that $v_h = 0$ on Γ_h.

Then since $h^2 \sum_{G_h} \sum (v_x^2 + v_y^2)$ remains bounded as $h \to 0$, we obtain from (13)

$$\frac{h^2}{r} \sum_{S_{r,h}} \sum v^2 \leq \kappa r, \tag{16}$$

where κ is a constant which does not depend on the function v or the mesh size. Extending the sum on the left to the difference $S_{\nu,h} - S_{\rho,h}$ of two boundary strips, the inequality (16) still holds with the same constant κ and we can pass to the limit $h \to 0$.

[19] By similar reasoning we can also establish the inequality

$$h^2 \sum_{G_h} \sum v^2 \leq c_1 h \sum_{\Gamma_h} v^2 + c_2 h^2 \sum_{G_h} \sum (v_x^2 + v_y^2) \tag{15}$$

in which the constants c_1 and c_2 depend only on the region G and not on the mesh division.

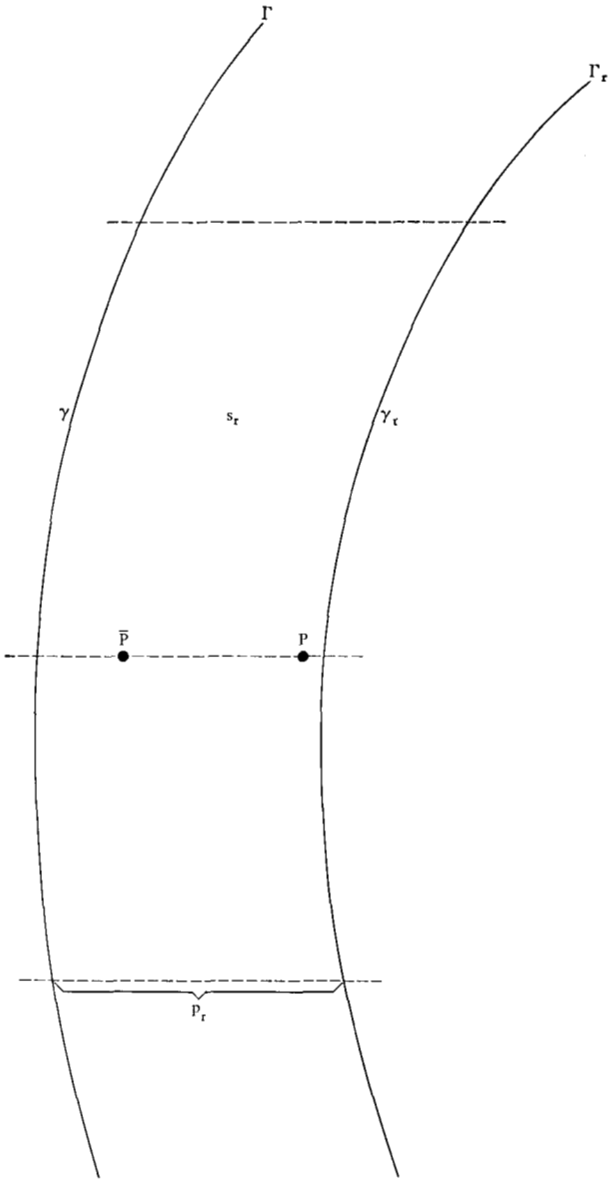


Figure 4

From the inequality (16) we then get

$$\frac{1}{r}\iint_{S_r - S_\rho} v^2\, dxdy \leq \kappa r, \qquad v = u - f.$$

Now letting the narrower boundary strip S_ρ approach the boundary we obtain the inequality

$$\frac{1}{r}\iint_{S_r} v^2\, dxdy = \frac{1}{r}\iint_{S_r} (u - f)^2\, dxdy \leq \kappa r$$

which states that the limit function u satisfies the prescribed boundary condition.

• 4. *Applicability of the method to other problems*

Our method is based essentially on the inequalities arising from the lemmas stated previously since the principal points of the proofs follow from the two last lemmas in Part 1 of Section 4.[20] No use is made of special fundamental solutions or special properties of the difference expression, and therefore the method can be carried over directly to the case of arbitrarily many independent variables as well as to the eigenvalue problem,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \lambda u = 0.$$

The same sort of convergence relations will obtain in this case as above.[21] Also the method applies to linear partial differential equations of other types, in particular its application to equations with variable coefficients requires only some minor modifications. The essential difference in this case lies only in proving the boundedness of $h^2 \sum\sum u_h^2$ which of course does not always hold for an arbitrary linear problem. However in case this sum is not bounded it can be shown that the general boundary value problem for the differential equation in question also possesses effectively no solutions, but that in this case there exist nonvanishing solutions of the corresponding homogeneous problem, i.e., eigenfunctions.[22]

• 5. *The boundary value problem* $\Delta\Delta u = 0$

In order to show that the method can be carried over to the case of differential equations of higher order, we will treat briefly the boundary value problem of the differential equation:

$$\frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2\, \partial y^2} + \frac{\partial^4 u}{\partial y^4} = 0.$$

We seek, in G, a solution of this equation for which the values of u and its first derivative are given on the boundary, being specified there by some function $f(x, y)$.

To this end we assume as previously that $f(x, y)$ together with its first and second derivatives is continuous in that region of the plane containing the region G.

We replace our differential equation problem by the new problem of solving the difference equation $\Delta u = 0$ in the mesh region G subject to the condition that at the points of the boundary strip $\Gamma_h + \Gamma_h'$ the function u takes on the values $f(x, y)$. From Section 2 we know that this boundary value problem has a unique solution. We will show that as the mesh size decreases, this solution, in each

[20] For an application of corresponding integral inequalities see K. Friedrichs, "Die Rand- und Eigenwertprobleme aus der Theorie der elastischen Platten", *Math. Ann.* 98, 222 (1926).

[21] Similarly one proves at the same time that every solution of such a differential equation problem has derivatives of every order.

[22] See Courant-Hilbert, *Methoden der Mathematischen Physik* 1, Ch. III, Section 3, where the theory of integral equations is handled with the help of the corresponding principle of the alternative. See also the Dissertation (Göttingen) of W. v. Koppenfels, which will appear soon.

interior subregion of G, converges to the solution of the differential equation, and that all of its difference quotients converge to the corresponding partial derivatives.

We note first that for the solution $u = u_h$, the sum

$$h^2 \sum_{G_h'}\sum (u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2)$$

remains bounded as $h \to 0$. That is, by applying the minimum requirements on the solution (Part 2, Section 2) one finds that this sum is not larger than the corresponding sum

$$h^2 \sum_{G_h'}\sum (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2),$$

and this converges as $h \to 0$ to

$$\iint_G \left(\frac{\partial^2 f}{\partial x^2} + 2\frac{\partial^2 f}{\partial x \partial y} + \frac{\partial^2 f}{\partial y^2}\right) dxdy$$

which exists, by hypothesis.

From the boundedness of the sum

$$h^2 \sum_{G_h'}\sum (u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2)$$

follows immediately the boundedness of $h^2 \sum_{G_h'} \sum (\Delta u)^2$ and also that of

$$h^2 \sum_{G_h'}\sum (u_x^2 + u_y^2) \quad \text{and} \quad h^2 \sum_{G_h}\sum u^2.$$

That is, for arbitrary w the following inequality holds (see Footnote 19),

$$h^2 \sum_{G_h}\sum w^2 \leq ch^2 \sum_{G_h}\sum (w_x^2 + w_y^2) + ch \sum_{\Gamma_h} w^2. \tag{15}$$

Then if one substitutes the first difference quotients of w for w itself in this inequality and applies the expression over the subregion of G_h for which they are defined, one finds the further inequality,

$$\begin{aligned} h^2 \sum_{G_h}\sum (w_x^2 + w_y^2) &\leq ch^2 \sum_{G_h'}\sum (w_{xx}^2 + 2w_{xy}^2 + w_{yy}^2) \\ &\quad + ch \sum_{\Gamma_h + \Gamma_h'} (w_x^2 + w_y^2), \end{aligned}$$

where again the constant c is independent of the function and of the mesh size. We apply this inequality to $w = u_h$ and thus find the boundedness of the sum over $\Gamma_h + \Gamma_h'$ on the right-hand side; by definition these boundary sums converge to the corresponding integral containing $f(x, y)$. Thus from the boundedness of

$$h^2 \sum_{G_h'}\sum (u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2)$$

follows the boundedness of $h^2 \sum_{G_h} \sum (u_x^2 + u_y^2)$ and thence that of $h^2 \sum_{G_h} \sum u^2$.

For the third step we substitute one after the other the expressions $\Delta u, \Delta u_x, \Delta u_y, \Delta u_{xx}, \cdots$, for w in the inequality

$$h^2 \sum_{G^{**}}\sum (w_x^2 + w_y^2) \leq ch^2 \sum_{G^*}\sum w^2 + ch^2 \sum_{G^*}\sum (\Delta w)^2$$

(see Part 2, Section 4) where G^* is a subregion of G containing G^{**} in its interior. The expressions introduced all satisfy the equation $\Delta w = 0$. It follows then that for each expression in turn and for all subregions G^* of G that the sums, $h^2 \sum_{G^*}\sum (w_x^2 + w_y^2)$, that is, $h^2 \sum_{G^*}\sum (\Delta u_x^2 + \Delta u_y^2)$, $h^2 \sum_{G^*}\sum (\Delta u_{xx}^2 + \Delta u_{xy}^2)$, $\cdots$ are bounded together with the sums:

$$h^2 \sum_{G_h}\sum u^2, \qquad h^2 \sum_{G_h}\sum (u_x^2 + u_y^2),$$

$$\text{and} \quad h^2 \sum_{G_h}\sum (\Delta u)^2,$$

which are already known to be bounded.

Finally we substitute into the inequality (10), in place of w, the sequence of functions $u_{xx}, u_{xy}, u_{yy}, u_{xxx}, \cdots$, for which

$$h^2 \sum_{G_h^*}\sum (\Delta w)^2, \quad \text{i.e.} \quad h^2 \sum_{G_h^*}\sum (\Delta u_{xx})^2, \cdots$$

are bounded as shown above. We then find that for all subregions the sums

$$h^2 \sum_{G_h^*}\sum (u_{xxx}^2 + u_{xxy}^2), \quad h^2 \sum_{G_h^*}\sum (u_{xyx}^2 + u_{xyy}^2), \cdots$$

remain bounded.

From these facts we can now conclude as previously that from our sequence of mesh functions a subsequence can be chosen which in each interior subregion of G converges (together with all its difference quotients) uniformly to a limit function (or respectively its derivatives) which is continuous in the interior of G.

We have yet to show that this limit function which obviously satisfies the differential equation $\Delta\Delta u = 0$ also takes on the prescribed boundary conditions. For this purpose we say here only that, analogous to the foregoing, the expressions

$$\iint_{S_r} (u - f)^2 \, dxdy \leq cr^2,$$

$$\iint_{S_r} \left[\left(\frac{\partial u}{\partial x} - \frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y} - \frac{\partial f}{\partial y}\right)^2\right] dxdy \leq cr^2$$

hold.[23] That the limit function fulfills these conditions may be seen by carrying over the treatment in Part 3, Section 4 to the function u and its first difference quotient.

[23] That the boundary values for the function and its derivatives actually are assumed is not difficult to prove. See for instance the corresponding treatment of K. Friedrichs, *loc. cit.*

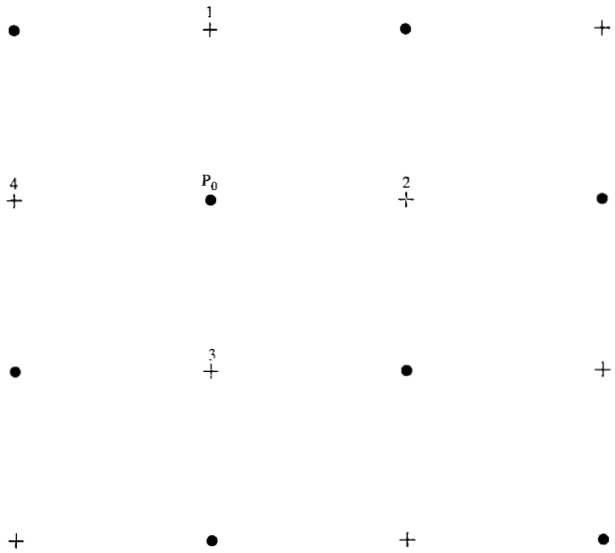


Figure 5

From the uniqueness of our boundary value problem we see furthermore that not only a selected subsequence, but also the original sequence of functions u possesses the asserted convergence properties.

II. The hyperbolic case

Section 1. The equation of the vibrating string

In the second part of this paper we shall consider the initial value problem for linear hyperbolic partial differential equations. We shall prove that under certain hypotheses the solutions of the difference equations converge to the solutions of the differential equations as the mesh size decreases.

We can discuss the situation most easily by considering the example of the approximation to the solution of the wave equation

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0 \tag{1}$$

We limit ourselves to the particular initial value problem where the value of the solution u, and its derivatives are given on the line $t = 0$.

In order to find the corresponding difference equation, we construct in the (x, t)-plane a square grid with lines parallel to the axes and with mesh width h. Following the notation of the first part of the paper we replace the differential equation (1) by the difference equation $u_{t\bar{t}} - u_{x\bar{x}} = 0$. If we select a grid point, P_0, then the corresponding difference equation relates the value of the function at this point to the values at four neighboring points. If we characterize the four neighboring values by the four indices 1, 2, 3, 4 (cf. Fig. 5), then the difference equation assumes the simple form

$$u_1 + u_3 - u_2 - u_4 = 0. \tag{2}$$

Note that the value of the function u at the point P_0 does not appear itself in the equation.

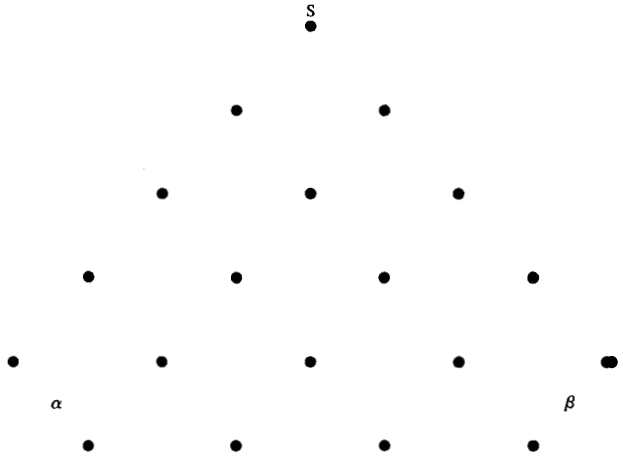
We consider the grid split up into two subgrids, indicated in Fig. 5 by dots and crosses respectively. The difference equation connects the values of the function over each of the subgrids separately, and so we shall consider only one of the two grids. As initial condition the values of the function are prescribed on the two rows of the grid, $t = 0$ and $t = h$. We can give the solution of this initial value problem explicitly; that is, we express the value of the solution at any point S in terms of the values given along the two initial rows. One can see at once that the value at a point of the row $t = 2h$ is uniquely determined by only the three values at the points close to it in the two first rows. The value at a point of the fourth row is uniquely determined by the values of the solution at three particular points in the second and third rows, and through them it is related to certain values in the first two rows. In general to a point S there will correspond a certain region of dependence in the first two rows; it may be found by drawing the lines $x + t =$ const. and $x - t =$ const., through the point S and extending them until they meet the second row at the points α and β respectively (cf. Fig. 6). The triangle $S\alpha\beta$ is called the *triangle of determination* because all the values of u in it remain unchanged provided the values on the first two rows of it are held fixed. The sides of the triangle are called *lines of determination.*

If one denotes the differences of u in the direction of the lines of determination by u' and $u^{\backslash}$, that is,

$$u_1' = u_1 - u_4, \qquad u_1^{\backslash} = u_1 - u_2$$

$$u_2' = u_2 - u_3, \qquad u_4^{\backslash} = u_4 - u_3,$$

Figure 6



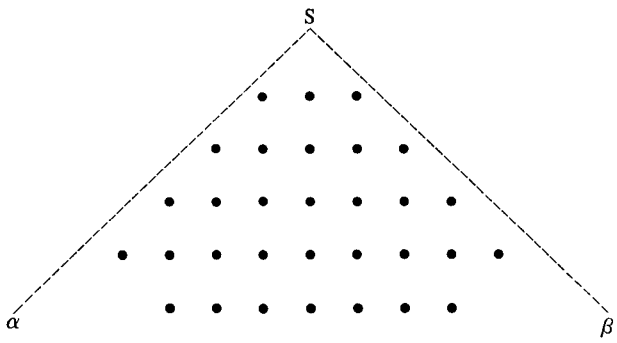


Figure 7

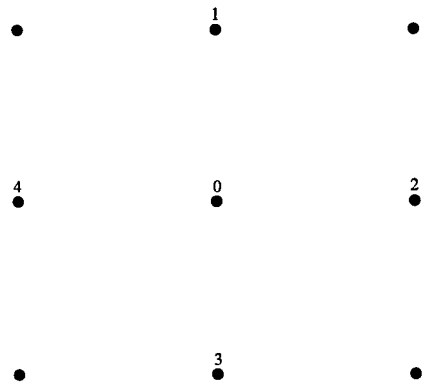


Figure 8

then the difference equation assumes the form

$$u_1' = u_2',$$

i.e., along a line of determination the differences with respect to the other direction of determination are constant, and thus are equal to one of the *given* differences between the value at two points on the first two rows. Moreover the difference $u_S - u_\alpha$ is a sum of differences u' along the determining line $\overline{\alpha S}$, so that using the remark above, we can obtain the final result (in obvious notation):

$$u_S = u_\alpha + \sum_{\alpha_1}^{\beta_1} u'. \tag{3}$$

We now let h go to zero, and let the prescribed values on the second and first rows converge uniformly to a twice continuously differentiable function, $f(x)$, and the difference quotients $u'/h\sqrt{2}$ there converge uniformly to a continuously differentiable function $g(x)$. Evidently the right-hand side of (3) goes over uniformly to the expression

$$f(x - t) + \frac{1}{\sqrt{2}} \int_{x-t}^{x+t} g(\xi)\, d\xi \tag{4}$$

if S converges to the point (x, t). This is the well-known expression for the solution of the wave equation (1) with initial values $u(x, 0) = f(x)$ and $\partial u(x, 0)/\partial t = f'(x) + \sqrt{2}g(x)$. Thus it is shown that as $h \to 0$ the solution of the difference equation converges to the solution of the differential equation provided the initial values converge appropriately (as above).

Section 2. On the influence of the choice of mesh. The domains of dependence of the difference and differential equations

The relationships developed in Section 1 suggest the following considerations.

In the same way that the solution of a linear hyperbolic equation at a point S depends only on a certain part of the initial line—namely the "domain of dependence" lying between the two characteristics drawn through S, the solution of the difference equation has also at the point S a certain domain of dependence defined by the lines of determination drawn through S. In Section 1 the directions for the lines of determination of the difference equation coincided with the characteristic directions for the differential equation so that the domains of dependence coincided in the limit. This result, however, was based essentially on the orientation of the mesh in the (x,t)-plane, and depended furthermore on the fact that a square mesh had been chosen. We shall now consider a more general rectangular mesh with mesh size equal to h (time interval) in the t-direction and equal to kh (space interval) in the x-direction, where k is a constant. The domain of dependence for the difference equation, $u_{t\bar{t}} - u_{x\bar{x}} = 0$ for this mesh will now either lie entirely within the domain of dependence of the differential equation, $\partial^2 u/\partial t^2 - \partial^2 u/\partial x^2 = 0$, or on the other hand will contain this latter region inside its own domain according as $k < 1$ or $k > 1$ respectively.

From this follows a remarkable fact: if for the case $k < 1$ one lets $h \to 0$, then the solution to the difference equation in general *cannot* converge to the solution of the differential equation. In this case a change in the initial values of the solution of the differential equation in the neighborhood of the endpoints α and β of the domain of dependence (see Fig. 7) causes, according to formula (4), a change in the solution itself at the point (x, t). For the solution of the difference equation at the point S, however, the changes at the points α and β are not relevant since these points lie outside of the domain of dependence of the difference equations. That convergence does occur for the case $k > 1$ will be proved in Section 3. See for example Fig. 9.

If we consider the differential equation

$$2\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 0 \tag{5}$$

in two space variables, x and y, and time, t, and if we replace it by the corresponding difference equation on a rectilinear grid, then in contrast to the case of only two independent variables it is impossible to choose the mesh division so that the domain of dependence of the difference and differential equations coincide, since the domain of dependence of the difference equation is a quadrilateral while that of the differential equation is a circle. Later (cf. Section 4) we shall choose the mesh division so that the domain of determination of the difference equation contains that of the differential equation in its interior, and shall show that once again convergence occurs.

On the whole an essential result of this section will be that in the case of each linear homogeneous hyperbolic equation of second order one can choose the mesh so that the solution of the difference equation converges to the solution of the differential equation as $h \to 0$, (see for instance Sections 3, 4, 7, 8).

Section 3. Limiting values for arbitrary rectangular grids

Now we consider further the wave equation

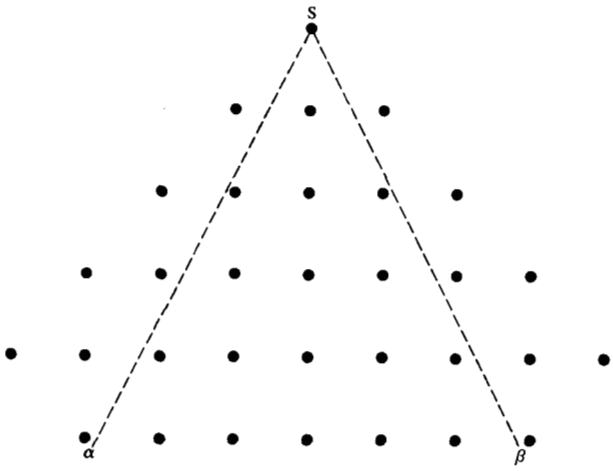
$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0, \tag{1}$$

but impose it now on a rectangular grid with time interval h and space interval kh. The corresponding difference equation is

$$L(u) = \frac{1}{h^2}(u_1 - 2u_0 + u_3) - \frac{1}{k^2h^2}(u_2 - 2u_0 + u_4) = 0, \tag{6}$$

where the indices represent a "fundamental rhombus" with midpoint P_0 and corners P_1, P_2, P_3, P_4 (see Fig. 8). According to the equation $L(u) = 0$ the value of the function u at a point S can be represented by its values on that section of the initial rows $t = 0$ and $t = h$ cut out by lines of determination through S parallel to the sides of an elementary rhombus. We assume that the initial values are prescribed in such a way that as $t \to 0$ for fixed k the first difference quotients formed from them converge uniformly to given continuous functions on the line $t = 0$. The initial values can be used to form an explicit representation of the solution of the difference equation (corresponding to (3) in Section 1); however it is too complicated to yield a limiting value easily as $h \to 0$. Thus we will try another approach which will also make it possible for us to treat the general problem.[24]

Figure 9



We multiply the difference expression $L(u)$ by $(u_1 - u_3)$ and form the product using the following identities:

$$(u_1 - u_3)(u_1 - 2u_0 + u_3) = (u_1 - u_0)^2 - (u_0 - u_3)^2, \tag{7}$$

$$(u_1 - u_3)(u_2 - 2u_0 + u_4) = (u_1 - u_0)^2 - (u_0 - u_3)^2 - \tfrac{1}{2}[(u_1 - u_2)^2 + (u_1 - u_4)^2 - (u_2 - u_3)^2 - (u_4 - u_3)^2]. \tag{8}$$

Then we obtain

$$2(u_1 - u_3)L(u) = \frac{2}{h^2}\left(1 - \frac{1}{k^2}\right)[(u_1 - u_0)^2 - (u_0 - u_3)^2] + \frac{1}{h^2k^2}[(u_1 - u_2)^2 + (u_1 - u_4)^2 - (u_2 - u_3)^2 - (u_4 - u_3)^2]. \tag{9}$$

The product (9) is now summed over all elementary rhombuses of the domain of determination, $S\alpha\beta$. The quadratic difference terms on the right-hand side always appear with alternate signs in two neighboring rhombuses so that they cancel out for any two rhombuses belonging to the triangle $S\alpha\beta$. Only the sums of squared differences over the "boundary" of the triangle remain, and we obtain the relation:

$$\begin{aligned} h^2 \sum_{S\alpha\beta}\sum 2\,\frac{u_1 - u_3}{h}\,L(u) &= h \sum_{S\alpha}\left[2\left(1 - \frac{1}{k^2}\right)\left(\frac{\dot u}{h}\right)^2 + \frac{1}{k^2}\left(\frac{u'}{h}\right)^2\right] \\ &+ h \sum_{S\beta}\left[2\left(1 - \frac{1}{k^2}\right)\left(\frac{\dot u}{h}\right)^2 + \frac{1}{k^2}\left(\frac{u'}{h}\right)^2\right] \\ &- h \sum_{\mathrm{I}\ \mathrm{II}}\left[2\left(1 - \frac{1}{k^2}\right)\left(\frac{\dot u}{h}\right)^2 + \frac{1}{k^2}\left(\frac{u'}{h}\right)^2 + \frac{1}{k^2}\left(\frac{u'}{h}\right)^2\right]. \end{aligned} \tag{10}$$

[24] For the following see K. Friedrichs and H. Lewy, "Über die Eindeutigkeit...etc.," *Math. Ann.* 98, 192 ff. (1928), where a similar transformation is used for integrals.

Here u' and $u^{\backslash}$ denote differences in the direction of determination defined in Section 1, while $\dot{u}$ designates the difference of the functional values at two neighboring points on a mesh line parallel to the t axis. The range in $\sum_{S\alpha}$ over which $(u')^2$ is taken is the outermost boundary line $S\alpha$ and its nearest parallel neighbor found by shifting the points of $S\alpha$ downward by the amount h. There is a similar range for $(u^{\backslash})^2$ in $\sum_{S\beta}$, and so all of the differences, u', $u^{\backslash}$, and $\dot{u}$ appear once and only once.

For the solution to the problem $L(u) = 0$ the right-hand side of (10) vanishes. The sum over the initial rows I and II which occurs there remains bounded as $h \to 0$ (for fixed k); specifically it goes over into an integral of the prescribed function along the initial line. Consequently the sums over $S\alpha$ and $S\beta$ in (10) also remain bounded. If now $k \geq 1$ as we must require (see previous discussion), then $1 - 1/k^2$ is non-negative, and we find that the individual sums

$$h \sum_{S\alpha} \left(\frac{u'}{h}\right)^2, \qquad h \sum_{S\beta} \left(\frac{u^{\backslash}}{h}\right)^2,$$

extended over any line of determination whatever, remain bounded.

From this we can derive the "uniform continuity" (equicontinuity) (cf. Section 4 of the first part of the paper) of the sequence of grid functions in all directions in the plane;[25] since the values of u on the initial line are bounded, there must exist a subset which converges uniformly to a limit function $u(x, t)$.

Both the first and second difference quotients of the function u also satisfy the difference equation $L(u) = 0$. Their initial values can be expressed through the equation $L(u) = 0$ in terms of the first, second and third difference quotients of u involving initial values at points on the two initial lines I and II only. We require that they tend to continuous limit functions, that is, that the given initial values $u(x, 0)$, $u_t(x, 0)$ be three times or respectively twice continuously differentiable with respect to x.

From this we can apply the convergence considerations set forth above to the first and second difference quotients of u, as well as to u itself, and we can choose a subsequence such that these difference quotients converge uniformly to certain functions, which must be the first or respectively second derivatives of the limit function $u(x, t)$. Hence the limit function satisfies the differential equation $\partial^2 u/\partial t^2 - \partial^2 u/\partial x^2 = 0$ which results as the limit of the difference equation $L(u) = 0$; it represents indeed the solution of the initial value problem. Since such a solution is uniquely determined, every subsequence of mesh functions converges, and therefore the sequence itself converges to the limit function.

[25] If S_1 and S_2 are two points at a distance δ, then one connects them by a path of two segments, S_1S and SS_2, where the former is parallel to one line of determination and the latter to the other. Then one finds the appraisal,

$$|u_{S_1} - u_S| \leq |u_{S_1} - u_S| + |u_S - u_{S_2}|$$
$$\leq \sqrt{\delta}\sqrt{h \sum_{S_1S} \left(\frac{u'}{h}\right)^2} + \sqrt{\delta}\sqrt{h \sum_{SS_2} \left(\frac{u^{\backslash}}{\delta}\right)^2}.$$

Section 4. The wave equation in three variables

We treat next the wave equation,

$$2\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 0 \tag{11}$$

and consider its relation to the observations on the domain of dependence discussed in Section 2. The domain of dependence of the differential equation (11) is a circular cone with axis parallel to the t-direction and with apex angle α, where $\tan \alpha = 1/\sqrt{2}$. In any rectilinear grid parallel to the axes we introduce the corresponding difference equation

$$2u_{t\bar{t}} - u_{x\bar{x}} - u_{y\bar{y}} = 0. \tag{12}$$

This equation relates the functional values of u at points of an elementary tetrahedron. In fact it allows the value of the function u at a point S to be expressed uniquely in terms of the values of the function at certain points of the two initial planes $t = 0$ and $t = h$. At each point S we obtain a pyramid of determination which cuts out from the two base planes two rhombuses as domains of dependence.

If we let the mesh widths tend to zero, keeping their ratios fixed, then we can expect convergence of the sequence of mesh functions to the solution of the differential equation only provided the pyramid of determination contains the cone of determination of the differential equation in its interior. The simplest grid with this property will be one constructed in such a way that the pyramid of determination is tangent to the exterior of the cone of determination. Our differential equation is chosen so that this occurs for a grid of cubes parallel to the axes.

The difference equation (12), in the notation of Fig. 10, assumes for such a grid the form:

$$L(u) = \frac{2}{h^2}(u'_\alpha - 2u_0 + u_\alpha) - \frac{1}{h^2}(u_1 - 2u_0 + u_3) - \frac{1}{h^2}(u_2 - 2u_0 + u_4), \tag{13}$$

in which the functional value, u_0, at the midpoint actually cancels out. The values of the solution on the two initial planes must be the values of a function having four continuous derivatives with respect to $x_1 y_2 t$.

For the convergence proof we again use the method developed in Section 3. We construct the triple sum

$$h^2 \sum \sum \sum 2\frac{u'_\alpha - u_\alpha}{h} L(u) = 0$$

for the solution to the difference equation, where the summation is to be extended over all elementary octahedrons of the pyramid of determination emanating from

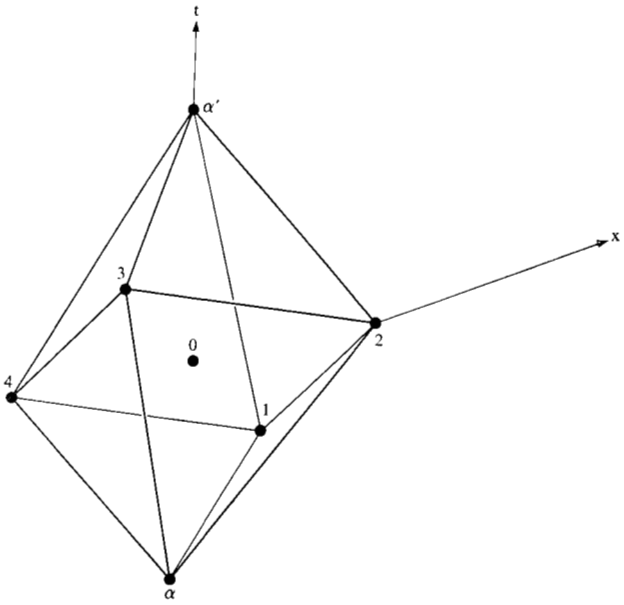


Figure 10

the point S. Then almost exactly as before we find that the values of the function u at the interior points of the pyramid of determination cancel out in the summation and that only the values on the two pyramids called F, and on the two base surfaces I and II remain.

If we denote by u' the difference of the values of the function at two points connected by a line of an elementary octahedron, then we can write the result as

$$\sum_F \sum (u')^2 - \sum_{\mathrm{I}} \sum_{\mathrm{II}} (u')^2 = 0, \tag{14}$$

where the sum is extended over all surfaces containing differences u'; each such difference is to appear only once.[26] The double sum over the two initial surfaces stays bounded since it goes over into an integral of the initial values. Therefore the sum over the "surface of determination" F remains bounded.

We now apply these results not to u itself, but to its first, second and third difference quotients, which themselves satisfy the difference equation (13). Their initial values can be expressed using only values on the first two initial planes by means of (13) using first through fourth difference quotients. If $w = w_h$ is one of the difference quotients of any order up to third order, then we know that the sum $h^2 \sum \sum (w'/h)^2$ extended over a surface of determination remains bounded. From this it follows, through exactly the methods used in Section 4 of the first part of the paper, that the function u and its first and second difference quotients are uniformly continuous (equicontinuous). Thus there exists a sequence of mesh widths decreasing to zero such that these quantities, which are bounded initially, converge to continuous limit functions and, in fact, converge to the solution of the differential equation and to the first and second derivatives of this solution, all exactly as we found earlier (Section 3).

[26] The grid ratio has been chosen in such a way that the differences between values of u appearing on the two neighboring surfaces in F do not occur, (as they did in the general case in one dimension treated in Section 3).

Appendix. *Supplements and generalizations*

Section 5. Example of a differential equation of first order

We have seen in Section 2 that in the case when the region of dependence of the differential equation covers only a part of the region of dependence of the difference equation, the influence of the rest of the region is not included in the limit. We can demonstrate this phenomenon explicitly by the example of the differential equation of first order, $\partial u/\partial t = 0$ if we replace it by the difference equation

$$2u_t - u_x + u_{\bar{x}} = 0. \tag{15}$$

In the notation of Fig. 5 this becomes

$$u_1 = \frac{u_2 + u_4}{2}. \tag{16}$$

As before, the difference equation connects only the points of a submesh with one another. The initial value problem consists of assigning as initial values for u at points $x = 2ih$ on the row $t = 0$ the values, f_{2i}, assumed there by a continuous function $f(x)$.

We consider the point S at a distance $2nh$ up along the t-axis. It is easy to verify that the solution u at S is represented as

$$u_S = \sum_{i=-n}^{n} \frac{1}{2^{2n}} \begin{pmatrix} 2n \\ n+i \end{pmatrix} f_{2i}. \tag{17}$$

As the mesh size decreases, that is as $n \to \infty$, the sum on the right-hand side tends simply to the value f_0. This can be demonstrated from the continuity of $f(x)$ and from the behavior of the binomial coefficients as n increases (see the following paragraph).

Section 6. The equation of heat conduction

The difference equation (16) of Section 5 can also be interpreted as the analogue of an entirely different differential equation, namely the equation of heat conduction,

$$2\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0. \tag{18}$$

In any rectangular mesh with mesh spacing l and h in the time and space directions, respectively, the corresponding difference equation becomes

$$2\left(\frac{u_1 - u_0}{l}\right) = \left(\frac{u_2 + u_4 - 2u_0}{h^2}\right). \tag{19}$$

In the limit as the mesh size goes to zero the difference equation preserves its form only if l and h^2 are decreased

proportionately. In particular if we set $l = h^2$, then the value u_0 drops out of the equation and the difference equation becomes

$$u_1 = \frac{u_2 + u_4}{2}. \tag{16}$$

The solution to (16) is given by formula

$$u(0, t) = \sum_{i=-n}^{n} \frac{1}{2^{2n}} \begin{bmatrix} 2n \\ n + i \end{bmatrix} f_{2i}.$$

As the mesh width decreases, a point ξ on the x-axis is always characterized by the index

$$2i = \xi/h. \tag{20}$$

The mesh width h is related to the ordinate t of a particular point by

$$2nh^2 = t. \tag{21}$$

We shall investigate what happens to formula (17) as $h \to 0$, that is $n \to \infty$. Using (21) we write (17) in the form

$$u(0, t) = \sum_{i=-n}^{n} \frac{\sqrt{2n}}{2 \cdot 2^{2n} \sqrt{t}} \begin{bmatrix} 2n \\ n + i \end{bmatrix} f_{2i} \cdot 2h. \tag{22}$$

For the coefficient of $2hf_{2i} = 2hf(\xi)$ we use the abbreviation

$$\frac{1}{2\sqrt{t}} g_{2n}(\xi) = \frac{\sqrt{2n}}{2 \cdot 2^{2n} \sqrt{t}} \begin{bmatrix} 2n \\ n + \dfrac{\xi}{\sqrt{2t}} \sqrt{n} \end{bmatrix}.$$

The limiting value of the coefficient, which is usually determined by using Stirling's formula, we will calculate here by considering the function $g_{2n}(\xi)$ as the solution of an ordinary difference equation which approaches a differential equation as $h \to 0$. As the difference equation one finds

$$\frac{1}{2h} [g_h(\xi + 2h) - g_h(\xi)] = -\frac{1}{2h} g_h(\xi) \frac{2i + 1}{n + i + 1}$$

(in which we have written $g_h(\xi)$ instead of $g_{2n}(\xi)$). Or

$$\frac{1}{2h} [g_h(\xi + 2h) - g_h(\xi)] = -g_h(\xi) \frac{\xi + h}{t + h\xi + 2h^2}.$$

$g_h(\xi)$ satisfies the normalization condition

$$\sum_{i=-n}^{n} g_h(\xi) \cdot 2h = 2\sqrt{t}.$$

This sum is over the region of dependence of the difference equation, and as $h \to 0$ this covers the entire x-axis.

It can be shown easily that $g_h(\xi)$ converges uniformly to the solution $g(x)$ of the differential equation

$$g'(x) = -g(x)x/t$$

with the auxiliary condition

$$\int_{-\infty}^{\infty} g(x)\, dx = 2\sqrt{t}.$$

From formula (22) after passing to the limit we find

$$u(0, t) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi t}} e^{-\xi^2/2t} f(\xi)\, d\xi,$$

which is the known solution of the heat conduction equation.

The results of this section can be carried over directly to the case of the differential equation,

$$4 \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 0$$

and so on for even more independent variables.

Section 7. The general homogeneous linear equation of second order in the plane

We consider the differential equation

$$\frac{\partial^2 u}{\partial t^2} - k^2 \frac{\partial^2 u}{\partial x^2} + \alpha \frac{\partial u}{\partial t} + \beta \frac{\partial u}{\partial x} + \gamma u = 0. \tag{23}$$

The coefficients are assumed to be twice continuously differentiable with respect to x and t, while the initial values on the line $t = 0$ are assumed three times continuously differentiable with respect to x. We replace the differential equation by the difference equation

$$L(u) = u_{ti}(x, t) - k^2 u_{x\bar{x}}(x, t) + \alpha u_t + \beta u_x + \gamma u = 0 \tag{24}$$

in a grid with time mesh width h and space mesh width ch so that in a neighborhood of the appropriate part of the initial value line the inequality $1 - k^2/c^2 > \varepsilon > 0$ holds for the constant, c. The initial values are to be chosen as in Section 3.

For the proof of convergence we again transform the sum,

$$h^2 \sum_{S\alpha\beta}\sum 2 \frac{u_1 - u_3}{h} L(u)$$

by using identities (7) and (8). In addition to a sum (see for example (10)) over the doubled boundary of the triangle $S\alpha\beta$ (Fig. 6) one obtains a sum over the entire triangle $S\alpha\beta$ because of the variability of the coefficient k and the presence of lower order derivatives in the differential equation. By using the differentiability of k and the Schwarz inequality one can show that this latter sum is bounded from above by

$$Ch^2 \sum_{S\alpha\beta}\sum \left[\left(\frac{u'}{h}\right)^2 + \left(\frac{u'}{h}\right)^2 + \left(\frac{\dot{u}}{h}\right)^2 + u^2 \right]$$

where the constant C is independent of the function u,

the mesh width h, and, in a certain neighborhood of the initial line, also independent of the point S.

Again we can estimate an upper bound for $h^2 \sum_{S\alpha\beta} \sum u^2$ by[27]

$$C_1 h^2 \sum_{S} \sum_{\alpha\beta} \left(\frac{\dot{u}}{h}\right)^2 + C_2 h \sum_{\mathrm{I\,II}} u^2,$$

where the same properties hold for C_1 and C_2 as are stated above for C.

Thus we obtain an inequality of the form

$$h \sum_{S\alpha} \left[2\left(1 - \frac{k^2}{c^2}\right)\left(\frac{\dot{u}}{h}\right)^2 + \frac{k^2}{c^2}\left(\frac{u'}{h}\right)^2 \right]$$
$$+ h \sum_{S\beta} \left[2\left(1 - \frac{k^2}{c^2}\right)\left(\frac{\dot{u}}{h}\right)^2 + \frac{k^2}{c^2}\left(\frac{u^{\backprime}}{h}\right)^2 \right]$$
$$\leq C_3 h^2 \sum_{S\alpha\beta} \sum \left[\left(\frac{\dot{u}}{h}\right)^2 + \left(\frac{u'}{h}\right)^2 + \left(\frac{u^{\backprime}}{h}\right)^2 \right] + D, \quad (25)$$

where D, for all points S and mesh widths h, is a fixed bound for the sums over the initial line.

As vertices of our triangles we choose a sequence of points $S_0, S_1, \cdots, S_n = S$ lying on a line parallel to the t-axis. By summing the corresponding sequence of inequalities (25) after multiplying by h we obtain the following inequality

$$h^2 \sum_{S} \sum_{S_0\alpha} \left[2\left(1 - \frac{k^2}{c^2}\right)\left(\frac{\dot{u}}{h}\right)^2 + \frac{k^2}{c^2}\left(\frac{u'}{h}\right)^2 \right]$$
$$+ h^2 \sum_{S} \sum_{S_0\beta} \left[2\left(1 - \frac{k^2}{c^2}\right)\left(\frac{\dot{u}}{h}\right)^2 + \frac{k^2}{c^2}\left(\frac{u^{\backprime}}{h}\right)^2 \right]$$
$$\leq nh^3 C_3 \sum_{S\alpha\beta} \sum \left[\left(\frac{\dot{u}}{h}\right)^2 + \left(\frac{u'}{h}\right)^2 + \left(\frac{u^{\backprime}}{h}\right)^2 \right] + nhD. \quad (26)$$

Now if we notice that one can express a difference u' or $u^{\backprime}$ in terms of two differences $\dot{u}$ and a difference $u^{\backprime}$ or respectively u', then we see that the left-hand side of (26) is larger than the simpler form

$$C_4 h^2 \sum_{S\alpha\beta} \sum \left[\left(\frac{\dot{u}}{h}\right)^2 + \left(\frac{u'}{h}\right)^2 + \left(\frac{u^{\backprime}}{h}\right)^2 \right],$$

with a suitable constant C_4.

Then by confining the discussion to a sufficiently small neighborhood, $0 \leq t \leq nh = \delta$ of the initial line where δ is small enough so that

$$C_4 - nhC_3 = C_5 > 0,$$

we find from (26),

$$C_5 h^2 \sum_{S\alpha\beta} \sum \left[\left(\frac{\dot{u}}{h}\right)^2 + \left(\frac{u'}{h}\right)^2 + \left(\frac{u^{\backprime}}{h}\right)^2 \right] \leq \frac{C_4}{C_3} D. \quad (27)$$

[27] For proof one makes use of the inequality used in Footnote 25.

The bound given by (27) when combined with (25) gives a bound on

$$h \sum_{S\alpha} \left(\frac{u'}{h}\right)^2 + h \sum_{S\beta} \left(\frac{u^{\backprime}}{h}\right)^2,$$

from which, as in Section 3, one can prove the uniform continuity of u.

We apply the inequality (25) now, instead of to the function u itself, to its first and second difference quotients, w, which also satisfy difference equations whose second-order terms are the same as those of (24). In the rest of the terms there will appear lower order differences of u which cannot be expressed in terms w, but they will appear in the above argument in the form of quadratic double sums multiplied by h^2. This is enough to let us reach the same conclusions for the difference equation for w as we found above for u. So we can conclude from this the uniform continuity (equicontinuity) and boundedness of the function u and its first and second derivatives. Consequently a subsequence exists which converges uniformly to the solution of the initial value problem for the differential equation. Again from the uniqueness of the solution we find that the sequence of functions itself converges.

In all of this the assumption must be made that the difference quotients up to third order involving values on the two initial lines converge to continuous limit functions.[28]

Section 8. The initial value problem for an arbitrary linear hyperbolic differential equation of second order

We shall now show that the methods developed so far are adequate for solving the initial value problem for an arbitrary homogeneous linear hyperbolic differential equation of second order. It suffices to limit the discussion to the case of three variables. The development can be extended immediately to the case of more variables. It is easy to see that a transformation of variables can reduce the most general problem of this type to the following: a function $u(x, y, t)$ is to be found which satisfies the differential equation

$$u_{tt} - (au_{xx} + 2bu_{xy} + cu_{yy})$$
$$+ \alpha u_t + \beta u_x + \gamma u_y + \delta u = 0, \quad (28)$$

and which, together with its first derivative, assumes prescribed values on the surface $t = 0$. The coefficients in Eq. (28) are functions of the variables x, y, and t and satisfy the condition

$$a > 0, \quad c > 0, \quad ac - b^2 > 0.$$

We assume that the coefficients are three times differentiable with respect to x, y, and t, and that the initial

[28] This assumption and also the assumptions on the differentiability of the coefficients of the differential equation, and further on the restriction to a sufficiently small region of the initial line can be weakened in special cases.

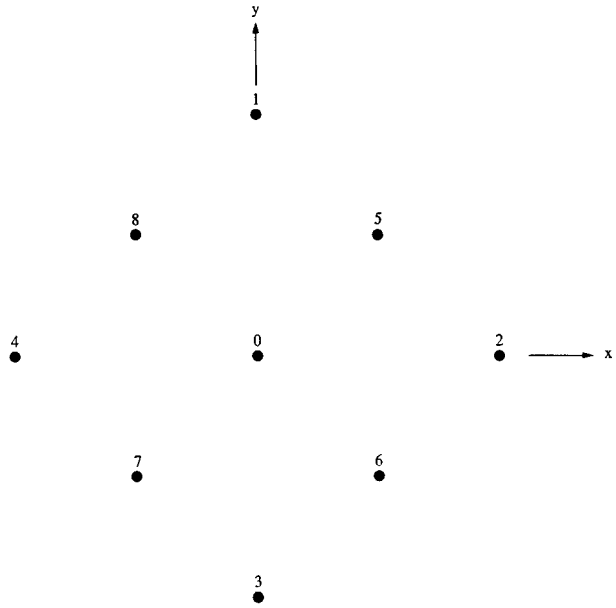


Figure 11

values u and u_t are respectively four and three times continuously differentiable with respect to x and y.

The coordinates x and y can be drawn from a given point on the initial plane in such a way that $b = 0$ there. Then of course in a certain neighborhood of this point the conditions

$$a - |b| > 0, \qquad c - |b| > 0$$

hold. We restrict our investigation to this neighborhood. We can choose a three times continuously differentiable function $d > 0$ so that

$$\left.\begin{array}{l} a - d \\ c - d \\ d - |b| \end{array}\right\} > \varepsilon > 0 \tag{29}$$

for some constant ε. Then we put the differential equation into the form

$$\begin{aligned} u_{tt} - (a - d)u_{xx} - (c - d)u_{yy} \\ - \tfrac{1}{2}(d + b)(u_{xx} + 2u_{xy} + u_{yy}) \\ - \tfrac{1}{2}(d - b)(u_{xx} - 2u_{xy} + u_{yy}) \\ + \alpha u_t + \beta u_x + \gamma u_y + \delta u = 0. \end{aligned} \tag{30}$$

We now construct in the space a grid of points, $t = lh$, $x + y = mkh$, $x - y = nkl$ $(l, m, n = \cdots -1, 0, 1, 2 \cdots)$ and we replace Eq. (30) by a difference equation $L(u) = 0$ over this mesh. We do this by assigning to each point P_0 the following neighborhood: The point $P_{\alpha'}$ or the point P_α which lies at a distance h or $-h$ respectively along the t-axis from P_0; also the points $P_1, \cdots, P_8$ which lie in the same (x, y)-plane with P_0 (see Fig. 11). These points constitute an "elementary octahedron" with vertices $P_{\alpha'}$, P_α, P_1, P_2, P_3, P_4. For each grid point lying in the interior of G we replace the derivatives appearing in Eq. (30) by difference quotients over the elementary octahedron about P_0.

We replace

$$\begin{aligned} u_{tt} &\text{ by } \frac{1}{h^2}(u_{\alpha'} - 2u_0 + u_\alpha) \\ u_{xx} &\text{ by } \frac{1}{k^2h^2}(u_2 - 2u_0 + u_4) \\ u_{yy} &\text{ by } \frac{1}{k^2h^2}(u_1 - 2u_0 + u_3) \\ u_{xx} + 2u_{xy} + u_{yy} &\text{ by } \frac{4}{k^2h^2}(u_6 - 2u_0 + u_8) \\ u_{xx} - 2u_{xy} + u_{yy} &\text{ by } \frac{4}{k^2h^2}(u_5 - 2u_0 + u_7). \end{aligned}$$

The first derivatives in (30) are replaced by the corresponding difference quotients in the elementary octahedron. The coefficients of the difference equation are given the values assumed by the coefficients of the differential equation at the point P_0.

On the initial planes $t = 0$ and $t = h$ we assume that the values of the function are assigned in such a way that as the mesh size approaches zero for fixed k, the function approaches the prescribed initial value, and the difference quotients over the two planes up through differences of fourth order converge uniformly to the prescribed derivatives.

The solution of the difference equation $L(u) = 0$ at a point is uniquely determined by the values on the two base surfaces of the pyramid of determination passing through the point.

To prove convergence we construct a sum

$$h^3 \sum \sum \sum 2\,\frac{u_{\alpha'} - u_\alpha}{h}\, L(u)$$

over all the elementary octahedrons of the pyramid of determination, and we transform it by using identities (7) and (8). This gives one space summation multiplied by h^3 and quadratic in the first difference quotients, and also over a double surface a sum which is multiplied by h^2 and contains the squares of all the difference quotients of the type $u_\alpha - u_0$, $u_\alpha - u_1$, $\cdots$, $u_\alpha - u_8$. In this expression according to (29) the coefficients are larger than some fixed positive constant in all those cases where the ratio $1/k$ between the time and space mesh sizes is taken sufficiently small.

From here on one can proceed exactly as before (Sections 7, 4) and prove that the solution of the difference equation converges to the solution of the differential equation.

(*Submitted to Math. Ann. September 1, 1927*)

[10] *Proc. Roy. Physiog. Soc. Lund.*, **20**, 250–253 (1950).
[11] *Comm. on Pure and Applied Math.*, v. 3 (1950).
[12] Courant-Hilbert, v. 2, pp. 488–490.
[13] *Acta Math.*, **41** (1918).

---

*INTEGRATION OF STIFF EQUATIONS**

BY C. F. CURTISS AND J. O. HIRSCHFELDER

THE NAVAL RESEARCH LABORATORY, DEPARTMENT OF CHEMISTRY, UNIVERSITY OF WISCONSIN, MADISON, WISCONSIN

Communicated by Farrington Daniels, December 29, 1951

In the study of chemical kinetics, electrical circuit theory, and problems of missile guidance a type of differential equation arises which is exceedingly difficult to solve by ordinary numerical procedures. A very satisfactory method of solution of these equations is obtained by making use of a forward interpolation process. This scheme has the unusual property of singling out and approximating a particular solution of the differential equation to the exclusion of the manifold of other solutions. This behavior may be explained by a simple geometrical interpretation of the significance of the forward interpolation process. The differential equations to which this method applies are called "stiff."

A typical example of a stiff equation is the equation representing the rate of formation of free radicals in a complex chemical reaction. The free radicals are created and destroyed so rapidly compared to the time scale for the over-all reaction that to a first approximation the rate of production is equal to the rate of depletion. This is the notion of the pseudo-stationary state. In some cases such as the fast reactions occurring in flames or detonations, this approximation is not sufficiently accurate. The method described in the present paper provides a means for obtaining solutions to equations of this type to any degree of accuracy.

The numerical procedure described here can easily be extended to sets of simultaneous first-order differential equations. In any particular region, the differential equations can be uncoupled by introducing suitable linear combinations of the original dependent variables. Some of the uncoupled equations may be "stiff" in which case they can be integrated by the methods discussed here; whereas other uncoupled equations may be integrated by the more usual procedures.

1. Concept of Stiff Equations.—Consider the first-order differential equation,

$$\frac{dy}{dx} = [y - G(x)]/a(x, y). \tag{1}$$

The right-hand side of this equation represents a general function of x and y which for each value of x has a root, $y = G(x)$. If Δx is the desired resolution of x or the interval which will be used in the numerical integration, the equation is "stiff" if

$$\left| \frac{a(x, y)}{\Delta x} \right| \ll 1 \tag{2}$$

and $G(x)$ is well behaved [i.e., varies with x considerably more slowly

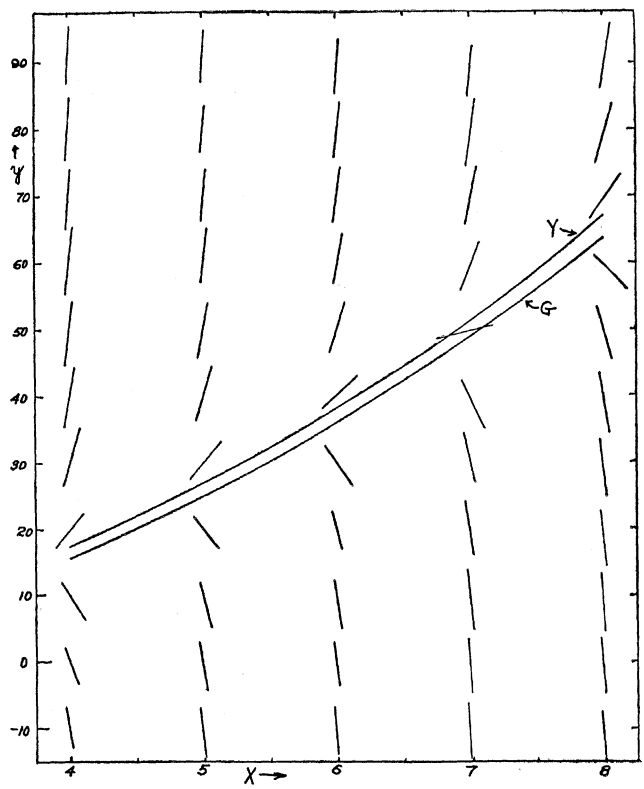


FIGURE 1

Slopes, dy/dx, for a Typical Stiff Equation, $dy/dx = 5(y - x^2)$.

than does exp $(x/a(x, G(x)))$]. The geometrical significance of the stiffness is shown in figure 1. Here the family of solutions horn out as one proceeds in the positive x direction. This can be chosen as the positive x direction without loss of generality. The slope, dy/dx, is drawn at regular intervals of x and y. For a particular value of x: for large values of y the slope is very large and positive; for small values of y the slope is very large in the

negative sense; in the vicinity of $y = G(x)$ the slope changes rapidly from large positive to large negative.

Looking at equation (1), it appears that if $a(x, y)$ is sufficiently small there is a special solution, $y = Y(x)$, which lies close to $y = G(x)$. To a first approximation, Y is given by $Y^{(1)}$,

$$Y^{(1)} = G + a(x, G)\frac{dG}{dx}. \tag{3}$$

The second approximation, $Y^{(2)}$, is obtained by substituting $Y^{(1)}$ for y in dy/dx and in $a(x, y)$,

$$Y^{(2)} = G + a(x, Y^{(1)})\frac{dY^{(1)}}{dx}. \tag{4}$$

In this manner, an arbitrary order of approximation of Y may be obtained in terms of the lower orders of approximation. If $a(x, y)$ does not depend upon y, we can write down the resulting expression for Y,

$$Y = \sum_{k=0}^{\infty} \frac{D^k G}{Dx^k} \tag{5}$$

where D/Dx is the differential operator, $a(x)d/dx$. Whereas the function $Y(x)$ remains close to $G(x)$, every other solution deviates exponentially. This can be seen by subtracting the differential equation for Y from equation (1). In the region about $Y(x)$ (assuming that $a(x, y)$ varies slowly with y),

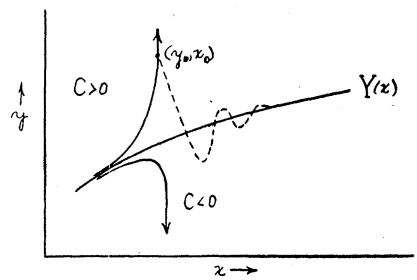


FIGURE 2

$$a(x, Y(x))\frac{d(y - Y)}{dx} = (y - Y). \tag{6}$$

This integrates to give

$$y - Y = C \exp\left[\int \frac{dx}{a(x, Y(x))}\right] \tag{7}$$

where C is a constant of integration. The qualitative appearance of this family of solutions is shown in figure 2.

2. *Numerical Solution of Stiff Equations Using Forward Interpolation.*— It is very difficult to integrate "stiff" equations by ordinary numerical methods. Small errors are rapidly magnified if the equations are inte-

grated in the direction such that the family of solutions horn out, whereas the numerical solutions oscillate violently about $Y(x)$ if the integration is carried out in the opposite direction.

We have discovered a method of numerically integrating "stiff" equations which has the desirable property that regardless of the starting conditions and regardless of the direction of integration, a solution is generated which rapidly converges to $Y(x)$. Thus for example, in figure 2, if we use this system of integration starting at the point (x_0, y_0) we obtain

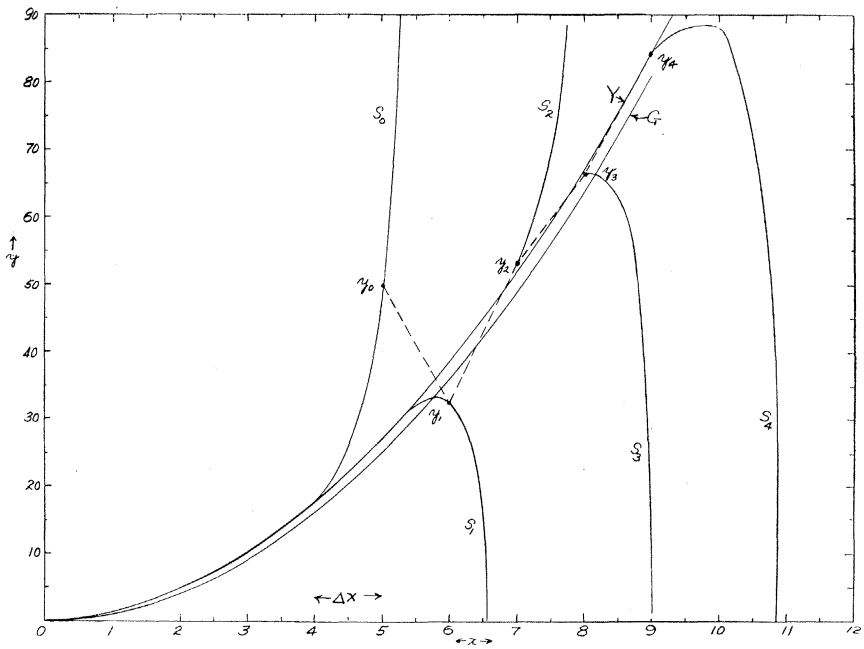


FIGURE 3

Integration of $dy/dx = 5(y - x^2)$ in the positive x direction from $(x = 5, y = 50)$. Here $a/\Delta x = 1/5$, $G(x) = x^2$, $Y(x) = 0.08 + 0.4x + x^2$.

the function indicated by the dotted line rather than the solution to the original differential equation which passes through this point and rapidly approaches infinity.

Let $x_0, x_1, \ldots, x_n, \ldots$ be a set of values of x spaced a distance Δx between successive points. Then the subscript on any quantity indicates that it is evaluated at the corresponding value of x. We wish to evaluate y_n from a knowledge of y at the previous points. Suppose, for example, that $y(x)$ can be approximated locally by a straight line passing through y_n and y_{n-1}. Then this straight line has the slope:

$$\left(\frac{dy}{dx}\right)_n = \frac{y_n - y_{n-1}}{\Delta x}. \tag{8}$$

Now, using this expression for the slope in evaluating the terms of the differential equation, equation (1), at the forward point one obtains:

$$\left(\frac{dy}{dx}\right)_n = \frac{y_n - y_{n-1}}{\Delta x} = \frac{y_n - G_n}{a(y_n, x_n)}. \tag{9}$$

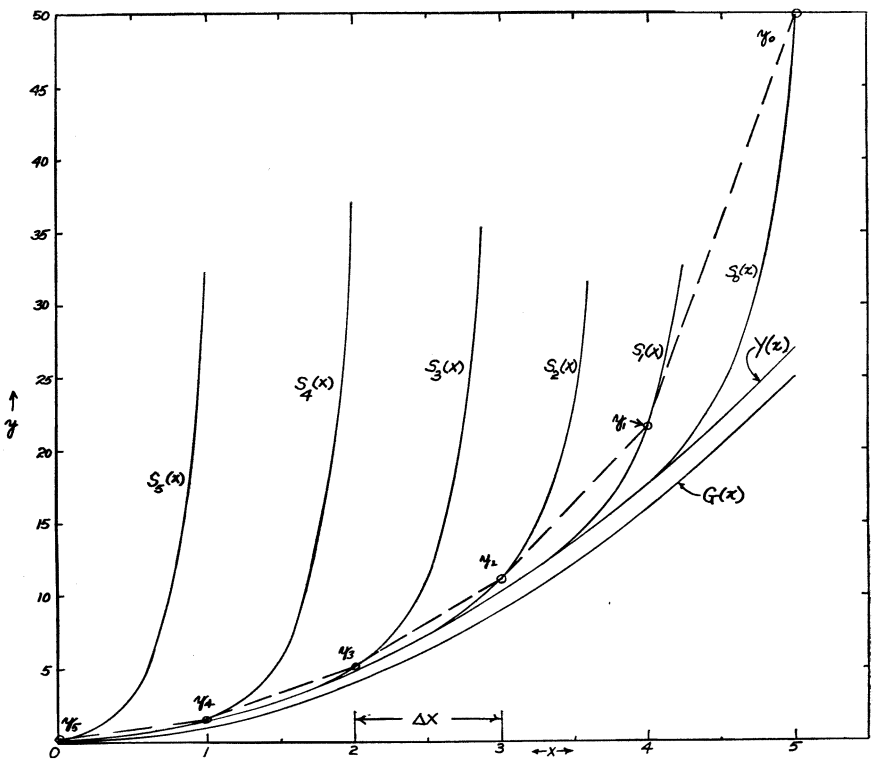


FIGURE 4

Integration of $dy/dx = 5(y - x^2)$ in the negative direction from $(x = 5, y = 50)$. Here $a/\Delta x = -1/5$, $G(x) = x^2$, $Y(x) = 0.08 + 0.4x + x^2$.

Thus,

$$y_n = \frac{G_n - \dfrac{a(x_n, y_n)}{\Delta x} y_{n-1}}{1 - \dfrac{a(x_n, y_n)}{\Delta x}}. \tag{10}$$

If $a(x, y)$ does not depend upon the value of y, equation (10) gives an explicit solution for y_n in terms of y_{n-1}; otherwise, equation (10) gives an implicit relationship between y_n and y_{n-1}. Thus starting from a point (x_0, y_0), equation (10) provides a numerical solution to the differential equation, equation (1). The curious fact is that quite regardless of the starting point, thus numerical solution seeks and approximates $Y(x)$ in the manner shown by the dotted curve in figure 2. The reason for this behavior may be seen geometrically from figure 3. Here y_0 lies on the curve $y = S_0(x)$ where $S_0(x)$ is a solution to the original differential equation. Equating $(dy/dx)_1$ from equations (8) and (9) is equivalent to asking what solution to equation (1), $y = S_1(x)$, has a tangent at the point x_1 which passes through the point (x_0, y_0). Similarly in passing from x_1 to x_2, etc. Since the slope for a stiff equation has a reasonable value (neither tremendously large in the positive or negative sense) only in the vicinity of $G(x)$ or $Y(x)$, it is clear that this numerical integration scheme limits us to approximating $Y(x)$.

It is interesting to notice that this method of integration may be used in either direction. An integration in the negative direction so that Δx is negative (i.e., in the direction of convergence of the manifold of solutions) is illustrated in figure 4. In this case the solution approaches the asymptote without oscillations. The integration in the two directions is not reversible since the definition of the forward point depends upon the direction of integration. This fact may be used numerically to hunt the special solution by first integrating in one direction and then reversing the direction.

3. *The Asymptotic Form.*—If $a(x, y)$ is a function only of x, it is easy to see how the numerical procedure leads to an approximation of Y. Since

$$G_n = Y_n - a(x_n)(dY/dx)_n, \tag{11}$$

equation (10) can be rewritten in the form

$$y_n - Y_n = -\left(\frac{\dfrac{a(x_n)}{\Delta x}}{1 - \dfrac{a(x_n)}{\Delta x}}\right)[y_{n-1} - Y_{n-1} + \epsilon] \tag{12}$$

where

$$\epsilon = -Y_n + Y_{n-1} + \Delta x\left(\frac{dY}{dx}\right)_n. \tag{13}$$

Expanding Y and (dY/dx) in Taylor series about the point, x_{n-1}, it follows that

$$\epsilon = \frac{1}{2}(\Delta x)^2 \left(\frac{d^2Y}{dx^2}\right)_{n-1} + \frac{1}{3}(\Delta x)^3 \left(\frac{d^3Y}{dx^3}\right)_{x_{n-1}} + \ldots. \tag{14}$$

Looking at equation (12) it appears that there are two reasons why y_n differs from Y_n. The rate of convergence of y toward Y is measured by the factor

$$b = \frac{\dfrac{a(x_n)}{\Delta x}}{1 - \dfrac{a(x_n)}{\Delta x}}. \tag{15}$$

The larger the interval size Δx, the smaller the value of b and the more rapidly y_n converges toward Y_n. The series converges for values of b less than unity and approaches an asymptote given approximately by

$$y = Y + a(x)\left[\frac{1}{2}(\Delta x)\frac{d^2Y}{dx^2} + \frac{1}{3}(\Delta x)^2\frac{d^2Y}{dx^3} + \ldots\right]. \tag{16}$$

This discrepancy between y and Y is due to the poorness of the fit of Y by successive linear segments. The smaller the intervals, Δx, the smaller this error.

The error in the asymptote of the series may be reduced by using a three-point formula for $(dy/dx)_n$ in which y is fit to a quadratic passing through y_n, y_{n-1} and y_{n-2}. Since the slope is evaluated at x_n, the same sort of forward interpolation is used as in the linear approximation. In place of equation (10), one obtains the following relation:

$$y_n = \frac{G_n - 2\dfrac{a(x_n, y_n)}{\Delta x}y_{n-1} + \dfrac{1}{2}\dfrac{a(x_n, y_n)}{\Delta x}y_{n-2}}{1 - \dfrac{3}{2}\dfrac{a(x_n, y_n)}{\Delta x}}. \tag{17}$$

In this case the difference between y_n and Y_n is given by[1]

$$y_n - Y_n = \left(\frac{\dfrac{a(x_n)}{\Delta x}}{1 - \dfrac{3}{2}\dfrac{a(x_n)}{\Delta x}}\right) \times \left[\begin{array}{l} -2(y_{n-1} - Y_{n-1}) + \dfrac{1}{2}(y_{n-2} - Y_{n-2}) \\ -\dfrac{1}{3}(\Delta x)^3\left(\dfrac{d^3Y}{dx^3}\right)_{n-1} - \dfrac{1}{12}(\Delta x)^4\left(\dfrac{d^4Y}{dx^4}\right)_{n-2} + \ldots \end{array}\right] \tag{18}$$

Hence the asymptotic form is approximately

$$y = Y + a(x)\left[\frac{2}{9}(\Delta x)^2\frac{d^3Y}{dx^3} + \frac{1}{18}(\Delta x)^3\frac{d^4Y}{dx^4} + \ldots\right]. \tag{19}$$

A geometrical argument of why y_n approximates Y_n follows the same line as for the linear case.

The use of a cubic to fit y leads to the expression

$$y_n = \frac{G_n + \dfrac{a(x_n, y_n)}{\Delta x}\left(-3y_{n-1} + \dfrac{3}{2}y_{n-2} - \dfrac{1}{3}y_{n-3}\right)}{1 - \dfrac{11}{6}\dfrac{a(x_n, y_n)}{\Delta x}} \tag{20}$$

with a discrepancy between y_n and Y_n given by[2]

$$y_n - Y_n = \left(\frac{\dfrac{a(x_n)}{\Delta x}}{1 - \dfrac{11}{6}\dfrac{a(x_n)}{\Delta x}}\right) \times$$

$$\left[\begin{array}{l} -3(y_{n-1} - Y_{n-1}) + \dfrac{3}{2}(y_{n-2} - Y_{n-2}) - \dfrac{1}{3}(y_{n-3} - Y_{n-3}) \\ -\dfrac{1}{4}(\Delta x)^4\left(\dfrac{d^4Y}{dx^4}\right)_{n-1} + \dfrac{1}{20}(\Delta x)^5\left(\dfrac{d^5Y}{dx^5}\right)_{n-1} + \ldots \end{array}\right]. \tag{21}$$

The use of a quartic leads to

$$y_n = \frac{G_n + \dfrac{a(x_n, y_n)}{\Delta x}\left(-4y_{n-1} + 3y_{n-2} - \dfrac{4}{3}y_{n-3} + \dfrac{1}{4}y_{n-4}\right)}{1 - \dfrac{25}{12}\dfrac{a(x_n, y_n)}{\Delta x}} \tag{22}$$

with the discrepancy between y_n and Y_n given by

$$y_n - Y_n = \left(\frac{\dfrac{a(x_n)}{\Delta x}}{1 - \dfrac{25}{12}\dfrac{a(x_n)}{\Delta x}}\right)\left[\begin{array}{l} -4(y_{n-1} - Y_{n-1}) + 3(y_{n-2} - Y_{n-2}) \\ -\dfrac{4}{3}(y_{n-2} - Y_{n-2}) + \dfrac{1}{4}(y_{n-4} - Y_{n-4}) \\ -\dfrac{1}{5}(\Delta x)^5\left(\dfrac{d^5Y}{dx^5}\right)_{n-1} + \ldots \end{array}\right]. \tag{23}$$

Thus by taking higher order polynomials to fit y at a large number of points, it is possible to obtain progressively more excellent fits of Y. In all cases, the use of forward interpolation in the evaluation of $(dy/dx)_n$ in terms of y_n, y_{n-1}, ... is the mathematical operation which forces y_n to seek out and approximate Y_n.

Acknowledgments.—The authors wish to thank Creighton Buck for suggesting the geometrical interpretation of the forward interpolation scheme; John Tukey for giving the authors the benefit of his experience with the solution of "stiff" equations; Albert Schaeffer, Kenneth Arnold and J. Barkley Rosser for their interest in this problem.

* This work was carried out under Contract NOrd-9938 with the Navy Bureau of Ordnance.

[1] Assuming the value of $a(x, y)$ to be independent of y.

---

*THE ELEMENTARY BASIC PRINCIPLES OF THE UNIFIED THEORY OF RELATIVITY**

BY VÁCLAV HLAVATÝ

DEPARTMENT OF MATHEMATICS, INDIANA UNIVERSITY

Communicated by T. Y. Thomas, January 14, 1952

1. Introduction.—The unified theory of relativity exposed recently by Einstein[1] is based on three principles: (A) The introduction of a non-symmetric tensor $g_{\lambda\mu}$ in the space X_4 of the relativity; (B) the introduction of a non-symmetric connection $\Gamma^{\nu}_{\lambda\mu}$ by means of

$$\frac{\partial}{\partial x^{\lambda}} g_{\omega\mu} = \Gamma^{\nu}_{\omega\lambda} g_{\nu\mu} + \Gamma^{\nu}_{\lambda\mu} g_{\omega\nu} \tag{1}$$

and finally (C) the introduction of a (seemingly) overdetermined system of conditions imposed on the $\Gamma^{\nu}_{\lambda\mu}$, which yields $g_{\lambda\mu}$.

In the subsequent sections we will deal with each of these basic principles. However, we confine ourselves to results only. The corresponding detailed proofs will be given in a subsequent series of three papers in the *Journal of Rational Mechanics and Analysis.*

2. Principle A.—Denote by $h_{\lambda\mu}(k_{\lambda\mu})$ the symmetric (the skew symmetric) part of $g_{\lambda\mu}$ and by g, h, k the corresponding determinants. Throughout this paper we assume $h \neq 0$. If $n = 4$ and if $h_{\lambda\mu}$ is of the signature $+++-$ then there are in general two sets of bivectors B_1, B_2 totally perpendicular, which are privileged in the sense that they are polar conjugate with respect to the cone $h_{\lambda\mu}$ as well as with respect to the linear complex $k_{\lambda\mu}$ of bivectors. Closely connected with them are four sets of (imaginary) bivectors each of them being polar self conjugate with respect to $h_{\lambda\mu}$ and $k_{\lambda\mu}$.

Projecting this configuration from an arbitrary point P of our space X_4 into the ideal space H_3 of the tangent space $T_4(P)$ of X_4 at P, one obtains a linear line complex K containing a linear line congruence C (whose axes

JOURNAL OF APPROXIMATION THEORY 6, 50–62 (1972)

On Calculating with B-Splines

CARL DE BOOR

Division of Mathematical Sciences, Purdue University, Lafayette, Indiana, 47907

Communicated by Oved Shisha

Received September 28, 1970

DEDICATED TO PROFESSOR J. L. WALSH ON THE OCCASION OF HIS 75TH BIRTHDAY

INTRODUCTION

In computational dealings with splines, the question of representation is of primary importance. For splines of fixed order on a fixed partition, this is a question of *choice of basis*, since such splines form a linear space. Only three kinds of bases for spline spaces have actually been given serious attention; those consisting of truncated power functions, of cardinal splines, and of B-splines. Truncated power bases are known to be open to severe ill-conditioning, while cardinal splines are difficult to calculate. By contrast, bases consisting of B-splines are well-conditioned, at least for orders $\leqslant 20$. Such bases are also local in the sense that at every point only a fixed number (equal to the order) of B-splines is nonzero. B-splines are also evaluated quite easily, using their definition as a divided difference of the truncated power function. Unfortunately, such calculations are ill-conditioned, particularly for partitions of widely varying interval lengths, as is indicated by the fact that special measures have to be taken in case of coincident knots.

In this note, a different way of evaluating B-splines is discussed which is very well conditioned yet efficient, and which needs no special adjustments in case of coincident knots. It is also shown that the condition of the B-spline basis increases exponentially with the order.

1. DEFINITIONS AND BASIC PROPERTIES OF (NORMALIZED) B-SPLINES

B-splines were first introduced by Schoenberg in [5, 2]. A nice compendium of many of their algebraic properties can be found in [3]. These functions are

* This work was supported in part by NSF grant GP-07163 and by the Office of Naval Research.

Copyright © 1972 by Academic Press, Inc.
All rights of reproduction in any form reserved.

also known as hump functions, patch functions or hill functions. In this section, we list a few facts about B-splines for later reference.

For simplicity, we deal with splines on a bi-infinite partition

$$\pi = \{t_i\}_{i=-\infty}^{\infty}; \qquad t_i \leqslant t_{i+1}, \qquad \text{all } i, \tag{1}$$

of the (open) subinterval $I = (\lim_{i\to-\infty} t_i, \lim_{i\to\infty} t_i)$ of the real line. Because of the localness of the B-splines, it is then a simple matter to specialize to the case of a finite partition of a finite interval (see, e.g., [3 or 1]). With k a positive integer, let

$$g_k(s; t) = (s - t)_+^{k-1} = \begin{cases} (s-t)^{k-1}, & s \geqslant t \\ 0, & s < t. \end{cases} \tag{2}$$

Then, the B-spline $M_{i,k}(t)$ is given as the k-th divided difference of $g_k(s; t)$ in s on $t_i, \ldots, t_{i+k}$ for fixed t, i.e.,

$$M_{i,k}(t) = g_k(t_i, \ldots, t_{i+k}; t), \tag{3}$$

while the *normalized B-spline* $N_{i,k}(t)$ is

$$\begin{aligned} N_{i,k}(t) &= (t_{i+k} - t_i)\, M_{i,k}(t) \\ &= g_k(t_{i+1}, \ldots, t_{i+k}; t) - g_k(t_i, \ldots, t_{i+k-1}; t). \end{aligned} \tag{4}$$

If $k > 1$ and if π is a k-extended partition [1], i.e., if at most $k - 1$ consecutive t_j's coincide, then both $M_{i,k}(t)$ and $N_{i,k}(t)$, as given by (3) and (4), respectively, are well-defined continuous functions. Otherwise, (3) and (4) make, in general, sense only for $t \neq t_j$, all j, because of the jump discontinuity of

$$(\partial/\partial s)^{k-1}\, g_k(s; t)$$

at $s = t$. Whenever this situation arises, we assume the definitions (3) and (4) to be augmented by the (admittedly arbitrary) demand that $N_{i,k}(t)$ and $M_{i,k}(t)$ be *right-continuous* everywhere. For instance, we let

$$M_{i,1}(t) = \begin{cases} (t_{i+1} - t_i)^{-1}, & t_i \leqslant t < t_{i+1} \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

hence

$$N_{i,1}(t) = \begin{cases} 1, & t_i \leqslant t < t_{i+1} \\ 0, & \text{otherwise}. \end{cases} \tag{6}$$

Note that these definitions imply

$$M_{i,1}(t) \equiv N_{i,1}(t) \equiv 0, \qquad \text{whenever } t_i = t_{i+1}. \tag{7}$$

Most of the known properties of B-splines can be derived from the simple identity

$$M_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} M_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_i} M_{i+1,k-1}(t), \tag{8}$$

which we now prove[1]. For the proof, recall Leibniz' formula

$$h(s_0, \ldots, s_k) = \sum_{r=0}^{k} f(s_0, \ldots, s_r)\, g(s_r, \ldots, s_k) \tag{9}$$

for the k-th divided difference of the function

$$h(s) = f(s)\, g(s)$$

in terms of the divided differences of $f(s)$ and $g(s)$. Apply (9) to the function

$$h(s) = g_k(s; t) = g_{k-1}(s; t)(s - t)$$

to get

$$g_k(t_i, \ldots, t_{i+k}; t) = g_{k-1}(t_i, \ldots, t_{i+k-1}; t) \cdot 1 + g_{k-1}(t_i, \ldots, t_{i+k}; t) \cdot (t_{i+k} - t),$$

since all divided differences of $(s - t)$ of order 2 or higher vanish. Hence, with (3),

$$M_{i,k}(t) = M_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_i} (M_{i+1,k-1}(t) - M_{i,k-1}(t)),$$

which is (8), slightly rewritten.

In terms of the $N_{i,k}$, (8) reads

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t). \tag{10}$$

The identity (8) states that, for any t, $M_{i,k}(t)$ is an average (or, a linear cross mean, as Aitken would call it) of the numbers $M_{i,k-1}(t)$ and $M_{i+1,k-1}(t)$. Further, for $t_i < t < t_{i+k}$, $M_{i,k}(t)$ is a strictly convex combination of these two numbers. Since $M_{i,1}(t)$ is positive for $t_i \leqslant t < t_{i+1}$ and zero otherwise, it therefore follows at once from (8) (by induction on k) that, for $k > 1$, $M_{i,k}(t)$ is positive for $t_i < t < t_{i+k}$ and zero otherwise. The normalized B-spline $N_{i,k}(t)$ satisfies, of course, the same condition.

It is this feature which makes the $N_{i,k}(t)$ so attractive for calculations. In order to evaluate the function

$$F(t) = \sum_i A_i N_{i,k}(t) \tag{11}$$

[1] This identity was also found by Lois Mansfield.

at a point $\hat{t} \in [t_j, t_{j+1})$, it is merely necessary to calculate the k numbers

$$N_{i,k}(\hat{t}), \qquad i = j - k + 1, \ldots, j;$$

$F(\hat{t})$ is then given by

$$F(\hat{t}) = \sum_{i=j-k+1}^{j} A_i N_{i,k}(\hat{t}).$$

Differentiation of $F(t)$ is equally simple. One has

$$\begin{aligned} N_{i,k}^{(1)}(t) &= (d/dt)[g_k(t_{i+1}, \ldots, t_{i+k}; t) - g_k(t_i, \ldots, t_{i+k-1}; t)] \\ &= -(k-1)[M_{i+1,k-1}(t) - M_{i,k-1}(t)]. \end{aligned}$$

Hence,

$$\begin{aligned} F^{(1)}(t) &= (k-1) \sum_i A_i [M_{i,k-1}(t) - M_{i+1,k-1}(t)] \\ &= (k-1) \sum_i A_i^{(1)} N_{i,k-1}(t), \end{aligned} \tag{12}$$

where

$$A_i^{(1)} = (A_i - A_{i-1})/(t_{i+k-1} - t_i). \tag{13}$$

More generally, with

$$\begin{aligned} A_i^{(0)} &= A_i, \\ A_i^{(j)} &= (A_i^{(j-1)} - A_{i-1}^{(j-1)})/(t_{i+k-j} - t_i), \qquad j > 0, \end{aligned} \tag{14}$$

one has

$$F^{(j)}(t) = (k-1) \cdots (k-j) \sum_i A_i^{(j)} N_{i,k-j}(t). \tag{15}$$

If π is uniform,

$$t_i = t_0 + ih, \qquad \text{all} \quad i,$$

then (15) reduces to

$$F^{(j)}(t) = h^{-j} \sum_i (\nabla^j A_i)\, N_{i,k-j}(t)$$

which is familiar from [5]. We return to the evaluation of the spline function $F(t)$ and its derivatives in Section 2.

Using the identity (8), it is possible to rewrite $F(t)$ in terms of normalized B-splines of lower order, with certain polynomial coefficients. One has

$$\begin{aligned} F(t) &= \sum_i A_i N_{i,k}(t) \\ &= \sum_i A_i\{(t - t_i)\, M_{i,k-1}(t) + (t_{i+k} - t)\, M_{i+1,k-1}(t)\} \\ &= \sum_i \{A_i(t - t_i) + A_{i-1}(t_{i+k-1} - t)\}\, M_{i,k-1}(t) \\ &= \sum_i A_i^{[1]}(t)\, N_{i,k-1}(t), \end{aligned}$$

where

$$A_i^{[1]}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} A_i + \frac{t_{i+k-1} - t}{t_{i+k-1} - t_i} A_{i-1}\,.$$

More generally, setting

$$A_i^{[j]}(t) = \begin{cases} A_i\,, & j = 0 \\ \dfrac{t - t_i}{t_{i+k-j} - t_i} A_i^{[j-1]}(t) + \dfrac{t_{i+k-j} - t}{t_{i+k-j} - t_i} A_{i-1}^{[j-1]}(t), & j > 0, \end{cases} \tag{16}$$

one gets

$$F(t) = \sum_i A_i^{[j]}(t)\, N_{i,k-j}(t). \tag{17}$$

Since $N_{i,1}(t) = 1$ for $t_i \leqslant t < t_{i+1}$ and is zero otherwise, it follows that

$$F(t) = A_i^{[k-1]}(t), \qquad t_i \leqslant t < t_{i+1}\,. \tag{18}$$

Hence, if $t \in [t_i\,, t_{i+1})$, then $F(t)$ can also be found, from $A_{i-k+1}\,,\ldots, A_i$, by forming repeatedly certain convex combinations according to (16).

Finally, we mention the important identity

$$(s - t)^{k-1} = \sum_i \varphi_{i,k}(s)\, N_{i,k}(t), \quad \text{with} \quad \varphi_{i,k}(s) = \prod_{r=1}^{k-1} (s - t_{i+r}), \quad \text{all } i, \tag{19}$$

which was first proved by Marsden [4], and which simplifies many dealings with splines. Its proof is straightforward : Setting

$$A_i^{[0]}(t) = A_i = \varphi_{i,k}(s), \qquad \text{all } i,$$

one gets from (16) that

$$\begin{aligned} A_i^{[1]}(t) &= \{(t - t_i)\, \varphi_{i,k}(s) + (t_{i+k-1} - t)\, \varphi_{i-1,k}(s)\}/(t_{i+k-1} - t_i) \\ &= \varphi_{i,k-1}(s)\{(t - t_i)(s - t_{i+k-1}) + (t_{i+k-1} - t)(s - t_i)\}/(t_{i+k-1} - t_i) \\ &= \varphi_{i,k-1}(s)(s - t); \end{aligned}$$

hence

$$\sum_i \varphi_{i,k}(s)\, N_{i,k}(t) = (s - t) \sum_i \varphi_{i,k-1}(s)\, N_{i,k-1}(t).$$

Since

$$\sum_i \varphi_{i,1}(s)\, N_{i,1}(t) \equiv \sum_i N_{i,1}(t) \equiv 1,$$

induction on k now proves (19).

By expanding both sides of (19) in powers of s and comparing coefficients of like powers, it follows, e.g., that

$$\sum_i N_{i,k}(t) \equiv 1, \tag{20}$$

reaffirming the conclusion from (16) and (18) that, for $t \in [t_i, t_{i+1})$, the number $F(t)$ is a *convex* combination of the numbers $A_{i-k+1}, \ldots, A_i$.

It also follows from (19) that

$$(s - t)^{k-1} = \sum_i \psi_{i,k}(t)\, N_{i,k}(s), \qquad \psi_{i,k}(t) = (t_{i+1} - t) \cdots (t_{i+k-1} - t); \tag{21}$$

hence

$$(s - t_j)_+^{k-1} = \sum_i \psi_{i,k}^+(t_j)\, N_{i,k}(s), \qquad \psi_{i,k}^+(t) = (t_{i+1} - t)_+ \cdots (t_{i+k-1} - t)_+ . \tag{22}$$

2. Evaluation of B-splines

The material in the preceding section suggests (at least) two stable, yet efficient ways to evaluate the function

$$F(t) = \sum_i A_i N_{i,k}(t)$$

at any particular t, which we now discuss. The resulting algorithms can, of

course, also be used to evaluate the single B-spline $N_{i,k}(t)$, merely by specializing to the situation

$$A_j = \delta_{ij}, \qquad \text{all } j.$$

The more obvious of the two algorithms is based on (16)-(18). Having found i such that $t_i \leqslant t < t_{i+1}$, one generates all the entries in the following table, using (16):

$$\begin{array}{lllll}
A_{i-k+1}^{[0]}(t) & & & & \\
A_{i-k+2}^{[0]}(t) & A_{i-k+2}^{[1]} & & & \\
\vdots & \vdots & \ddots & & \\
A_{i-1}^{[0]}(t) & A_{i-1}^{[1]}(t) & \cdots & A_{i-1}^{[k-2]}(t) & \\
A_{i}^{[0]}(t) & A_{i}^{[1]}(t) & \cdots & A_{i}^{[k-2]}(t) & A_{i}^{[k-1]}(t)
\end{array}$$

The right-most entry is then the desired number $F(t)$.

Set

$$\begin{aligned} A(r, s) &= A_{i-k+r}^{[s-1]}(t), \qquad r = s,\ldots, k; \qquad s = 1,\ldots, k, \\ \left.\begin{aligned} DP(r) &= t_{i+r} - t, \\ DM(r) &= t - t_{i-k+r}, \end{aligned}\right\} & \; r = 1,\ldots, k, \end{aligned} \tag{23}$$

to simplify notation. Then

$$\begin{aligned} A(r, 1) &= A_{i-k+r}, \qquad r = 1,\ldots, k, \\ A(r, s+1) &= (DM(r) * A(r, s) + DP(r-s) * A(r-1, s))/(DM(r) + DP(r-s)) \\ &\qquad r = s+1,\ldots, k; \qquad s = 1,\ldots, k-1. \end{aligned} \tag{24}$$

Note that

$$\begin{aligned} DM(r) + DP(r-s) &= t - t_{i-k+r} + t_{i+r-s} - t \\ &= t_{i+r-s} - t_{i+r-k} \geqslant t_{i+1} - t_i > 0, \end{aligned}$$

so that coincident points in the partition π cause no additional difficulty if, as we assume, i is chosen so that $t_i \leqslant t < t_{i+1}$.

The calculation of the $A(r, s)$ can either be carried out column by column, i.e., with

$$r = s,\ldots, k; \qquad s = 2,\ldots, k$$

or row by row, i.e., with

$$s = 2,\ldots, r; \qquad r = 2,\ldots, k$$

or downward diagonal by downward diagonal, i.e., with

$$s = 2,\ldots, j, \qquad r = s + k - j; \qquad j = 2,\ldots, k.$$

Each way requires only one one-dimensional array with k entries for the actual storage of the successively calculated numbers $A(r, s)$. In the first way, one would precalculate the array DP, in the last, one would precalculate the array DM, while the second would require initial calculation of both the DP and the DM array.

If the value of $F(t)$ *and* of some of its derivatives are required at the same time, then it is probably better to use an algorithm which generates at the same time all the numbers $N_{i,j}(t)$ which are not zero for the given t. With the assumption that

$$t_i \leqslant t < t_{i+1},$$

this amounts to generating all the entries of the following triangular table:

$$\begin{array}{ccccc}
N_{i,1}(t) & N_{i-1,2}(t) & \cdots & N_{i-k+2,k-1}(t) & N_{i-k+1,k}(t) \\
 & N_{i,2}(t) & \cdots & N_{i-k+3,k-1}(t) & N_{i-k+2,k}(t) \\
 & & \ddots & \vdots & \vdots \\
 & & & N_{i,k-1}(t) & N_{i-1,k}(t) \\
 & & & & N_{i,k}(t)
\end{array}$$

The $(k - j)$th column of this table contains the numbers needed for the evaluation of $F^{(j)}(t)$ using (15), $j = 0,\ldots, k - 1$. For this reason, we describe here only how to generate this table column by column.

Set

$$\begin{aligned} N(r, s) &= N_{i+r-s,s}(t), \\ \left.\begin{aligned} DP(r) &= t_{i+r} - t, \\ DM(r) &= t - t_{i+1-r}, \end{aligned}\right\} & \; r = 1,\ldots, k, \end{aligned} \tag{25}$$

to simplify notation. The needed table entries are then

$$N(r, s), \qquad r = 1,\ldots, s; \qquad s = 1,\ldots, k,$$

while

$$N(r, s) = 0, \qquad \text{for} \quad r > s \qquad \text{or} \quad r < 1. \tag{26}$$

With the abbreviations (25), we get from (10) that

$$N(r, s+1) = DM(s+1-r+1)\frac{N(r-1, s)}{DP(r-1) + DM(s+1-r+1)} + DP(r)\frac{N(r, s)}{DP(r) + DM(s+1-r)}. \tag{27}$$

We emphasize that, once again, this formula is unaffected by the possible presence of coincident t_j's, since

$$DP(r) + DM(s+1-r) = t_{i+r} - t_{i+r-s} \geqslant t_{i+1} - t_i > 0$$

for all values of r and s of interest.

Equations (25) and (27) lead to the following algorithm for the generation of the $N(r, s)$:

```
Set N(1, 1) = 1;
for s = 1,..., k − 1, do:
⋮       set DP(s) = t_{i+s} − t, DM(s) = t − t_{i+1−s},
⋮       set N(1, s + 1) = 0;
⋮       for r = 1,..., s, do:
⋮         ⋮     set M = N(r, s)/(DP(r) + DM(s + 1 − r)),
⋮         ⋮     set N(r, s + 1) = N(r, s + 1) + DP(r)* M,
⋮.....⋮...set N(r + 1, s + 1) = DM(s + 1 − r) * M.
```

This algorithm can, of course, be modified so as to use only a one-dimensional array of k entries for the storage of the $N(r, s)$, by overwriting successive columns.

3. Condition of the B-spline basis

A limited number of numerical experiments have shown both algorithms presented in the preceding section to be extremely stable when used for the evaluation of $F(t)$. This is not suprising, since both algorithms arrive at $F(t)$ by repeatedly forming convex combinations. Even for $k = 80$, the absolute error in the computed value for $F(t)$ was only about the size of roundoff in the coefficients A_i.

These numerical experiments showed, incidentally, the unpleasant but important fact that the condition number of the normalized B-spline basis increases exponentially with the order k. This can be confirmed, in the case of a uniform partition, by the following calculations.

The condition number $\varkappa(k, \pi)$ of the normalized B-spline basis $\{N_{i,k}\}_i$ for the partition $\pi = \{t_i\}$ is defined by

$$\varkappa(k, \pi) = \sup_{\|A\|_\infty=1} \left\| \sum_i A_i N_{i,k} \right\|_\infty \Big/ \inf_{\|A\|_\infty=1} \left\| \sum_i A_i N_{i,k} \right\|_\infty, \tag{28}$$

where

$$\| A \|_\infty = \sup_i | A_i |$$

and

$$\|f\|_\infty = \sup_{t \in I} |f(t)|,$$

I being the interval for which π is a partition. It is assumed, of course, that $\{N_{i,k}\}$ is linearly independent, i.e., that no t_i agrees with more than $k-1$ other t_j's. By (20),

$$\left\| \sum_i A_i N_{i,k} \right\|_\infty \leqslant \| A \|_\infty$$

with equality when $A_i = 1$, all i. Hence

$$\varkappa(k, \pi) = 1 \Big/ \inf_{\|A\|_\infty=1} \left\| \sum_i A_i N_{i,k} \right\|_\infty . \tag{29}$$

It was proved in [1] that, for each k,

$$D_k = \sup_\pi \varkappa(k, \pi) < \infty$$

Preliminary calculations based on the argument in [1] give upper bounds for D_k which increase about as fast as $k!$, as k increases. These bounds are probably not sharp. But it can be shown that D_k must increase at least exponentially with k.

THEOREM. *If the partition π is uniform,*

$$t_j = t_0 + jh, \qquad \textit{all} \quad j,$$

then

$$\varkappa(k, \pi) = 1/\varphi_k(\pi), \tag{30}$$

where

$$\varphi_k(u) = \sum_j \psi_k(u + 2\pi j),$$

$$\psi_k(u) = \left(\frac{\sin(u/2)}{u/2}\right)^k.$$

Hence,

$$\begin{aligned} \varkappa(k, \pi) &\geqslant (\pi/2)^{k-2}, \qquad \text{for} \quad k > 1, \\ \lim_{k\to\infty} \varkappa(k, \pi)/(\pi/2)^k &= 2. \end{aligned} \tag{31}$$

Proof. We show how to obtain (30) from Schoenberg's recent paper [6] on Cardinal Interpolation. First, we note that $\varkappa(k, \pi)$ is invariant under a linear change of the independent variable; hence we may restrict attention to the particular uniform partition

$$\pi = \{j\}.$$

If $A = (A_i)$ is any sequence, then

$$\left\| \sum_i A_i N_{i,k} \right\|_\infty \geqslant \| C_A \|_\infty , \tag{32}$$

with

$$C_A(i) = \sum_j A_j N_{j,k}(i + k/2), \qquad \text{all } i.$$

Since we are dealing with the particular partition $\pi = \{j\}$, we have

$$N_{j,k}(t) = N_{0,k}(t - j) = M_k(t - j - k/2),$$

where, in the notation of [6 or 5],

$$M_k(t) = \frac{1}{(k-1)!}\, \delta^k t_+^{k-1}.$$

Therefore,

$$C_A(i) = \sum_j A_j M_k(i - j), \qquad \text{all } i.$$

In this form, the linear sequence-to-sequence transformation

$$A \to C_A$$

has been studied in detail in [6], where it is proved (see, in particular Section 6 of [6]) that

$$\| C_A \|_\infty \geqslant \varphi_k(\pi) \| A \|_\infty , \quad \text{with equality if} \quad A_i = -A_{i+1} , \quad \text{all} \quad i. \tag{33}$$

Combining (32) and (33), we get that

$$\inf_{\|A\|_\infty = 1} \left\| \sum_j A_j N_{j,k} \right\|_\infty \geqslant \varphi_k(\pi). \tag{34}$$

On the other hand, if

$$A_i = -A_{i+1} , \quad \text{all} \quad i,$$

then it easily follows (e.g., from Lemma 12 of [6]) that

$$\left\| \sum_j A_j N_{j,k} \right\|_\infty = \| C_A \|_\infty ,$$

hence, by (33), then

$$\left\| \sum_j A_j N_{j,k} \right\|_\infty = \varphi_k(\pi) \| A \|_\infty .$$

Combining this with (34) and (29) gives (30); Q.E.D.

We note that the inequality (31) alone can be derived directly by calculation of $\Sigma_j (-1)^j N_{j,k}(k/2)$. Further, the theorem implies that, for a uniform partition,

$$\varkappa(k, \pi) \approx 10^{k/5}. \tag{35}$$

This implies that, on a typical 7 decimal digit machine and with $k = 40$, the calculated value of $F(t) = \Sigma_j A_j N_{j,k}(t)$ at some point $\hat{t}$ may well be inaccurate in the first nonzero digit. Since, on the other hand, the normalized B-spline basis is, at present, the only suitable basis for dealing with splines in computations, this seems to limit the use of splines in solving functional equations on a computer to splines of relatively low order, say of order $k < 20$, unless one is willing to pay the price of multiple-precision arithmetic.

References

1. C. de Boor, On uniform approximation by splines, *J. Approximation Theory* **1** (1968), 219–235.
2. H. B. Curry and I. J. Schoenberg, On spline distributions and their limits: the Polya distributions, *Abstr. Bull. Amer. Math. Soc.* **53** (1947), 1114.

3. H. B. Curry and I. J. Schoenberg, On Polya frequency functions IV: The fundamental spline functions and their limits, *J. Anal. Math.* **17** (1966), 71–107.
4. M. Marsden, An identity for spline functions and its application to variation diminishing spline approximations, *J. Approximation Theory* **3** (1970), 7–49.
5. I. J. Schoenberg, Contributions to the problem of approximation of equidistant data by analytic functions, *Quart. Appl. Math.* **4** (1946), 45–99; 112–141.
6. I. J. Schoenberg, Cardinal interpolation and spline functions. *J. Approximation Theory* **2** (1969), 167–206.

Added in proof: The following additional references are of interest.

C. de Boor, "Subroutine package for calculating with *B*-splines," Los Alamos Scient. Lab. Report LA-4728-MS, Aug. 1971.

M. G. Cox, "The numerical evaluation of *B*-splines," National Physical Laboratory Report DNAC 4, Aug. 1971.

T. Segethova, "Numerical construction of hill functions," Univ. Maryland Computer Science Center Technical Rep. 70-110, NGL-21-002-008, April 1970.

J. SIAM NUMER. ANAL.
Ser. B, Vol. 2, No. 2
Printed in U.S.A., 1965

CALCULATING THE SINGULAR VALUES AND PSEUDO-INVERSE OF A MATRIX*

G. GOLUB† AND W. KAHAN‡

Abstract. A numerically stable and fairly fast scheme is described to compute the unitary matrices U and V which transform a given matrix A into a diagonal form $\Sigma = U^*AV$, thus exhibiting A's singular values on Σ's diagonal. The scheme first transforms A to a bidiagonal matrix J, then diagonalizes J. The scheme described here is complicated but does not suffer from the computational difficulties which occasionally afflict some previously known methods. Some applications are mentioned, in particular the use of the pseudo-inverse $A^I = V\Sigma^I U^*$ to solve least squares problems in a way which dampens spurious oscillation and cancellation.

1. Introduction. This paper is concerned with a numerically stable and fairly fast method for obtaining the following decomposition of a given rectangular matrix A:

$$A = U\Sigma V^*, \tag{1.1}$$

where U and V are unitary matrices and Σ is a rectangular diagonal matrix of the same size as A with nonnegative real diagonal entries. These diagonal elements are called the *singular values* or *principal values* of A; they are the nonnegative square roots of the eigenvalues of A^*A or AA^*.

Some applications of the decomposition (1.1) will be mentioned in this paper. In particular, the pseudo-inverse A^I of A will be represented in the form

$$A^I = V\Sigma^I U^*, \tag{1.2}$$

where Σ^I is obtained from Σ by replacing each positive diagonal entry by its reciprocal. The properties and applications of A^I are described in papers by Greville [15], Penrose [25], [26], and Ben-Israel and Charnes [3]. The pseudo-inverse's main value, both conceptually and practically, is that it provides a solution for the following least-squares problem.

Of all the vectors $\mathbf{x}$ *which minimize the sum of squares* $\| \mathbf{b} - A\mathbf{x} \|^2$, *which is the shortest* (*has the smallest* $\| \mathbf{x} \|^2 = \mathbf{x}^*\mathbf{x}$)?

The solution is $\mathbf{x} = A^I\mathbf{b}$. If there were only one vector $\mathbf{x}$ which minimized

* Received by the editors July 14, 1964. This paper was first presented at the Symposium on Matrix Computations at Gatlinburg, Tennessee, in April, 1964.

† Computation Center, Stanford University, Stanford, California. The work of the first author was in part supported by Contract Nonr 225(37) at Stanford University and by The Boeing Scientific Research Laboratories, Seattle, Washington.

‡ University of Toronto, Toronto, Ontario. The second author wishes to thank the National Research Council (of Canada) for their support of the Institute of Computer Sciences at the University of Toronto.

$\| \mathbf{b} - A\mathbf{x} \|$ we would save a bit of work by using

$$A^I = (A^*A)^{-1}A^*$$

instead of (1.2), and this is what we often try to do. But if A^*A is (nearly) singular then there will be infinitely many vectors $\mathbf{x}$ which (nearly) minimize $\| \mathbf{b} - A\mathbf{x} \|$ and the last formula will have to be modified in a way which takes A's rank into account (cf. [4], [6], [7]). The methods considered in this paper simplify the problem of assigning a rank to A.

In the past the conventional way to determine the rank of A was to convert A to a row-echelon form, e.g.,

$$\begin{pmatrix} x & x & x & x & x & \cdot & \cdot & \cdot \\ 0 & x & x & x & x & \cdot & \cdot & \cdot \\ 0 & 0 & x & x & x & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot \end{pmatrix}, \qquad (\text{rank} = 3),$$

in which x's represent nonzero elements and 0's represent zeros. The transformation was accomplished by premultiplying A by a succession either of elementary matrices (cf. [5]) or of unitary matrices (cf. [17]) designed to liquidate the subdiagonal elements of each column in turn. In order to obtain a simple picture like the one above it would have been necessary to perform column-interchanges to ensure that the largest possible numbers were being left on the diagonal (cf. "complete pivoting" as described by Wilkinson [33]). It is certainly possible to arrange that in the row-echelon form of A each row will have its largest element on the diagonal. Consequently the rank of A is just the number r of consecutive nonzero terms on the diagonal of its row-echelon form; all rows after the rth are zero. And Σ, correspondingly, should have just r nonzero singular values on its diagonal.

But in floating-point calculations it may not be so easy to decide whether some number is effectively zero or not. Rather, one will try to determine the rank r by observing whether all rows after the rth are negligible in comparison to the first r, with the expectation that the same will be true of the singular values. Even this criterion is hard to apply, as the following example shows:

$$\begin{pmatrix} 1 & -1 & -1 & -1 & -1 & -1 & \cdot & \cdot & \cdot \\ & 1 & -1 & -1 & -1 & -1 & \cdot & \cdot & \cdot \\ & & 1 & -1 & -1 & -1 & \cdot & \cdot & \cdot \\ & & & 1 & -1 & -1 & \cdot & \cdot & \cdot \\ & & & & 1 & -1 & \cdot & \cdot & \cdot \\ & & \mathbf{0} & & & 1 & \cdot & \cdot & \cdot \\ & & & & & & \cdot & \cdot & \cdot \end{pmatrix}.$$

If this matrix, already in row-echelon form, has a sufficiently large number

of rows and columns, then, although it may not appear to the naked eye to be deficient in rank, it is violently ill-conditioned (it has a very tiny singular value), as can be seen by applying the matrix to the column vector whose elements are, in turn,

$$1, 2^{-1}, 2^{-2}, 2^{-3}, \cdots, 2^{-n}, \cdots.$$

On the other hand, when all the -1's in the matrix are replaced by $+1$'s then the resulting matrix is quite docile. Therefore, it would be very hard to tell, by looking at only the diagonal elements of the row-echelon form, whether or not the original matrix A had a singular value sufficiently small to be deleted during the calculation of A^I. In other words, without looking explicitly at the singular values there seems to be no satisfactory way to assign a rank to A.

The singular values of a matrix A are the nonnegative square roots of the eigenvalues of A^*A or AA^*, whichever has fewer rows and columns (see [1]). But the calculation of A^*A using ordinary floating point arithmetic does serious violence to the smaller singular values as well as to the corresponding eigenvectors which appear in U and V in (1.1). A discussion of these points can be found in a paper by Osborne [24], which also contains a nice proof of the existence of the decomposition (1.1). Since the columns of U are the eigenvectors of AA^* and the columns of V are the eigenvectors of A^*A, there is some possibility that a simple calculation of the decomposition (1.1) could be accomplished by using double-precision arithmetic to deal with A^*A and AA^* directly in some way. Such a scheme would be convenient with a machine like the IBM 7094 which has double-precision hardware. But for most other machines, and especially when a programming language deficient in double-precision facilities is used, the complicated scheme described in this paper seems to be the best we have.

Kogbetliantz [18], Hestenes [16], and Forsythe and Henrici [9] have proposed rotational or Jacobi-type methods for obtaining the decomposition (1.1). Kublanovskaja [19] has suggested a QR-type method. These methods are accurate but are slow in terms of total number of operations.

Our scheme is based upon an idea exploited by Lanczos [20]; the matrix

$$\tilde{A} = \begin{pmatrix} 0 & A \\ A^* & 0 \end{pmatrix}$$

has for its eigenvalues the singular values of A, each appearing with both a positive and a negative sign. The representation $\tilde{A}$ could not be treated directly by a standard eigenvalue-vector program without dealing with the problems which we shall discuss in detail in what follows.

2. A matrix decomposition. In order to facilitate the computation of the singular values and the pseudo-inverse of the complex $m \times n$ matrix A, we

describe a convenient matrix decomposition. We assume throughout our discussion that $m \geqq n$ without any loss of generality.

THEOREM 1. *Let A be any $m \times n$ matrix with complex elements. Then A can be decomposed as*

$$A = PJQ^*,$$

where P and Q are unitary matrices and J is an $m \times n$ bidiagonal matrix of the form

$$J = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \cdot & \cdot & \cdot & 0 \\ & \alpha_2 & \beta_2 & 0 & \cdot & \cdot & \cdot \\ & \mathbf{0} & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \\ & & & & & \cdot & \beta_{n-1} \\ & & & & & & \alpha_n \\ & & \mathbf{0} & & & & \end{pmatrix} \left.\vphantom{\begin{matrix}a\\a\\a\\a\\a\\a\\a\end{matrix}}\right\} (m-n) \times n.$$

Proof. The proof will be a constructive one in which Householder transformations (see [17], [21], [32]) are used. Let $A = A^{(1)}$ and let $A^{(3/2)}$, $A^{(2)}, \cdots, A^{(n)}, A^{(n+1/2)}$ be defined as follows:

$$A^{(k+1/2)} = P^{(k)}A^{(k)}, \qquad k = 1, 2, \cdots, n,$$

$$A^{(k+1)} = A^{(k+1/2)}Q^{(k)}, \qquad k = 1, 2, \cdots, n-1.$$

$P^{(k)}$ and $Q^{(k)}$ are hermitian, unitary matrices of the form

$$P^{(k)} = I - 2\mathbf{x}^{(k)}\mathbf{x}^{(k)*}, \qquad \mathbf{x}^{(k)*}\mathbf{x}^{(k)} = 1,$$

$$Q^{(k)} = I - 2\mathbf{y}^{(k)}\mathbf{y}^{(k)*}, \qquad \mathbf{y}^{(k)*}\mathbf{y}^{(k)} = 1.$$

The unitary transformation $P^{(k)}$ is determined so that

$$a_{i,k}^{(k+1/2)} = 0, \qquad i = k+1, \cdots, m,$$

and $Q^{(k)}$ is determined so that

$$a_{k,j}^{(k+1)} = 0, \qquad j = k+2, \cdots, n,$$

and $A^{(k+1)}$ has the form

$$A^{(k+1)} = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \cdot & \cdot & & & & & \\ 0 & \alpha_2 & \beta_2 & 0 & \cdot & & & \mathbf{0} & & \\ \cdot & 0 & \cdot & \cdot & \cdot & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & & & & & \\ & & & & \alpha_k & \beta_k & & & & \\ & & & & & x & x & \cdot & \cdot & \cdot \\ & & \mathbf{0} & & & x & x & \cdot & \cdot & \cdot \\ & & & & & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & & & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

We illustrate the derivation of the formula for $P^{(k)}$. In order not to disturb those elements which have already been annihilated we set

$$x_i^{(k)} = 0, \qquad i = 1, 2, \cdots, k-1.$$

Since $P^{(k)}$ is a unitary transformation, length is preserved and consequently

$$|\alpha_k|^2 = \sum_{i=k}^{m} |a_{i,k}^{(k)}|^2. \tag{2.1}$$

Also, since $P^{(k)}$ is hermitian,

$$P^{(k)}A^{(k+1/2)} = A^{(k)},$$

so that

$$(1 - 2|x_k^{(k)}|^2)\alpha_k = a_{k,k}^{(k)},$$
$$-2x_i^{(k)}\bar{x}_k^{(k)}\alpha_k = a_{i,k}^{(k)}, \qquad i = k+1, \cdots, m,$$

and hence

$$|x_k^{(k)}|^2 = \frac{1}{2}\left(1 - \frac{a_{k,k}^{(k)}}{\alpha_k}\right), \tag{2.2}$$

$$x_i^{(k)} = \frac{-a_{i,k}^{(k)}}{2\alpha_k\bar{x}_k^{(k)}}. \tag{2.3}$$

Equations (2.1), (2.2), and (2.3) define two possible vectors $\mathbf{x}^{(k)}$ to within scalar factors of modulus one. In the interest of numerical stability, let us choose sgn α_k so that $x_k^{(k)}$ is as large as possible. Thus

$$\alpha_k = -\frac{a_{k,k}^{(k)}}{|a_{k,k}^{(k)}|}\left(\sum_{i=k}^{m} |a_{i,k}^{(k)}|^2\right)^{1/2}.$$

Summarizing, we have

$$A^{(k+1/2)} = A^{(k)} - \mathbf{x}^{(k)}\cdot 2(\mathbf{x}^{(k)*}A^{(k)}),$$

with

$$s_k = \left(\sum_{i=k}^{m} |a_{i,k}^{(k)}|^2\right)^{1/2},$$

$$\alpha_k = -s_k\left(\frac{a_{k,k}^{(k)}}{|a_{k,k}^{(k)}|}\right),$$

$$x_i^{(k)} = 0 \quad \text{for} \quad i < k,$$

$$x_k^{(k)} = \left[\frac{1}{2}\left(1 + \frac{|a_{k,k}^{(k)}|}{s_k}\right)\right]^{1/2}, \quad \text{(say)},$$

$$c_k = \left(2s_k\frac{a_{k,k}^{(k)}}{|a_{k,k}^{(k)}|}x_k^{(k)}\right)^{-1},$$

and

$$x_i^{(k)} = c_k a_{i,k}^{(k)} \quad \text{for} \quad i > k.$$

(If $s_k = 0$, just set $\alpha_k = 0$ and $\mathbf{x}^{(k)} = \mathbf{0}$.) Similarly,

$$A^{(k+1)} = A^{(k+1/2)} - 2(A^{(k+1/2)}\mathbf{y}^{(k)}) \cdot \mathbf{y}^{(k)*},$$

with

$$t_k = \left(\sum_{j=k+1}^{n} | a_{k,j}^{(k+1/2)} |^2 \right)^{1/2},$$

$$\beta_k = -t_k \cdot \frac{a_{k,k+1}^{(k+1/2)}}{| a_{k,k+1}^{(k+1/2)} |},$$

$$y_j^{(k)} = 0 \quad \text{for} \quad j \leqq k,$$

$$y_{k+1}^{(k)} = \left[\frac{1}{2} \left(1 + \frac{| a_{k,k+1}^{(k+1/2)} |}{t_k} \right) \right]^{1/2}, \quad \text{(say)},$$

$$d_k = \left(2t_k \frac{a_{k,k+1}^{(k+1/2)}}{| a_{k,k+1}^{(k+1/2)} |} y_{k+1}^{(k)} \right)^{-1},$$

and

$$y_j^{(k)} = d_k \bar{a}_{k,j} \quad \text{for} \quad j > k + 1.$$

An alternative approach to bidiagonalizing A is to generate the columns of P and Q sequentially as is done by the Lanczos algorithm for tridiagonalizing a symmetric matrix. The equations

$$AQ = PJ \quad \text{and} \quad P^*A = JQ^*$$

can be expanded in terms of the columns $\mathbf{p}_i$ of P and $\mathbf{q}_i$ of Q to yield

$$\begin{aligned} A\mathbf{q}_1 &= \alpha_1 \mathbf{p}_1, \\ \left.\begin{aligned} A\mathbf{q}_i &= \beta_{i-1}\mathbf{p}_{i-1} + \alpha_i \mathbf{p}_i, \\ \mathbf{p}_{i-1}^* A &= \alpha_{i-1}\mathbf{q}_{i-1}^* + \beta_{i-1}\mathbf{q}_i^*, \end{aligned}\right\} & \quad i = 2, 3, \cdots, n, \\ \mathbf{p}_n^* A &= \alpha_n \mathbf{q}_n^*. \end{aligned}$$

These lead to the following algorithm.

(2.4) *Choose* $\mathbf{q}_1$ *arbitrarily with* $\| \mathbf{q}_1 \| = 1$; *then set* $\mathbf{w}_1 = A\mathbf{q}_1$; $\alpha_1 = \| \mathbf{w}_1 \|$, $\mathbf{p}_1 = (\alpha_1)^{-1}\mathbf{w}_1$. *Set* $\mathbf{z}_i^* = \mathbf{p}_i^* A - \alpha_i \mathbf{q}_i^*$, $\beta_i = \| \mathbf{z}_i \|$, $\mathbf{q}_{i+1}^* = (\beta_i)^{-1}\mathbf{z}_i$ *for* $i = 1, 2, \cdots, n-1$; set $\mathbf{w}_i = A\mathbf{q}_i - \beta_{i-1}\mathbf{p}_{i-1}$, $\alpha_i = \| \mathbf{w}_i \|$, $\mathbf{p}_i = (\alpha_i)^{-1}\mathbf{w}_i$ *for* $i = 2, \cdots, n$.

Of course if α_k (β_k) equals zero, one must choose a new vector $\mathbf{p}_k$ ($\mathbf{q}_k$) which is orthogonal to the previously computed $\mathbf{p}_i$'s ($\mathbf{q}_i$'s). It is easy to show then by an inductive proof that the $\mathbf{p}_i$'s and $\mathbf{q}_i$'s generated by (2.4) are the first n columns of the desired unitary matrices P and Q.

Unless an α_k or β_k vanishes, the vector $\mathbf{q}_1$ will completely determine the rest of the vectors $\mathbf{p}_i$ and $\mathbf{q}_i$. Consequently $\mathbf{q}_1$ could be so chosen that the Lanczos-type algorithm would be mathematically identical to the Householder-type algorithm except for a diagonal unitary similarity transformation. But the Lanczos-type algorithm is unstable in the presence of rounding error unless reorthogonalization along the lines suggested by Wilkinson [30] is used. That is, one must restore the orthogonality of the generated vectors by using the Gram-Schmidt method to reorthogonalize each newly generated vector $\mathbf{p}_i$ or $\mathbf{q}_i$ to the previously generated vectors $\mathbf{p}_i$ or $\mathbf{q}_i$, respectively. With the extra work involved in this reorthogonalization, the Lanczos-type algorithm is noticeably slower than the previously described Householder algorithm except possibly if A is a sparse matrix.

3. Computation of the singular values. The singular values of A and of J are the same; they are the positive square roots of J^*J. Let them be called, in order,

$$\sigma_1 \geqq \sigma_2 \geqq \cdots \geqq \sigma_n, \quad \text{where} \quad \sigma_n \geqq 0.$$

These are the numbers which appear on the diagonal of the matrix Σ which was introduced in (1.1), i.e.,

$$\Sigma = \begin{pmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & 0 & \\ & & \sigma_3 & & & \\ & 0 & & \ddots & & \\ & & & & \sigma_n & \\ & & & 0 & & \end{pmatrix} \left.\vphantom{\begin{matrix}0\\0\\0\\0\\0\\0\end{matrix}}\right\} (m-n) \times n$$

Analogous to (1.1) is the decomposition

$$J = X\Sigma Y^* \tag{3.1}$$

in which X and Y are unitary matrices which, when they have been calculated, will lead via Theorem 1, $A = PJQ^*$, to the desired decomposition (1.1), namely,

$$A = U\Sigma V^*,$$

with $U = PX$, $V = QY$.

Evidently the last $m - n$ rows of zeros in J do not contribute to the singular values, nor do they have more than a trivial effect upon the determination of X and Y. Therefore it is convenient to delete J's last $m - n$ rows and write

$$J = \begin{pmatrix} \alpha_1 & \beta_1 & & & & & \\ & \alpha_2 & \beta_2 & & & & \\ & & \alpha_3 & \beta_3 & & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot & \beta_{n-1} \\ & \mathbf{0} & & & & & \cdot & \alpha_n \end{pmatrix}$$

without introducing any new notation to distinguish this $n \times n$ matrix J from the $m \times n$ matrix J. This can be done because the previous equations remain valid after the following process of "abbreviation":

(i) delete the last $m - n$ rows of zeros in J and Σ;

(ii) delete the last $m - n$ columns of P and U;

(iii) delete the last $m - n$ rows and columns of X; these coincide with the last rows and columns of an $m \times m$ unit matrix. In this section and the next we deal only with the abbreviated matrices.

The singular values σ_i of J are known (cf. [20]) to be related to the eigenvalues of the $2n \times 2n$ matrix

$$\tilde{J} = \begin{pmatrix} 0 & J \\ J^* & 0 \end{pmatrix},$$

whose eigenvalues are just $+\sigma_i$ and $-\sigma_i$ for $i = 1, 2, \cdots, n$. The calculation of the eigenvalues of $\tilde{J}$ is simplified conceptually by a transformation to tridiagonal form via a permutation similarity which will be exhibited now.

Consider the matrix equation

$$\begin{pmatrix} 0 & J \\ J^* & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \pm\mathbf{y} \end{pmatrix} = \pm\, \sigma \begin{pmatrix} \mathbf{x} \\ \pm\mathbf{y} \end{pmatrix}, \tag{3.2}$$

which, when expanded, takes the form

$$J\mathbf{y} = \sigma\mathbf{x}, \qquad J^*\mathbf{x} = \sigma\mathbf{y},$$

that is, $\alpha_i y_i + \beta_i y_{i+1} = \sigma x_i$, $\bar{\alpha}_1 x_1 = \sigma y_1$, $\alpha_n y_n = \sigma x_n$, $\bar{\beta}_{i-1} x_{i-1} + \bar{\alpha}_i x_i = \sigma y_i$. Now the substitution

$$z_{2i} = x_i, \qquad z_{2i-1} = \pm y_i,$$

leads to the equation

$$T\mathbf{z} = \pm\sigma\mathbf{z}$$

in which T is a $2n \times 2n$ tridiagonal matrix

$$T = \begin{pmatrix} 0 & \bar{\alpha}_1 & & & & & \\ \alpha_1 & 0 & \beta_1 & & & \mathbf{0} & \\ & \bar{\beta}_1 & 0 & \bar{\alpha}_2 & & & \\ & & \alpha_2 & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & \mathbf{0} & & & \cdot & \cdot & \bar{\alpha}_n \\ & & & & & \alpha_n & 0 \end{pmatrix}.$$

Clearly there exists a unitary diagonal matrix D such that the similarity transformation

$$DTD^* = S = \begin{pmatrix} 0 & s_1 & & & & & \\ s_1 & 0 & t_1 & & & \mathbf{0} & \\ & t_1 & 0 & s_2 & & & \\ & & s_2 & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & \mathbf{0} & & & \cdot & \cdot & s_n \\ & & & & & s_n & 0 \end{pmatrix} \tag{3.3}$$

yields a tridiagonal matrix S whose elements

$$s_i = |\alpha_i| \quad \text{and} \quad t_i = |\beta_i|$$

are all real and nonnegative.

There are a number of methods for obtaining the eigenvalues of a tridiagonal symmetric matrix. One of the most accurate and effective methods is to use Sturm sequences; an ALGOL program is given by Wilkinson [35]. One can simplify the algorithm, of course, by taking advantage of the fact that the diagonal elements of T are zero.

Another method of computing the singular values of J is to compute the eigenvalues of

$$J^*J = \begin{pmatrix} |\alpha_1|^2 & \bar{\alpha}_1\beta_1 & & & & \\ \alpha_1\bar{\beta}_1 & |\alpha_2|^2 + |\beta_1|^2 & \bar{\alpha}_2\beta_2 & & \mathbf{0} & \\ & \alpha_2\bar{\beta}_2 & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & \mathbf{0} & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \bar{\alpha}_{n-1}\beta_{n-1} \\ & & & & & \alpha_{n-1}\bar{\beta}_{n-1} & |\alpha_n|^2 + |\beta_{n-1}|^2 \end{pmatrix}.$$

Note again that since J^*J is a tridiagonal hermitian matrix there exists a

diagonal unitary matrix Δ such that

$$\Delta(J^*J)\Delta^* = K = \begin{pmatrix} s_1^2 & s_1 t_1 & & & & \\ s_1 t_1 & s_2^2 + t_1^2 & s_2 t_2 & & & \\ & s_2 t_2 & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & s_{n-1} t_{n-1} \\ & & & & & s_{n-1} t_{n-1} & s_n^2 + t_{n-1}^2 \end{pmatrix},$$

where $s_i = |\alpha_i|$ and $t_i = |\beta_i|$. Hence K is a real, symmetric, positive semi-definite, tridiagonal matrix and its eigenvalues can be computed by the Sturm sequence algorithm.

Although the smaller eigenvalues of A^*A are usually poorly determined, a simple error analysis shows that all the eigenvalues of K are as well-determined as those of T. The reason for this is that the computation of the Sturm sequences is algebraically the same for both T and K. Thus to use K is preferable since the total number of operations in calculating its eigenvalues is certainly less than in computing the eigenvalues of T.

4. Orthogonal vectors properly paired. We consider now the calculation of the unitary matrices X and Y which were introduced in (3.1):

$$J = X\Sigma Y^*.$$

As pointed out in §3, J can be transformed into a real matrix by means of unitary diagonal transformations, and we shall assume henceforth that this has been done (cf. (3.3)).

To each singular value σ_i corresponds a column $\mathbf{x}_i$ of X and $\mathbf{y}_i$ of Y satisfying

$$J\mathbf{y}_i = \sigma_i \mathbf{x}_i, \qquad J^t\mathbf{x}_i = \sigma_i \mathbf{y}_i. \tag{4.1}$$

Since $J^tJ\mathbf{y}_i = \sigma_i^2\mathbf{y}_i$ one could, in principle, calculate $\mathbf{y}_i$ as the normalized eigenvector of J^tJ corresponding to the eigenvalue σ_i^2, and $\mathbf{x}_i$ could be obtained fom the vector $J\mathbf{y}_i$ either by dividing it by σ_i or by normalizing it. However, if σ_i is small but not quite negligible, then $J\mathbf{y}_i$ will be so much contaminated by the roundoff errors left over after cancellation that the calculated $\mathbf{x}_i$ may well be neither normalized nor orthogonal to the previously calculated $\mathbf{x}$'s.

Another way to calculate X and Y might be to obtain the eigenvectors $\mathbf{x}_i$ and $\mathbf{y}_i$ of JJ^t and J^tJ independently, and then to order the vectors according to the ordering of the corresponding singular values σ_i. But if some of the singular values are too close together then the equations (4.1) are unlikely to be satisfied.

A third way, which seems at first to be free from the objections in the two preceding paragraphs, is to obtain the eigenvectors $\mathbf{z}_i$ of the $2n \times 2n$ tridiagonal matrix S of (3.3). Then the odd-numbered components of $\mathbf{z}_i$ would constitute a vector $\mathbf{y}_i$ and the even-numbered components a vector $\mathbf{x}_i$ which would satisfy (4.1). But in practice trouble shows up here in two ways. First, the facts that (4.1) is very nearly satisfied and that $\mathbf{z}_i$ has been normalized so that $\mathbf{z}_i{}^t\mathbf{z}_i = 2$ do not, in practice, though they should in theory, ensure that $\mathbf{x}_i{}^t\mathbf{x}_i = \mathbf{y}_i{}^t\mathbf{y}_i = 1$. Fortunately, unless σ_i is nearly negligible, one can normalize $\mathbf{x}_i$ and $\mathbf{y}_i$ separately without causing serious extra error. And if σ_i is negligible one can find $\mathbf{x}_i$ and $\mathbf{y}_i$ separately and ensure that they are normalized. The claims in the last two sentences can be proved, but there is no point in doing so because the second source of trouble is more drastic; if the $\mathbf{z}_i$'s are not orthogonal then neither will the $\mathbf{x}_i$'s be orthogonal, nor will the $\mathbf{y}_i$'s. The problem of ensuring that the $\mathbf{z}_i$'s are orthogonal is, in the present state of the art of computation, a serious one.

One way to ensure the orthogonality of calculated eigenvectors of a symmetric matrix is to use Jacobi's method [13], but this is slow. Another way is to reorthogonalize the calculated eigenvectors obtained, say, by inverse iteration with a tridiagonal matrix (cf. [30]); but the extra work done here is no guarantee that the vectors after orthogonalization will still be acceptable as eigenvectors. A third method, and one which seems very promising, involves the use of deflation to "remove" each eigenvector as it is obtained and thereby ensure orthogonality. We shall digress to discuss deflation methods suitable for use with symmetric tridiagonal matrices, and then adapt them to our bidiagonal matrix.

In this digression let K be some real symmetric tridiagonal matrix,

$$K = \begin{pmatrix} a_1 & b_1 & & & & \\ b_1 & a_1 & b_2 & & & \mathbf{0} \\ & b_2 & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ \mathbf{0} & & & & \cdot & \cdot & b_{n-1} \\ & & & & & b_{n-1} & a_n \end{pmatrix},$$

of which we already know an eigenvalue λ and its eigenvector $\mathbf{v}$. Rutishauser [27] shows how, in principle, to construct an orthogonal Hessenberg matrix H from the vector $\mathbf{v}$ so that $K_1 = H^tKH$ will have zero in place of b_{n-1}. After deleting the last row and column of the tridiagonal matrix K_1, another eigenvalue, eigenvector and deflation would be calculated, and so on. The eigenvectors of K would be the columns of an orthogonal matrix obtained by multiplying together all the H's. The orthogonality of the eigenvectors would be guaranteed (to within the limits of acceptable rounding

error) irrespective of the closeness of some eigenvalues of K. Rutishauser's method needs some modification because, as Wilkinson [34, p. 189] has shown, the effect of rounding errors in the transformation $K_1 = H^tKH$ could destroy K's tridiagonal form if $\mathbf{v}$'s first few components were too small.

In Rutishauser's deflation the matrix H can be interpreted as a product of 2×2 Jacobi-like rotations applied in succession to the first and second, second and third, third and fourth, $\cdots$, $(n-1)$th and nth rows and columns of K. After the first rotation, each rotation is chosen to annihilate a spurious term which was introduced by the previous rotation. For example, an asterisk in the following figure marks the spurious term which the third rotation must annihilate:

$$\begin{pmatrix} x & x & & & & & \\ x & x & x & * & & \mathbf{0} & \\ & x & x & x & & & \\ & * & x & x & x & & \\ & & & x & x & x & \\ & & & & \cdot & \cdot & \cdot \\ & \mathbf{0} & & & \cdot & \cdot & \cdot \\ & & & & & \cdot & \cdot \end{pmatrix}.$$

The first rotation, which fixes all the subsequent ones, can be determined from the first two elements of K's eigenvector $\mathbf{v}$ as suggested by Rutishauser [28, p. 226] or else from the first two elements of $K - \lambda I$. In effect, the deflation of the tridiagonal matrix K is equivalent to a QR-transformation applied to $K - \lambda I$ in the manner suggested by Ortega and Kaiser [22]. Unfortunately, this method also can be shown to be numerically unsatisfactory whenever $\mathbf{v}$'s last few components are abnormally small, because then the element in K_1 which replaces b_{n-1} in K remains too large, in general, to be ignored. Wilkinson [34, p. 187] hints at another method analogous to the one he described in [31, pp. 351–353]; we shall outline this method briefly because we believe it to be an effective compromise between Rutishauser's two schemes.

Having found an eigenvalue λ of K we calculate the corresponding eigenvector $\mathbf{v}$ and normalize it so that its largest component lies between $\frac{1}{2}$ and 2, say. The calculation of $\mathbf{v}$ can be accomplished using the inverse iteration described by Wilkinson [30]; but since there is no way to prove that, in general, one of his inverse iterations will suffice to provide an adequately accurate $\mathbf{v}$, we describe the following method whose properties can be established rigorously. We require that λ be chosen to be the algebraically greatest or else the least eigenvalue of K; this is no inconvenience since in the course of K's successive deflations each of its eigenvalues will be at some time the greatest or the smallest of the current matrix on hand. Next we apply

Gaussian elimination to $K - \lambda I$ *without* pivotal interchanges; there will be no trouble here (cf. [33, pp. 285–286]) provided floating point calculation is used and provided λ, if not exactly right, is larger than K's largest or smaller than K's smallest eigenvalue by perhaps a unit or so in λ's last place. The point here is that each nonzero pivot u_i in the elimination process must be of the same sign as $(K - \lambda I)$'s diagonal elements. The result of the elimination process is to express $K - \lambda I = LU$, where

$$L = \begin{pmatrix} 1 & & & & & \\ l_1 & 1 & & & & \\ & l_2 & 1 & & & \\ & & l_3 & \cdot & & \\ & & & \cdot & \cdot & \\ & & & & \cdot & \cdot \\ & & & & l_{n-1} & 1 \end{pmatrix}$$

and

$$U = \begin{pmatrix} u_1 & b_1 & & & & \\ & u_2 & b_2 & & & \\ & & u_3 & \cdot & & \\ & & & \cdot & \cdot & \\ & & & & \cdot & b_{n-1} \\ & & & & & u_n \end{pmatrix}.$$

Here $u_1 = a_1 - \lambda$ and $l_i = b_i/u_i$, $u_{i+1} = a_{i+1} - \lambda - l_i b_i$ for $i = 1, 2, \cdots, n - 1$. Next we attempt the solution of $(K - \lambda I)\mathbf{v} = \mathbf{r}$ using for $\mathbf{r}$ a vector whose elements all have the same magnitude but signs chosen to maximize the elements of $\mathbf{v}$. The choice of sign is accomplished by first solving $L\,\mathbf{s} = \mathbf{r}$ as follows:

$$s_1 = +1,$$

$$s_{i+1} = (-l_i s_i) + \operatorname{sgn}(-l_i s_i), \quad i = 1, 2, \cdots, n - 1.$$

The solution of $U\mathbf{v} = \mathbf{s}$ for $\mathbf{v}$ and the subsequent normalization of $\mathbf{v}$ complete the calculation. Provided no two pivots u_i have opposite signs one can show that the elements of $\mathbf{v}$ each have the same signs as the corresponding elements of the desired eigenvector despite the rounding errors committed during $\mathbf{v}$'s calculation. Furthermore, the elements of $\mathbf{r}$ exhibit the same signs as those of $+\mathbf{v}$ or $-\mathbf{v}$, depending upon the sign of the u_i's. Consequently the cosine of the angle between $\mathbf{r}$ and the correct eigenvector is at least $N^{-1/2}$ in magnitude, and finally we conclude that $K\mathbf{v}$ must differ from $\lambda\mathbf{v}$ by no more than a few units in the last place (cf. the argument in [30]). Now even if $\mathbf{v}$ is contaminated by components of the eigenvectors corresponding to other

eigenvalues pathologically close to λ, it will look enough like a true eigenvector to permit the deflation process to proceed. This process for calculating $\mathbf{v}$ is simpler and a little faster than Wilkinson's.

Now that we have $\mathbf{v}$ we proceed to the deflation along the lines outlined by Wilkinson. Each 2×2 rotation is embedded in an $n \times n$ matrix

$$P_j = \begin{pmatrix} 1 & & & & & & & & \\ & 1 & & & & & & 0 & \\ & & \ddots & & & & & & \\ & & & 1 & & & & & \\ & & & & c_j & s_j & & & \\ & & & & -s_j & c_j & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & 0 & & & & & & \ddots \\ & & & & & & & & & 1 \end{pmatrix}, \quad j = 1, 2, \cdots, n-1,$$

with c_j for its jth and $(j+1)$th diagonal elements, where $c_j^2 + s_j^2 = 1$. Suppose the products

$$P_j P_{j-1} \cdots P_1 (K - \lambda I) P_1^t \cdots P_{j-1}^t P_j^t \quad \text{and} \quad P_j P_{j-1} \cdots P_1 \mathbf{v}$$

have the forms

$$P_j\, P_{j-1} \cdots P_1 (K - \lambda I) P_1^t \cdots P_{j-1}^t\, P_j^t = \begin{pmatrix} x & x & & & & & \\ x & x & x & & & & \\ & x & x & h_j & w_j & & \\ & & h_j & x & x & & \\ & & w_j & x & x & x & \\ & & & & x & x & x \\ & & & & & x & x \end{pmatrix}$$

and

$$P_j\, P_{j-1} \cdots P_1\, \mathbf{v} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \phi_{j+1} \\ v_{j+2} \\ x \\ x \end{pmatrix}.$$

At the start we can take

$$h_0 = a_1 - \lambda, \qquad w_0 = b_1, \qquad \phi_1 = v_1.$$

To continue the deflation we must so determine P_{j+1} that its application will simultaneously annihilate the spurious element w_j in the jth row and column of the matrix as well as the vector's $(j+1)$th element ϕ_{j+1}. But in practice the accumulation of rounding errors will prevent the exact annihilation of both elements; instead we shall have to be satisfied with a P_{j+1} which leaves negligible residuals in place of w_j and ϕ_{j+1}. Wilkinson, having scaled $K - \lambda I$ so that its largest element lies between $\frac{1}{2}$ and 2, would use whichever of the equations

$$w_j c_{j+1} = h_j s_{j+1}, \qquad \phi_{j+1} c_{j+1} = -v_{j+2} s_{j+1},$$

contained the largest coefficient $|w_j|$, $|h_j|$, $|\phi_{j+1}|$, or $|v_{j+2}|$ to determine, in conjunction with $c_{j+1}^2 + s_{j+1}^2 = 1$, the values c_{j+1} and s_{j+1}. This method seems to be effective and we believe that it should always work, but since we cannot prove the method's infallibility, our work is incomplete.

Now we can show how to construct a deflation process for the bidiagonal matrix J. The first step is to obtain J's largest singular value σ; σ^2 is the largest eigenvalue of the tridiagonal matrix J^tJ (see §3). The next step requires the corresponding vectors $\mathbf{x}$ and $\mathbf{y}$ which can be obtained either by solving $J^tJ\mathbf{y} = \sigma^2\mathbf{y}$ for $\mathbf{y}$ and setting $\mathbf{x} = \sigma^{-1}J\mathbf{y}$, or by calculating σ's eigenvector $\mathbf{z}$ of S in (3.3) and hence obtaining $\mathbf{x}$ and $\mathbf{y}$ from $\mathbf{z}$'s even and odd components respectively. Both methods for getting $\mathbf{x}$ and $\mathbf{y}$ are numerically stable when performed in floating point. The deflation of J is accomplished by a sequence of 2×2 rotations applied in succession to its first and second columns, its first and second rows, its second and third columns, its second and third rows, its third and fourth columns, $\cdots$, its $(n-1)$th and nth rows. The ith rotation applied to rows i and $i+1$ of J must simultaneously annihilate a spurious subdiagonal element, introduced into row $i+1$ by the previous column rotation, and the ith element in the current $\mathbf{x}$-vector. The ith column rotation, except for the first, must annihilate a spurious term introduced by the previous row rotation into the $(i+1)$th column just above the first superdiagonal, and simultaneously the transpose of the ith column rotation must liquidate the ith element of the current $\mathbf{y}$-vector. The first column rotation would when applied to $J^tJ - \sigma^2 I$ annihilate the element in its first row and second column. At the end of the deflation process J's element b_{n-1} should have been replaced by zero. Of course, rounding errors will prevent the rotations from performing their roles exactly upon both the matrix J and the vectors $\mathbf{x}$ and $\mathbf{y}$, but just as in the deflation of a tridiagonal matrix we are able so to determine the rotations that negligible residuals are left behind in place of the elements we wished to liquidate.

After deflating J we delete its last row and column and repeat the process until J is deflated to a 1×1 matrix or the deflated J becomes negligibly small. At the end we multiply the rotations in reverse order to construct the

matrices X and Y which put J into the form (3.1):

$$J = X\Sigma Y.$$

(If J was complex, a unitary diagonal transformation should be incorporated here.) Finally the matrices P and Q of Theorem 1 are multiplied thus:

$$U = PX, \qquad V = QY,$$

to exhibit the decomposition (1.1):

$$A = U\Sigma V.$$

The two matrix multiplications PX and QY take most of the work.

5. Applications. The basic decomposition given by (1.1) has many applications in data analysis and applied mathematics. Suppose the matrix A arises from statistical observation, and we wish to replace A by another matrix $\hat{A}$ (say) which has lower rank p and is the best approximation to A in some sense. If we use the Frobenius norm (i.e., $\| A \|^2 = \operatorname{trace} A^*A$) then the problem has been solved [8] as follows.

THEOREM 2. *Let A be an $m \times n$ matrix of rank r which has complex elements. Let S_p be the set of all $m \times n$ matrices of rank $p < r$. Then for all $B \in S_p$,*

$$\| A - \hat{A} \| \leqq \| A - B \|,$$

where

$$\hat{A} = U\hat{\Sigma}V^*$$

and $\hat{\Sigma}$ is obtained from the Σ of (1.1) *by setting to zero all but its p largest singular values σ_i.*

Proof. Since $A = U\Sigma V^*$ and the Frobenius norm is unitarily invariant,

$$\| A - B \| = \| \Sigma - U^*BV \|.$$

Let $U^*BV = C$. Then

$$\| \Sigma - C \|^2 = \sum_{i=1}^{n} | \sigma_i - c_{ii} |^2 + \sum_{i \neq j} | c_{ij} |^2 \geqq \sum_{i=1}^{n} | \sigma_i - c_{ii} |^2.$$

Now it is convenient to order the singular values in such a way that $\sigma_i \geqq \sigma_{i+1}$. Thus, $\| A - B \|^2$ is minimized if $c_{ii} = \sigma_i$ for $i = 1, 2, \cdots, p$, and $c_{ij} = 0$ otherwise, i.e., for $C = \hat{\Sigma}$. Obviously,

$$\| A - \hat{A} \| = (\sigma_{p+1}^2 + \cdots + \sigma_r^2)^{1/2}.$$

Finding the vector $\mathbf{x}$ of shortest length which minimizes $\| \mathbf{b} - A\mathbf{x} \|$ is equivalent to finding the vector $\mathbf{y}$ of shortest length which minimizes

$\| \mathbf{c} - J\mathbf{y} \|$, where $\mathbf{c} = P^*\mathbf{b}$ and $\mathbf{y} = Q^*\mathbf{x}$. Here a natural question arises: is there any method which bypasses the complicated scheme in §3 and §4 for exhibiting J's singular values explicitly, and instead takes advantage of J's simple bidiagonal form to solve the least squares problem or to calculate J^I? Such a method, if it exists, must retain provision for intentional perturbations designed to delete, in effect, negligible singular values without inducing too large a discrepancy in J or A. Unfortunately, J's simple form is deceptive; even J's rank is hard to estimate without further calculation. For example, if J's rank r is less than n, then at least $n - r$ of the α_i's, and possibly more, should vanish; but in practice none of the α_i's may be negligible even though several may be very small compared with adjacent β_i's and, in consequence, a few of J's singular values may be negligible.

Perhaps the recurrence described by Greville [15] can be modified by the introduction of pivoting and then applied to J to calculate J^I. Until this scheme is worked out, the best method we can suggest for solving the least squares problem together with controllable perturbations is the following. Compute explicitly the representation

$$A = U\Sigma V^*,$$

decide which of the singular values are small enough to ignore, replace the remaining singular values by their reciprocals to obtain Σ^I, and finally use

$$A^I = V\Sigma^I U^*$$

to obtain the least squares solution $\mathbf{x} = A^I\mathbf{b}$. Once again, to ignore some singular values $\sigma_{r+1}, \sigma_{r+2}, \cdots, \sigma_n$ is equivalent to perturbing A by a matrix whose norm is $(\sum_{i=r+1}^{n} \sigma_i^2)^{1/2}$.

In some scientific calculations it is preferable that a given square matrix A be perturbed as little as possible (just rounding errors), but instead a perturbation $\delta\mathbf{b}$ in the right-hand side $\mathbf{b}$ of the equation $A\mathbf{x} = \mathbf{b}$ is permissible provided $\| \delta\mathbf{b} \|$ does not exceed a given tolerance ϵ. The substitution

$$\mathbf{y} = V^*\mathbf{x}, \qquad \mathbf{c} = U^*\mathbf{b}, \qquad \delta\mathbf{c} = U^*\delta\mathbf{b},$$

transforms the perturbed equation $A\mathbf{x} = \mathbf{b} + \delta\mathbf{b}$ into an equivalent diagonal system

$$\Sigma\mathbf{y} = \mathbf{c} + \delta\mathbf{c},$$

in which the permissible perturbation $\delta\mathbf{c}$ still satisfies

$$\| \delta\mathbf{c} \| < \epsilon. \tag{5.1}$$

Subject to this constraint, $\delta\mathbf{c}$ may be chosen to optimize some other criterion. For example, suppose we wish to minimize $\| \mathbf{x} \| = \| \mathbf{y} \|$. Then ideally $\delta\mathbf{c}$

should satisfy $\Sigma^2\delta\mathbf{c} + \lambda(\mathbf{c} + \delta\mathbf{c}) = 0$ with some suitable positive value of the Lagrange multiplier λ sufficiently small so that (5.1) is satisfied too. But for most practical purposes it is sufficient to use trial and error to determine λ to within a factor of two so that $\delta\mathbf{c} = -(I + \lambda^{-1}\Sigma^2)^{-1}\mathbf{c}$ will satisfy $\delta\mathbf{c}^*\delta\mathbf{c} < \epsilon^2$. The use of such a technique in least squares problems tends to suppress violent oscillation and cancellation which might otherwise detract from the usefulness of the solution $\mathbf{x}$.

A similar technique is valuable for the solution of the sets of linear equations which approximate integral equations of the form

$$\int A(i,j)x(j)\,dj = b(i).$$

Here the numerical treatment of the integral equation, in using singular values, is similar to the theoretical treatment found in [29]. Once again, the use of the decomposition $A = U\Sigma V^*$ aids the suppression of spurious oscillations in the function x.

We close with a warning; diagonal transformations can change A's singular values and A^I in a nontrivial way. Therefore some sort of equilibration may be necessary to allow each row and column of A to communicate its proper significance to the calculation. Two useful forms of equilibration are:

(i) scale each row and column of A in such a way that all the rows have roughly the same norm and so have all the columns;

(ii) scale each row and column of A in such a way that the absolute uncertainty in each element of A does not vary much from element to element. On least squares problems such equilibration is accomplished by weighting each residual in the sum of squares (see [2], [10], [11], [23] on equilibration algorithms, and [14]).

REFERENCES

[1] A. R. Amir-Moéz and A. L. Fass, *Elements of Linear Spaces*, Pergamon, New York, 1962, Chap. 12.

[2] F. L. Bauer, *Optimally scaled matrices*, Numer. Math., 5 (1963), pp. 73–87.

[3] A. Ben-Israel and A. Charnes, *Contributions to the theory of generalized inverses*, J. Soc. Indust. Appl. Math., 11 (1963), pp. 667–699.

[4] A. Ben-Israel and S. J. Wersan, *An elimination method for computing the generalized inverse of an arbitrary complex matrix*, J. Assoc. Comput. Mach., 10 (1963), pp. 532–537.

[5] G. Birkhoff and S. MacLane, *A Survey of Modern Algebra*, Macmillan, New York, 1953, Chap. VII, §6.

[6] J. W. Blattner, *Bordered matrices*, J. Soc. Indust. Appl. Math., 10 (1962), pp. 528–536.

[7] J. C. G. Boot, *The computation of the generalized inverse of singular or rectangular matrices*, Amer. Math. Monthly, 70 (1963), pp. 302–303.

[8] C. ECKART AND G. YOUNG, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[9] G. E. FORSYTHE AND P. HENRICI, *The cyclic Jacobi method for computing the principal values of a complex matrix*, Trans. Amer. Math. Soc., 94 (1960), pp. 1–23.

[10] G. E. FORSYTHE AND E. G. STRAUS, *On best conditioned matrices*, Proc. Amer. Math. Soc., 6 (1955), pp. 340–345.

[11] D. R. FULKERSON AND P. WOLFE, *An algorithm for scaling matrices*, SIAM Rev., 4 (1962), pp. 142–146.

[12] W. GIVENS, *Numerical computation of the characteristic values of a real symmetric matrix*. Oak Ridge National Laboratory, Report No. 1574, 1954.

[13] H. H. GOLDSTINE, F. J. MURRAY, AND J. VON NEUMANN, *The Jacobi method for real symmetric matrices*, J. Assoc. Comput. Mach., 6 (1959), pp. 59–96.

[14] G. H. GOLUB, *Comparison of the variance of minimum variance and weighted least squares regression coefficients*, Ann. Math. Statist., 34 (1963), pp. 984–991.

[15] T. N. E. GREVILLE, *Some applications of the pseudo-inverse of a matrix*, SIAM Rev., 2 (1960), pp. 15–22.

[16] M. R. HESTENES, *Inversion of matrices by biorthogonalization and related results*, J. Soc. Indust. Appl. Math., 6 (1958), pp. 51–90.

[17] A. S. HOUSEHOLDER, *Unitary triangularization of a nonsymmetric matrix*, J. Assoc. Comput. Mach., 5 (1958), pp. 339–342.

[18] E. G. KOGBETLIANTZ, *Solution of linear equations by diagonalization of coefficients matrix*, Quart. Appl. Math., 13 (1955), pp. 123–132.

[19] V. N. KUBLANOVSKAJA, *Some algorithms for the solution of the complete problem of eigenvalues*, V. Vyčisl. Mat.i. Mat. Fiz., 1 (1961), pp. 555–570.

[20] C. LANCZOS, *Linear Differential Operators*, Van Nostrand, London, 1961, Chap. 3.

[21] D. D. MORRISON, *Remarks on the unitary triangularization of a nonsymmetric matrix*, J. Assoc. Comput. Mach., 7 (1960), pp. 185–186.

[22] J. M. ORTEGA AND H. F. KAISER, *The LL^t and QR methods for symmetric tridiagonal matrices*, Comput. J., 6 (1963), pp. 99–101.

[23] E. E. OSBORNE, *On pre-conditioning of matrices*, J. Assoc. Comput. Mach., 7 (1960), pp. 338–345.

[24] ———, *On least squares solutions of linear equations*, Ibid., 8 (1961), pp. 628–636.

[25] R. PENROSE, *A generalized inverse for matrices*, Proc. Cambridge Philos. Soc., 51 (1955), pp. 406–413.

[26] ———, *On best approximate solutions of linear matrix equations*, Ibid., 52 (1956), pp. 17–19.

[27] H. RUTISHAUSER, *Deflation bei Bandmatrizen*, Z. Angew. Math. Phys., 10 (1959), pp. 314–319.

[28] ———, *On Jacobi rotation patterns*, Proc. Symp. Appl. Math. XV, Experimental Arithmetic, High Speed Computing, and Mathematics, Amer. Math. Soc., 1963, pp. 219–240.

[29] F. SMITHIES, *Integral Equations*, Cambridge University Press, Cambridge, 1958, Chap. VIII.

[30] J. H. WILKINSON, *The calculation of the eigenvectors of codiagonal matrices*, Comput. J., 1 (1958), pp. 148–152.

[31] ———, *Stability of the reduction of a matrix to almost triangular and triangular forms by elementary similarity transformations*, J. Assoc. Comput. Mach., 6 (1959), pp. 336–359.

[32] ———, *Householder's method for the solution of the algebraic eigenproblem*, Comput. J., 3 (1960), pp. 23–27.

[33] ———, *Error analysis of direct methods of matrix inversion*, J. Assoc. Comput. Mach., 8 (1961), pp. 281–330.

[34] ———, *Error analysis of eigenvalue techniques based on orthogonal transformations*, J. Soc. Indust. Appl. Math., 10 (1962), pp. 162–195.

[35] ———, *Calculation of the eigenvalues of a symmetric tridiagonal matrix by the method of bisection*, Numer. Math., 4 (1962), pp. 362–367.

JOURNAL OF COMPUTATIONAL PHYSICS 73, 325–348 (1987)

A Fast Algorithm for Particle Simulations*

L. GREENGARD AND V. ROKHLIN

Department of Computer Science, Yale University, New Haven, Connecticut 06520

Received June 10, 1986; revised February 5, 1987

An algorithm is presented for the rapid evaluation of the potential and force fields in systems involving large numbers of particles whose interactions are Coulombic or gravitational in nature. For a system of N particles, an amount of work of the order $O(N^2)$ has traditionally been required to evaluate all pairwise interactions. unless some approximation or truncation method is used. The algorithm of the present paper requires an amount of work proportional to N to evaluate all interactions to within roundoff error. making it considerably more practical for large-scale problems encountered in plasma physics, fluid dynamics, molecular dynamics, and celestial mechanics. © 1987 Academic Press, Inc.

1. INTRODUCTION

The study of physical systems by means of particle simulations is well established in a number of fields and is becoming increasingly important in others. The most classical example is probably celestial mechanics, but much recent work has been done in formulating and studying particle models in plasma physics, fluid dynamics, and molecular dynamics [5].

Thee are two major classes of simulation methods. Dynamical simulations follow the trajectories of N particles over some time interval of interest. Given initial positions $\{x_i\}$ and velocities, the trajectory of each particle is governed by Newton's second law of motion:

$$m_i \frac{d^2 x_i}{dt^2} = -\nabla_i \Phi \qquad \text{for} \quad i = 1, ..., N,$$

where m_i is the mass of ith particle and the force is obtained from the gradient of a potential function Φ. When one is interested in an equilibrium configuration of a set of particles rather than their time-dependent properties, an alternative approach is the Monte Carlo method. In this case, the potential function Φ has to be evaluated for a large number of configurations in an attempt to determine the potential minimum.

* The authors were supported in part by the Office of Naval Research under Grant N00014-82-K-0184.

0021-9991/87 $3.00

Copyright © 1987 by Academic Press, Inc.
All rights of reproduction in any form reserved

We restrict our attention in this paper to the case where the potential (or force) at a point is a sum of pairwise interactions. More specifically, we consider potentials of the form

$$\Phi = \Phi_{\text{far}} + (\Phi_{\text{near}} + \Phi_{\text{external}}),$$

where Φ_{near} (when present) is a rapidly decaying potential (e.g., Van der Waals), Φ_{external} (when present) is independent of the number of particles, and Φ_{far}, the far-field potential, is Coulombic or gravitational. Such models describe classical celestial mechanics and many problems in plasma physics and molecular dynamics. In the vortex method for incompressible fluid flow calculations [4], an important and expensive portion of the computation has the same formal structure (the stream function and the vorticity are related by Poisson's equation).

In a system of N particles, the calculation of Φ_{near} requires an amount of work proportional to N, as does the calculation of Φ_{external}. The decay of the Coulombic or gravitational potential, however, is sufficiently slow that all interactions must be accounted for, resulting in CPU time requirements of the order $O(N^2)$. In this paper a method is presented for the rapid (order $O(N)$) evaluation of these interactions for all particles.

There have been a number of previous efforts aimed at reducing the computational complexity of the N-body problem. Particle-in-cell methods [5] have received careful study and are used with much success, most notably in plasma physics. Assuming the potential satisfies Poisson's equation, a regular mesh is layed out over the computational domain and the method proceeds by:

(1) interpolating the source density at mesh points,

(2) using a "fast Poisson solver" to obtain potential values on the mesh,

(3) computing the force from the potential and interpolating to the particle positions.

The complexity of these methods is of the order $O(N + M \log M)$, where M is the number of mesh points. The number of mesh points is usually chosen to be proportional to the number of particles, but with a small constant of proportionality so that $M \ll N$. Therefore, although the asymptotic complexity for the method is $O(N \log N)$, the computational cost in practical calculations is usually observed to be proportional to N. Unfortunately, the mesh provides limited resolution, and highly nonuniform source distributions cause a significant degradation of performance. Further errors are introduced in step (3) by the necessity for numerical differentiation to obtain the force.

To improve the accuracy of particle-in-cell calculations, short-range interactions can be handled by direct computation, while far-field interactions are obtained from the mesh, giving rise to so-called particle–particle/particle–mesh (P^3M) methods [5]. For an implementation of these ideas in the context of vortex calculations, see [1]. While these algorithms still depend for their efficient performance on a

reasonably uniform distribution of particles, in theory they do permit arbitrarily high accuracy to be obtained. As a rule, when the required precision is relatively low, and the particles are distributed more or less uniformly in a rectangular region, P^3M methods perform satisfactorily. However, when the required precision is high (as, for example, in the modeling of highly correlated systems), the CPU time requirements of such algorithms tend to become excessive.

Appel [2] introduced a "gridless" method for many-body simulation with a computational complexity estimated to be of the order $O(N \log N)$. It relies on using a monopole (center-of-mass) approximation for computing forces over large distances and sophisticated data structures to keep track of which particles are sufficiently clustered to make the approximation valid. For certain types of problems, the method achieves a dramatic speed-up compared to the naive $O(N^2)$ approach. It is less efficient when the distribution of particles is relatively uniform and the required precision is high.

The algorithm we present uses multipole expansions to compute potentials or forces to whatever precision is required, and the CPU time expended is proportional to N. The approach we use is similar to the one introduced in [7] for the solution of boundary value problems for the Laplace equation. In the following section, we describe the necessary analytical tools, while Section 3 is devoted to a detailed description of the method.

2. Physical and Mathematical Preliminaries

In this paper, we consider a two-dimensional physical model which consists of a set of N charged particles with the potential and force obtained as the sum of pairwise interactions from Coulomb's law. Suppose that a point charge of unit strength is located at the point $(x_0, y_0) = \mathbf{x}_0 \in \mathbb{R}^2$. Then, for any $\mathbf{x} = (x, y) \in \mathbb{R}^2$ with $\mathbf{x} \neq \mathbf{x}_0$, the potential and electrostatic field due to this charge are described by the expressions

$$\phi_{\mathbf{x}_0}(x, y) = -\log(\|\mathbf{x} - \mathbf{x}_0\|)$$

and

$$E_{\mathbf{x}_0}(x, y) = \frac{(\mathbf{x} - \mathbf{x}_0)}{\|\mathbf{x} - \mathbf{x}_0\|^2},$$

respectively.

It is well known that $\phi_{\mathbf{x}_0}$ is harmonic in any region not containing the point $\mathbf{x}_0$. Moreover, for every harmonic function u, there exists an analytic function $w: \mathbb{C} \to \mathbb{C}$ such that $u(x, y) = \mathrm{Re}(w(x, y))$ and w is unique except for an additive constant. In the remainder of the paper we will work with analytic functions, making no distinction between a point $(x, y) \in \mathbb{R}^2$ and a point $x + iy = z \in \mathbb{C}$. We note that

$$\phi_{\mathbf{x}_0}(\mathbf{x}) = \mathrm{Re}(-\log(z - z_0)),$$

and, following standard practice, we will refer to the analytic function $\log(z)$ as the potential due to a charge. As we develop expressions for the potential due to more complicated charge distributions, we will continue to use complex notation and will refer to the corresponding analytic functions themselves as the potentials. The following lemma is an immediate consequence of the Cauchy–Riemann equations.

LEMMA 2.1. *If $u(x, y) = \mathrm{Re}(w(x, y))$ describes the potential field at (x, y), then the corresponding force field is given by*

$$\nabla u = (u_x, u_y) = (\mathrm{Re}(w'), -\mathrm{Im}(w')),$$

where w' is the derivative of w.

The following lemma is used in obtaining the multipole expansion for the field due to m charges.

LEMMA 2.2. *Let a point charge of intensity q be located at z_0. Then for any z such that $|z| > |z_0|$,*

$$\phi_{z_0}(z) = q \log(z - z_0) = q \left(\log(z) - \sum_{k=1}^{\infty} \frac{1}{k} \left(\frac{z_0}{z} \right)^k \right). \tag{2.1}$$

Proof. Note first that $\log(z - z_0) - \log(z) = \log(1 - z_0/z)$ and that $|z_0/z| < 1$. The lemma now follows from the expansion

$$\log(1 - \omega) = (-1) \sum_{k=1}^{\infty} \frac{w^k}{k},$$

which is valid for any w such that $|w| < 1$. ∎

THEOREM 2.1 (multipole expansion). *Suppose that m charges of strengths $\{q_i, i = 1, ..., m\}$ are located at points $\{z_i, i = 1, ..., m\}$, with $|z_i| < r$. Then for any $z \in \mathbb{C}$ with $|z| > r$, the potential $\phi(z)$ is given by*

$$\phi(z) = Q \log(z) + \sum_{k=1}^{\infty} \frac{a_k}{z^k}, \tag{2.2}$$

where

$$Q = \sum_{i=1}^{m} q_i \qquad \text{and} \qquad a_k = \sum_{i=1}^{m} \frac{-q_i z_i^k}{k}. \tag{2.3}$$

Furthermore, for any $p \geqslant 1$,

$$\left| \phi(z) - Q \log(z) - \sum_{k=1}^{p} \frac{a_k}{z^k} \right| \leqslant \alpha \left| \frac{r}{z} \right|^{p+1} \leqslant \left(\frac{A}{c-1} \right) \left(\frac{1}{c} \right)^p, \tag{2.4}$$

where

$$c = \left|\frac{z}{r}\right|, \qquad A = \sum_{i=1}^{m} |q_i|, \qquad \textit{and} \qquad \alpha = \frac{A}{1 - |r/z|}. \tag{2.5}$$

Proof. The form of the multipole expansion (2.2) is an immediate consequence of the preceding lemma and the fact that $\phi(z) = \sum_{i=1}^{m} \phi_{z_i}(z)$. To obtain the error bound (2.4), observe that

$$\left|\phi(z) - Q\log(z) - \sum_{k=1}^{p} \frac{a_k}{z^k}\right| = \left|\sum_{k=p+1}^{\infty} \frac{a_k}{z^k}\right|.$$

Substituting for a_k the expression in (2.3), we have

$$\left|\sum_{k=p+1}^{\infty} \frac{a_k}{z^k}\right| \leqslant A \sum_{k=p+1}^{\infty} \frac{r^k}{k\,|z|^k} \leqslant A \sum_{k=p+1}^{\infty} \left|\frac{r}{z}\right|^k = \alpha \left|\frac{r}{z}\right|^{p+1} = \left(\frac{A}{c-1}\right)\left(\frac{1}{c}\right)^p.$$

In particular, if $c \geqslant 2$, then

$$\left|\phi(z) - Q\log(z) - \sum_{k=1}^{p} \frac{a_k}{z^k}\right| \leqslant A\left(\frac{1}{2}\right)^p. \quad \blacksquare \tag{2.6}$$

Finally, we demonstrate, with a simple example, how multipole expansions can be used to speed up calculations with potential fields. Suppose that charges of strengths $q_1, q_2, ..., q_m$ are located at the points $x_1, x_2, ..., x_m \in \mathbb{C}$ and that $\{y_1, y_2, ..., y_n\}$ is another set of points in $\mathbb{C}$ (Fig. 1). We say that the sets $\{x_i\}$ and $\{y_i\}$ are *well separated* if there exist points $x_0, y_0 \in \mathbb{C}$ and a real $r > 0$ such that

$$|x_i - x_0| < r \qquad \text{for all} \quad i = 1, ..., m,$$
$$|y_j - y_0| < r \qquad \text{for all} \quad j = 1, ..., n,$$
$$|x_0 - y_0| > 3r.$$

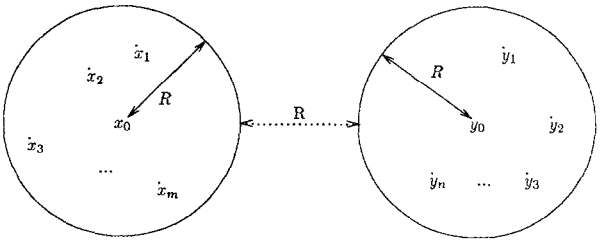


FIG. 1. Well-separated sets in the plane.

In order to obtain the potential (or force) at the points $\{y_j\}$ due to the charges at the points $\{x_i\}$ directly, we could compute

$$\sum_{i=1}^{m} \phi_{x_i}(y_j) \qquad \text{for all} \quad j=1, ..., n. \tag{2.7}$$

This clearly requires order nm work (evaluating m fields at n points). Now suppose that we first compute the coefficients of a p-term multipole expansion of the potential due to the charges $q_1, q_2, ..., q_m$ about x_0, using Theorem 2.1. This requires a number of operations proportional to mp. Evaluating the resulting multipole expansion at all points y_j requires order np work, and the total amount of computation is of the order $O(mp+np)$. Moreover, by (2.6),

$$\left| \sum_{i=1}^{m} \phi_{x_i}(y_j) - Q \log(y_j - x_0) - \sum_{k=1}^{p} \frac{a_k}{|y_j - x_0|^k} \right| \leqslant A \left(\frac{1}{2}\right)^p,$$

and in order to obtain a relative precision ε (with respect to the total charge), p must be of the order $-\log_2(\varepsilon)$. Once the precision is specified, the amount of computation has been reduced to

$$O(m) + O(n),$$

which is significantly smaller than nm for large n and m.

2.1. *Translation Operators and Error Bounds*

The following three lemmas constitute the principal analytical tool of this paper, allowing us to manipulate multipole expansions in the manner required by the fast algorithm. Lemma 2.3 provides a formula for shifting the center of a multipole expansion, Lemma 2.4 describes how to convert such an expansion into a local (Taylor) expansion in a circular region of analyticity, and Lemma 2.5 furnishes a mechanism for shifting the center of a Taylor expansion within a region of analyticity. We also derive error bounds associated with these translation operators which allow us to carry out numerical computations to any specified accuracy.

LEMMA 2.3. *Suppose that*

$$\phi(z) = a_0 \log(z - z_0) + \sum_{k=1}^{\infty} \frac{a_k}{(z - z_0)^k} \tag{2.8}$$

is a multipole expansion of the potential due to a set of m charges of strengths q_1, $q_2, ..., q_m$, *all of which are located inside the circle D of radius R with center at* z_0. *Then for z outside the circle* D_1 *of radius* $(R + |z_0|)$ *and center at the origin,*

$$\phi(z) = a_0 \log(z) + \sum_{l=1}^{\infty} \frac{b_l}{z^l}, \tag{2.9}$$

where

$$b_l = \left(\sum_{k=1}^{l} a_k z_0^{l-k} \binom{l-1}{k-1} \right) - \frac{a_0 z_0^l}{l}, \tag{2.10}$$

with $\binom{l}{k}$ *the binomial coefficients. Furthermore, for any* $p \geqslant 1$,

$$\left| \phi(z) - a_0 \log(z) - \sum_{l=1}^{p} \frac{b_l}{z^l} \right| \leqslant \left(A \Big/ \left(1 - \left| \frac{|z_0| + R}{z} \right| \right) \right) \left| \frac{|z_0| + R}{z} \right|^{p+1} \tag{2.11}$$

with A *defined in* (2.5).

Proof. The coefficients of the shifted expansion (2.9) are obtained by expanding the expression (2.8) into a Taylor series with respect to z_0. For the error bound (2.11), observe that the terms $\{b_l\}$ are the coefficients of the (unique) multipole expansion about the origin of those charges contained in the circle D, and Theorem 2.1 applies immediately with r replaced by $|z_0| + R$. ∎

Remark. Once the values $\{a_0, a_1, ..., a_p\}$ in the expansion (2.8) about z_0 are computed, we can obtain $\{b_1, ..., b_p\}$ exactly by (2.10). In other words, we may shift the center of a truncated multipole expansion without any loss of precision.

LEMMA 2.4. *Suppose that m charges of strengths* $q_1, q_2, ..., q_m$ *are located inside the circle* D_1 *with radius R and center at* z_0, *and that* $|z_0| > (c+1)R$ *with* $c > 1$ (*Fig.* 2). *Then the corresponding multipole expansion* (2.8) *converges inside the circle* D_2 *of radius R centered about the origin. Inside* D_2, *the potential due to the charges is described by a power series*:

$$\phi(z) = \sum_{l=0}^{\infty} b_l \cdot z^l, \tag{2.12}$$

where

$$b_0 = \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} (-1)^k + a_0 \log(-z_0), \tag{2.13}$$

and

$$b_l = \left(\frac{1}{z_0^l} \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} \binom{l+k-1}{k-1} (-1)^k \right) - \frac{a_0}{l \cdot z_0^l}, \qquad \textit{for} \quad l \geqslant 1. \tag{2.14}$$

Furthermore, for any $p \geqslant \max(2, 2c/(c-1))$, *an error bound for the truncated series is given by*

$$\left| \phi(z) - \sum_{l=0}^{p} b_l \cdot z^l \right| < \frac{A(4e(p+c)(c+1) + c^2)}{c(c-1)} \left(\frac{1}{c} \right)^{p+1}, \tag{2.15}$$

where A is defined in (2.5) *and e is the base of natural logarithms.*

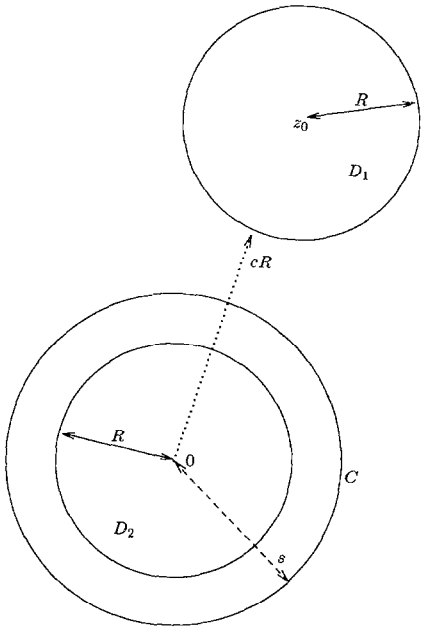


FIG. 2. Source charges $q_1, q_2, ..., q_l$ are contained in the circle D_1. The corresponding multipole expansion about z_0 converges inside D_2. C is a circle of radius s, with $s > R$.

Proof. We obtain the coefficients of the local expansion (2.12) from Maclaurin's theorem applied to the multipole expansion (2.8). To derive the error bound (2.15), we let $\gamma_0 = a_0 \log(-z_0)$, $\gamma_l = -(a_0/l \cdot z_0^l)$ for $l \geqslant 1$, and $\beta_l = b_l - \gamma_l$ for $l \geqslant 0$. Then

$$\left|\phi(z) - \sum_{l=0}^{p} b_l \cdot z^l\right| = \left|\sum_{l=p+1}^{\infty} b_l \cdot z^l\right| \leqslant S_1 + S_2 \tag{2.16}$$

with

$$S_1 = \left|\sum_{l=p+1}^{\infty} \gamma_l \cdot z^l\right| \qquad \text{and} \qquad S_2 = \left|\sum_{l=p+1}^{\infty} \beta_l \cdot z^l\right|.$$

A bound for S_1 is easily found by observing that

$$\begin{aligned} S_1 = \left|\sum_{l=p+1}^{\infty} \gamma_l z^l\right| &\leqslant |a_0| \sum_{l=p+1}^{\infty} \frac{z^l}{l \cdot z_0^l} \leqslant A \sum_{l=p+1}^{\infty} \frac{z^l}{l \cdot z_0^l} \\ &\leqslant A \sum_{l=p+1}^{\infty} \left(\frac{1}{c+1}\right)^l < A \sum_{l=p+1}^{\infty} \left(\frac{1}{c}\right)^l = \left(\frac{A}{c-1}\right)\left(\frac{1}{c}\right)^p. \end{aligned}$$

To obtain a bound for S_2, let C be a circle of radius s, where $s = cR((p-1)/p)$ (Fig. 2). Note first that for any $p \geqslant 2c/(c-1)$,

$$R < \frac{cR+R}{2} < s < cR.$$

Defining the function $\phi_1 \colon \mathbb{C} \backslash D_1 \to \mathbb{C}$ by the expression

$$\phi_1(z) = \phi(z) - a_0 \cdot \log(z - z_0),$$

and using Taylor's theorem for complex analytic functions (see [6, p. 190]), we obtain

$$S_2 = \left| \phi_1(z) - \sum_{l=0}^{p} \beta_l z^l \right| = \left| \sum_{l=p+1}^{\infty} \beta_l z^l \right| \leqslant M \Big/ \left(1 - \frac{|z|}{s}\right) \left(\frac{|z|}{s}\right)^{p+1},$$

where

$$M = \max_C |\phi_1(t)|.$$

Obviously, for any t lying on C,

$$|\phi_1(t)| \leqslant \sum_{k=1}^{\infty} \left| \frac{a_k}{(t-z_0)^k} \right|,$$

and it is easy to see that

$$|a_k| \leqslant AR^k \qquad \text{and} \qquad |t - z_0| \geqslant R + cR - s = R + cR/p.$$

After some algebraic manipulation, we have

$$M \leqslant A\left(\frac{pR+cR}{cR}\right) \qquad \text{and} \qquad 1 - \frac{|z|}{s} \geqslant \frac{cR-R}{cR+R}.$$

Observing that for any positive integer n and any integer $p \geqslant 2$,

$$\left(1 + \frac{1}{n}\right)^n \leqslant e \qquad \text{and} \qquad \left(1 + \frac{1}{p-1}\right)^2 \leqslant 4,$$

we obtain

$$\begin{aligned} S_2 &\leqslant \frac{A(pR+cR)(cR+R)}{cR(cR-R)} \left(\frac{|z|}{cR}\right)^{p+1} \left(\frac{p}{p-1}\right)^{p+1} \\ &\leqslant \frac{A(p+c)(c+1)}{c(c-1)} \left(\frac{1}{c}\right)^{p+1} \left(1 + \frac{1}{p-1}\right)^{p-1} \left(1 + \frac{1}{p-1}\right)^2 \\ &\leqslant \frac{4Ae(p+c)(c+1)}{c(c-1)} \left(\frac{1}{c}\right)^{p+1}. \end{aligned}$$

Adding the last expression to the error bound for S_1 completes the proof. ∎

The following lemma is an immediate consequence of Maclaurin's theorem. It describes an exact translation operation with a finite number of terms, and no error bound is needed.

LEMMA 2.5. *For any complex* z_0, z, and $\{a_k\}$, $k=0, 1, 2, ..., n$,

$$\sum_{k=0}^{n} a_k(z-z_0)^k = \sum_{l=0}^{n} \left(\sum_{k=l}^{n} a_k \binom{k}{l} (-z_0)^{k-l} \right) z^l. \tag{2.17}$$

3. THE FAST MULTIPOLE ALGORITHM

In this section, we present an algorithm for the rapid evaluation of the potentials and/or electrostatic fields due to distributions of charges. The central strategy used is that of clustering particles at various spatial lengths and computing interactions with other clusters which are sufficiently far away by means of multipole expansions. Interactions with particles which are nearby are handled directly.

To be more specific, let us consider the geometry of the computational box, depicted in Fig. 3. It is a square with sides of length one, centered about the origin of the coordinate system, and is assumed to contain all N particles of the system under consideration. The eight nearest neighbor boxes are also shown and will be needed in the next section when considering various boundary conditions. First, we will describe the method for free-space problems, where the boundary can be ignored and the only interactions to be accounted for involve particles within the computational box itself.

Fixing a precision ε, we choose $p \approx \log_2(\varepsilon)$ and specify that no interactions be computed for clusters of particles which are not *well separated.* This is precisely the condition needed for the error bounds (2.4), (2.11), and (2.15) to apply with $c=2$,

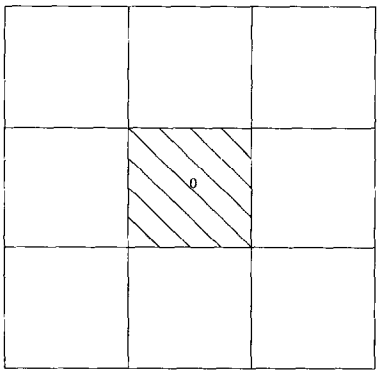


FIG. 3. The computational box (shaded) and its nearest periodic images. The box is centered at the origin "0" and has area one.

the truncation error to be bounded by 2^{-p}, and the desired precision to be achieved. In order to impose such a condition, we introduce a hierarchy of meshes which refine the computational box into smaller and smaller regions (Fig. 4). Mesh level 0 is equivalent to the entire box, while mesh level $l+1$ is obtained from level l by subdivision of each region into four equal parts. The number of distinct boxes at mesh level l is equal to 4^l. A tree structure is imposed on this mesh hierarchy, so that if *ibox* is a fixed box at level l, the four boxes at level $l+1$ obtained by subdivision of *ibox* are considered its children.

Other notation used in the description of the algorithm includes

$\Phi_{l,i}$ the p-term multipole expansion (about the box center) of the potential field created by the particles contained inside box i at level l,

$\Psi_{l,i}$ the p-term local expansion about the center of box i at level l, describing the potential field due to all particles outside the box and its nearest neighbors,

$\tilde{\Psi}_{l,i}$ the p-term local expansion about the center of box i at level l, describing the potential field due to all particles outside i's *parent* box and the *parent* box's nearest neighbors.

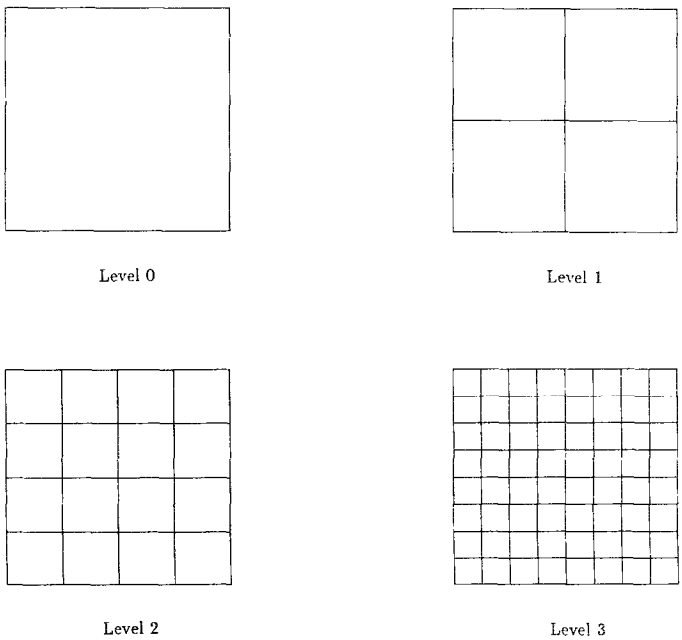


FIG. 4. The computational box and three levels of refinement.

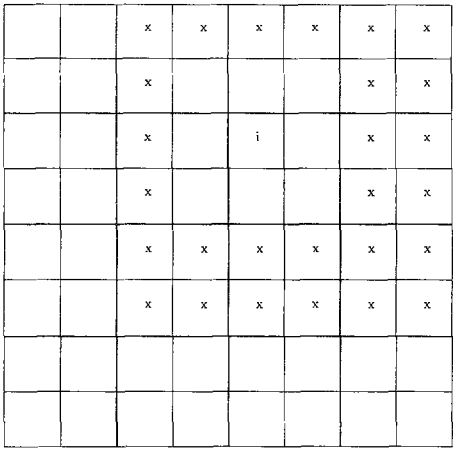


FIG. 5. Interaction list for box i. Thick lines correspond to mesh level 2 and thin lines to level 3. Boxes marked with an "x" are *well separated* from box i and contained within the nearest neighbors of box i's parent.

Interaction list for box i at level l, it is the set of boxes which are children of the nearest neighbors of i's parent and which are well separated from box i (Fig. 5).

Suppose now that at level $l-1$, the local expansion $\Psi_{l-1,i}$ has been obtained for all boxes. Then, by using Lemma 2.5 to shift (for all i) the expansion $\Psi_{l-1,i}$ to each of box i's children, we have, for each box j at level l, a local representation of the potential due to all particles outside of j's parent's neighbors, namely $\tilde{\Psi}_{l,j}$. The interaction list is, therefore, precisely that set of boxes whose contribution to the potential must be added to $\tilde{\Psi}_{l,j}$ in order to create $\Psi_{l,j}$. This is done by using Lemma 2.4 to convert the multipole expansions of these interaction boxes to local expansions about the current box center and adding them to the expansion obtained from the parent. Note also that with free-space boundary conditions, $\Psi_{0,i}$ and $\Psi_{1,i}$ are equal to zero since there are no well-separated boxes to consider, and we can begin forming local expansions at level 2.

The following is a formal description of the algorithm.

ALGORITHM.

Initialization

Choose a level of refinement $n \approx \log_4 N$, a precision ε, and set $p \approx \log_2(\varepsilon)$.

Upward Pass

Step 1

Comment [Form multipole expansions of potential field due to particles in each box about the box center at the finest mesh level.]

do $ibox = 1, ..., 4^n$

Form a p-term multipole expansion $\Phi_{n,ibox}$, by using Theorem 2.1.

enddo

Step 2

Comment [Form multipole expansions about the centers of all boxes at all coarser mesh levels, each expansion representing the potential field due to all particles contained in one box.]

do $l = n - 1, ..., 0$

do $ibox = 1, ..., 4^l$

Form a p-term multipole expansion $\Phi_{l,ibox}$, by using Lemma 2.3 to shift the center of each child box's expansion to the current box center and adding them together.

enddo

enddo

Downward Pass

Comment [In the downward pass, interactions are consistently computed at the coarsest possible level. For a given box, this is accomplished by including interactions with those boxes which are well separated and whose interactions have not been accounted for at the parent's level.]

Step 3

Comment [Form a local expansion about the center of each box at each mesh level $l \leqslant n - 1$. This local expansion describes the field due to all particles in the system that are not contained in the current box or its nearest neighbors. Once the local expansion is obtained for a given box, it is shifted, in the second inner loop to the centers of the box's children, forming the initial expansion for the boxes at the next level.]

Set $\tilde{\Psi}_{1,1} = \tilde{\Psi}_{1,2} = \tilde{\Psi}_{1,3} = \tilde{\Psi}_{1,4} = (0, 0, ..., 0)$

do $l = 1, ..., n - 1$

do $ibox = 1, ..., 4^l$

Form $\Psi_{l,ibox}$ by using Lemma 2.4 to convert the multipole expansion $\Phi_{l,j}$ of each box j in *interaction list* of box *ibox* to a local expansion about the center of box *ibox*, adding these local expansions together, and adding the result to $\tilde{\Psi}_{l,ibox}$.

enddo

do $ibox = 1, ..., 4^l$

Form the expansion $\tilde{\Psi}_{l+1,j}$ for *ibox*'s children by using Lemma 2.5 to expand $\Psi_{l,ibox}$ about the children's box centers.

enddo

enddo

Step 4

Comment [Compute interactions at finest mesh level.]

do $ibox = 1, ..., 4^n$

Form $\Psi_{l,ibox}$ by using Lemma 2.4 to convert the multipole expansion $\Phi_{l,j}$ of each box j in *interaction list* of box *ibox* to a local expansion about the center of box *ibox*, adding these local expansions together, and adding the result to $\tilde{\Psi}_{l,ibox}$.

enddo

Comment [Local expansions at finest mesh level are now available. They can be used to generate the potential or force due to all particles outside the nearest neighbor boxes at finest mesh level.]

Step 5

Comment [Evaluate local expansions at particle positions.]

do $ibox = 1, ..., 4^n$

For every particle p_j located at the point z_j in box *ibox*, evaluate $\Phi_{n,ibox}(z_j)$.

enddo

Step 6

Comment [Compute potential (or force) due to nearest neighbors directly.]

do $ibox = 1, ..., 4^n$

For every particle p_j in box *ibox*, compute interactions with all other particles within the box and its nearest neighbors.

enddo

Step 7

do $ibox = 1, ..., 4^n$

For every particle in box *ibox*, add direct and far-field terms together.

enddo

Remark. Each local expansion is described by the coefficients of a p-term polynomial. Direct evaluation of this polynomial at a point yields the potential. But, by Lemma 2.1, the force is immediately obtained from the derivative which is available analytically. There is no need for numerical differentiation. Furthermore, due to the analyticity of Φ', there exist error bounds for the force of exactly the same form as (2.4), (2.11), and (2.15).

A brief analysis of the algorithmic complexity is given below.

Step	Operation count	Explanation
1	order Np	Each particle contributes to one expansion at the finest level.
2	order Np^2	At the lth level, 4^l shifts involving order p^2 work per shift must be performed.
3	order $\leqslant 28Np^2$	There are at most 27 entries in the interaction list for each box at each level. An extra order Np^2 work is required for the second loop.
4	order $\leqslant 27Np^2$	Again, there are at most 27 entries in the interaction list for each box and $\approx N$ boxes.
5	order $\leqslant 27Np^2$	One p-term expansion is evaluated for each particle.
6	order $\frac{9}{2}Nk_n$	Let k_n be a bound on the number of particles per box at the finest mesh level. Interactions must be computed within the box and its eight nearest neighbors, but using Newton's third law, we need only compute half of the pairwise interactions.
7	order N	Adding two terms for each particle.

The estimate for the running time is therefore

$$N(-2a\log_2(\varepsilon)+56b(\log_2(\varepsilon))^2+4.5dk_n+e),$$

with the constants a, b, c, d, and e determined by the computer system, language, implementation, etc.

In addition to the asymptotic time complexity, asymptotic storage requirements are an important characteristic of a numerical procedure. The algorithm requires that $\Phi_{l,j}$ and $\Psi_{l,j}$ be stored, as well as the locations of the particles, their charges, and the results of the calculations (the potentials and/or electric fields). Since every box at every level has a pair of p-term expansions, Φ and Ψ, associated with it, and the lengths of all other storage arrays are proportional to N, it is easy to see that the asymptotic storage requirements of the algorithm are of the form

$$(\alpha+\beta p)\cdot N$$

or

$$(\alpha-\beta\log_2(\varepsilon))\cdot N,$$

with the coefficients α and β determined, as above, by the computer system, language, implementation, etc. In our numerical experiments, the actual storage requirements were of the order

$$(25 - \log_2(\varepsilon)) \cdot N$$

single precision words.

Remark. It is clear that the operation count for Step 6 assumes a reasonably homogeneous distribution of particles. If the distribution were highly non-homogeneous, then we would need to refine only those portions of space where the number of particles is large. Although its description is more involved, an adaptive version retains both the accuracy and the computational speed of the algorithm (see [3]).

4. Boundary Conditions

A variety of boundary conditions are used in particle simulations, including periodic boundary conditions, homogeneous Dirichlet or Neumann conditions, and several types of mixed conditions. The periodic case will be treated first in some detail. We then turn to the imposition of Dirichlet conditions and end with a brief discussion of the other cases.

4.1. *Periodic Boundary Conditions*

We begin by reconsidering the computational domain depicted in Fig. 5. At the end of the upward pass of the algorithm, we have a net multipole expansion

$$\Phi_{0,1}(z) = \sum_{k=1}^{p} \frac{a_k}{z^k} \tag{4.1}$$

for the entire computational box. This is then the expansion for each of the periodic images of the box with respect to its own center. All of these images except for the ones depicted in Fig. 3 are *well separated* from the computational box itself, and their induced fields are accurately representable by a p-term local expansion where, as before, $p \approx -\log_2(\varepsilon)$ is the number of terms needed to achieve a relative precision ε. We assume that the periodic particle model has no net charge and, therefore, that the local representation given by Lemma 2.4 can be written as

$$\Psi_{0,1} = \sum_{m=1}^{p} b_m \cdot z^m \tag{4.2}$$

with

$$b_m = \frac{1}{z_0^m} \sum_{k=1}^{p} \frac{a_k}{z_0^k} \binom{m+k-1}{k-1} (-1)^k, \qquad \text{with} \quad m = 0, 1, ..., p, \tag{4.3}$$

with z_0 the center of the image box under consideration.

Remark. In certain problems (e.g., cosmology), the computational box obviously cannot satisfy the condition of no net charge (mass). This condition is necessary for the potential to be well defined, since the logarithmic term becomes unbounded as $z_0 \to \infty$. Force calculations, however, may still be carried out. Indeed, using the notation of the algorithm, $\Phi_{l,i}$, $\Psi_{l,i}$, $\tilde{\Psi}_{l,i}$ are expansions of analytic functions representing the potential, so that their derivatives are also analytic functions (with the same regions of analyticity). Moreover, it is clear from Theorem 2.1 that the derivatives $\Phi'_{l,i}$ are described by pure inverse power series. Therefore, the identical formal structure of the algorithm can, due to Lemma 2.1, be used to evaluate force fields everywhere, bypassing the difficulty introduced by the logarithmic term. The only change required is that the initial expansions computed be the derivatives of the multipole expansions and not the multipole expansions themselves.

Note now that well-separated images of the computational cell are boxes whose centers z_0 have integer real and imaginary parts, with $\mathrm{Re}(z_0) \geqslant 2$ or $\mathrm{Im}(z_0) \geqslant 2$. Let S be the set of such centers. To account for the field due to all well-separated images, we form the coefficients for the local representation by adding the local shifted expansions of the form (4.3) for all $z_0 \in S$ to obtain

$$b_m^{\text{total}} = \sum_{k=1}^{p} a_k \binom{m+k-1}{k-1} (-1)^k \left(\sum_S \frac{1}{z_0^{m+k}} \right). \tag{4.4}$$

The summation over S for each inverse power of z_0 can be precomputed and stored. For $(m+k) > 2$, the series is absolutely convergent. However, for $(m+k) \leqslant 2$, the series is not absolutely convergent, and the computed value depends on the order of addition. Choosing a reasonable value for the sum of the series requires careful consideration of the physical model.

Suppose first that the only particle in the simulation is a charge of unit strength located at the origin. Then the periodic model corresponds to a uniform lattice of charges, and Newton's third law requires that the net force on each particle be zero. But the net force on the particle at the origin corresponds to the summation over S of $1/z_0$, so that we set

$$\sum_S \frac{1}{z_0} = 0.$$

To determine a value for the second term,

$$\sum_S \frac{1}{z_0^2},$$

suppose that the only particle in the simulation is a dipole of strength one, oriented along the x-axis and located at the origin. Then the periodic model is again a

uniform lattice and the difference in potential between the equivalent sites $(-\frac{1}{2}, 0)$ and $(\frac{1}{2}, 0)$ must be zero; i.e.,

$$\Phi_{(1/2,0)} - \Phi_{(-1/2,0)} \equiv \delta\Phi = 0. \tag{4.5}$$

The contribution to the potential difference, $\delta\Phi$, of a single dipole located at z_0 is

$$\frac{1}{z_0 - 1/2} - \frac{1}{z_0 + 1/2} = \frac{1}{z_0^2 - 1/4}.$$

Thus, we find that the potential difference due to the original dipole located at the origin is -4. For an image dipole located at z_0, with $|z_0| \geqslant 1$, we can expand the contribution to $\delta\Phi$ as follows:

$$\frac{1}{z_0^2 - 1/4} = \frac{1}{z_0^2} + \frac{1}{4z_0^4 - z_0^2}.$$

Now let S' be the set of the centers of all image boxes. That is, S' is the set of all points z_0 with integer real and imaginary parts, excluding the origin. Then

$$\delta\Phi = -4 + \sum_{S'} \frac{1}{z_0^2} + \sum_{S'} \frac{1}{4z_0^4 - z_0^2}.$$

A somewhat involved calculation shows that

$$\sum_{S'} \frac{1}{4z_0^4 - z_0^2} = 4 - \pi.$$

Therefore, to satisfy (4.5), we set

$$\sum_{S'} \frac{1}{z_0^2} = \pi.$$

Now

$$\sum_{S'} \frac{1}{z_0^2} = \sum_{S} \frac{1}{z_0^2} + \sum_{S' \setminus S} \frac{1}{z_0^2},$$

and the sum $\sum_{S' \setminus S} (1/z_0^2)$ is easily evaluated and found to be equal to zero. Therefore, we have

$$\sum_{S} \frac{1}{z_0^2} = \pi,$$

and the summation over S for every inverse power of z_0 is defined.

The procedure of converting the multipole expansion of the whole computational

cell $\Phi_{0,1}$ into a local expansion $\Psi_{0,1}$ which describes the potential field due to all well-separated images can be written, in the notation of the algorithm, as

$$\Psi_{0,1} = T \cdot \Phi_{0,1},$$

where T is a constant p by p matrix whose entries are defined by the formula

$$T_{m,k} = \binom{m+k-1}{k-1} (-1)^k \left(\sum_S \frac{1}{z_0^{m+k}} \right).$$

This can be viewed as the first step in the downward pass of the algorithm for periodic boundary conditions. At this point, we have accounted for all interactions excluding the ones within the immediate neighbors of the computational box as depicted in Fig. 3. But the expansions $\Phi_{l,i}$ for boxes inside the computational cell are also the expansions of the corresponding boxes inside the nearest neighbor images of the computational cell. By adding to the interaction list the appropriate boxes, we maintain the formal structure of the algorithm and the associated computational complexity.

4.2. *Dirichlet Boundary Conditions*

We turn now to the imposition of homogeneous Dirichlet boundary conditions, namely

$$\Phi(x, y) = 0 \qquad \text{for} \quad (x, y) \in \partial D,$$

where ∂D is the boundary of the computational domain. Analytically speaking, this can be accomplished by the method of images, described in detail below. In general terms, we consider the potential field to be composed of two parts; that is,

$$\Phi = \Phi_{\text{sources}} + \Phi_{\text{images}},$$

where Φ_{sources} is the field due to the particles inside the computational cell and Φ_{images} is the field due to selected image charges located outside the computational cell. The image charge positions and strengths are chosen so that

$$\Phi_{\text{sources}}(x, y) = -\Phi_{\text{images}}(x, y) \qquad \text{for} \quad (x, y) \in \partial D.$$

For the computational domain we are considering, appropriate locations for the image charges can be determined by an iterative process, illustrated in Fig. 6. We first reflect each particle p_i of charge strength σ_i in the computational cell across the top boundary line and place an image charge of strength $-\sigma_i$ at that location, generating an image box which we denote $\bar{C}$ (Fig. 6(b)). The set of image charges is denoted by V_1, and the field they induce is called Φ_{V_1}. Adding Φ_{V_1} to Φ_{sources} clearly enforces the desired condition along the top boundary. To impose the boundary condition along the bottom of the computational cell, we must reflect all charges (source and image) currently in the model across the bottom boundary,

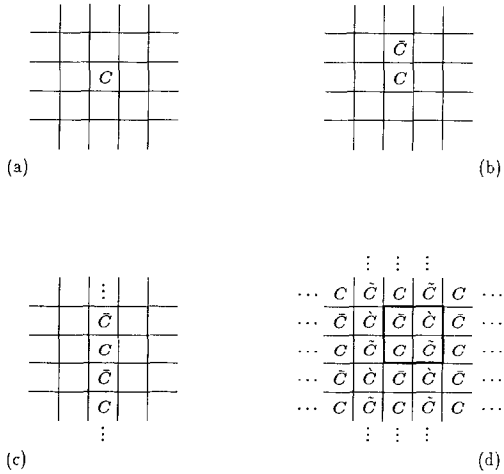


FIG. 6. The computational cell centered at the origin is represented by C. The cell $\bar{C}$ is the image of cell C reflected across the top boundary, with corresponding particles assigned charges of opposite sign. The cell $\tilde{C}$ is the image of cell C reflected across the left boundary, again with corresponding particles assigned charges of opposite sign. The cell $\grave{C}$ is the image of cell C reflected through the origin, with corresponding particles assigned charges of the same sign. Successive reflections across the four boundaries of the computational cell yield an infinite expansion of image boxes as indicated in (d).

generating two more image boxes (which are copies of C and $\bar{C}$). The set of all image charges after this second reflection step is denoted by V_2. Now, while $\Phi_{\text{sources}} + \Phi_{V_2}$ is equal to zero along the bottom boundary, the resulting field violates the top boundary condition. We therefore reflect again across the top boundary, creating two new image boxes and a new set of image charges V_3, such that $\Phi_{\text{sources}} + \Phi_{V_3}$ satisfies the top condition but violates the bottom one. By iterating in this manner, we generate a sequence of sets of image charges $\{V_i\}$ with

$$V_1 \subset V_2 \subset V_3 \subset \cdots \subset V,$$

where $V = \bigcup_{i=1}^{\infty} V_i$ is the set of charges contained in the infinite array of image boxes depicted in Fig. 6(c). It is easy to see that the corresponding sequence of image fields $\{\Phi_{V_i}\}$ converges inside the computational cell and that the potential field $\Phi_{\text{sources}} + \Phi_V$ does satisfy both the top and bottom boundary conditions.

In order to enforce the Dirichlet condition on the remaining two sides, we proceed analogously. First, we reflect all the charges currently in the model (the original sources plus the images in V) across the left boundary. This obviously does not affect the top and bottom conditions and enforces the homogeneous boundary condition along the left side of the computational cell. The current set of (all) image charges is now denoted H_1. Reflecting across the right boundary creates a new set H_2, with the field $\Phi_{\text{sources}} + \Phi_{H_2}$ satisfying the Dirichlet condition along the right

(but not the left) boundary. Repeated reflection across the left and right boundaries of the computational cell yields a sequence $\{H_i\}$ of infinite sets of image charges,

$$H_1 \subset H_2 \subset H_3 \subset \cdots \subset H,$$

where $H = \bigcup_{i=1}^{\infty} H_i$ is the set of charges contained in the two-dimensional family of image boxes depicted in Fig. 6(d). It is easy to see that the sequence $\{\Phi_{H_i}\}$ converges inside the computational cell, and we denote its limit by Φ_H. Finally, we observe that $\Phi_{\text{sources}} + \Phi_H = 0$ on the entire boundary ∂D.

From a computational point of view, the rate of convergence of the method of images is quite unsatisfactory. In conjunction with our algorithm, however, this method can be turned into an extremely efficient numerical tool. In the terminology previously introduced, all of the image boxes except the nearest neighbors of the computational cell are well separated and their induced fields can be represented by a single local expansion, denoted $\Psi_{0,1}$. Once the coefficients of this local expansion have been computed, we need only account for interactions within the nearest neighbors of the computational cell itself. To do this, as in the periodic case, we simply add the appropriate image boxes to the interaction lists of the boxes inside the computational cell.

Thus, it remains only to calculate $\Psi_{0,1}$. We first observe that the plane of images has a periodic structure with unit "supercell" centered at $(\frac{1}{2}, \frac{1}{2})$, indicated by thick lines in Fig. 6(d). But then, by the method developed above for periodic problems, we can obtain an expansion about the point $(\frac{1}{2}, \frac{1}{2})$ which accounts for all interactions beyond the nearest neighbors of the supercell. This expansion can be converted, by using Lemma 2.5, into an expansion about the origin (the center of the computational cell), which we call $\tilde{\Psi}_{0,1}$. It remains to account for the well-separated boxes which are contained inside the supercell's nearest neighbors. There are exactly 27 of these boxes, and their multipole expansions can be shifted (by using Lemma 2.4) to local expansions about the origin which are then added to $\tilde{\Psi}_{0,1}$ to finally form $\Psi_{0,1}$.

4.3. *Other Boundary Conditions*

While in certain applications, periodic or Dirichlet boundary conditions are called for, in others, Neumann or mixed conditions have to be imposed on the boundary of the computational domain. A typical example of a problem with mixed conditions is the computational cell with Neumann conditions on two opposing sides and Dirichlet conditions on the two others. Other models require periodic boundary conditions on the left and right sides of the computational cell and Dirichlet or Neumann conditions on the top and bottom. The imposition of these conditions is achieved by a procedure essentially identical to the one described above. By reflection and/or periodic extension, one first generates an entire plane of images. The local expansion $\Psi_{0,1}$ is then computed by an appropriate summation over all well-separated image boxes, and the remaining image interactions are handled as above.

5. Numerical Results

A computer program has been implemented utilizing the algorithm of this paper and capable of handling free-space problems and problems with periodic, homogeneous Dirichlet or homogeneous Neumann boundary conditions.

For testing purposes, we randomly assigned charged particles to positions in the computational cell (Fig. 7), with charge strengths between 0 and 1, and with the numbers of particles varying from 100 to 12,800. The calculations were performed on a VAX-8600, and the number of terms in the expansions $\Psi_{l,i}$, $\tilde{\Phi}_{l,i}$, $\Phi_{l,i}$ was set to 20, guaranteeing roughly 5-digit accuracy of the result. In each case, we performed the calculation in three ways: (1) via the algorithm of the present paper in single precision arithmetic; (2) directly (via formula (2.7)) in single precision arithmetic; and (3) via formula (2.7) in double precision arithmetic. The first two calculations were used to compare the speed and accuracy of our algorithm to those of the direct method. The direct evaluation of the field in double precision was used as a standard for comparing the relative accuracies of the first two computations. In all cases, the calculation was performed for a periodic model, the periodic boundary condition being imposed by means of the algorithm described in Section 4 of this paper.

The results of these numerical experiments are summarized in Table I. The first column of the table contains the numbers N of particles for which calculations have been performed. The second and third columns contain the CPU times T_{alg} that were required by the algorithm of the present paper to obtain the fields at all N particles, and the greatest relative error δ_{alg} obtained at any of the particles, respectively. Columns 4 and 5 contain the CPU times T_{dir} that were required by the direct algorithm (2.7) to obtain the fields at all N particles, and the greatest relative error δ_{dir} obtained at any one particle, respectively.

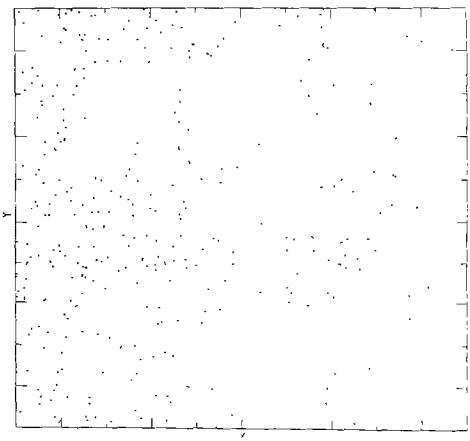


FIG. 7. 1600 randomly located charges in the computational cell.

TABLE I

Computational Results

N	T_{alg} (s)	δ_{alg}	T_{dir} (s)	δ_{dir}
100	0.6	1.1×10^{-5}	1.1	1.9×10^{-5}
200	1.4	4.1×10^{-5}	4.5	3.2×10^{-5}
400	2.0	3.6×10^{-5}	18	6.6×10^{-5}
800	3.8	4.6×10^{-5}	69	7.3×10^{-5}
1,600	6.6	1.4×10^{-5}	272	7.0×10^{-5}
3,200	16.5	0.9×10^{-5}	1088	3.1×10^{-5}
6,400	24.7	7.2×10^{-5}	4480	6.8×10^{-5}
12,800	60.9	3.0×10^{-5}	17920 (est.)	1.8×10^{-5}

Remark. For the example involving 12,800 particles, the algorithm of the present paper required about one minute of CPU time (see Table I). However, it was not considered practical to use the direct algorithm to evaluate the field at all 12,800 points, since it would take about 5 h of CPU time, without producing much useful information. Therefore, we used the direct algorithm to evaluate the field at only 100 of the 12,800 particles, both in single and double precision, and used the resulting data to estimate δ_{alg} and δ_{dir}. The value for T_{dir} in this case was estimated by scaling.

The following observations can be made from Table I:

1. The accuracy of the results produced by the algorithm is about the same as that predicted by the estimates (2.4), (2.11), and (2.15) for the number of terms we are using in the expansions $\Phi_{l,i}$, $\tilde{\Psi}_{l,i}$, $\Psi_{l,i}$. There is no evidence of accuracy problems due to truncation errors.

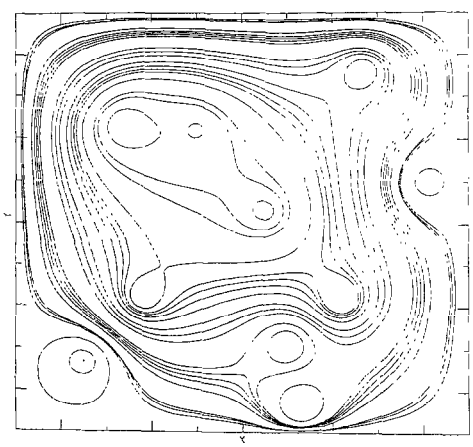


FIG. 8. The equipotential lines for the electrostatic field due to 10 randomly located charges in the computational cell, with homogeneous Dirichlet boundary conditions.

2. The calculation time grows linearly with the number of charges in the model, even though its behavior is somewhat erratic.

3. For as few as 1600 particles in the model, the computational effort required by the direct algorithm is roughly 40 times greater than that required by the algorithm of the present paper. For 12,800 particles, the effort is nearly 300 times greater.

Similar calculations have been performed for homogeneous Dirichlet and Neumann boundary conditions, and the observations made above for the periodic model are equally applicable in these cases.

For illustration, the equipotential lines for a box with 10 randomly distributed particles and Dirichlet boundary conditions are shown in Fig. 8. The entire calculation required 15 s of CPU time; about half the time was spent evaluating the field at more than 10,000 points, while the rest was used up by the plotting routine.

6. Conclusions

An algorithm has been constructed for the rapid evaluation of potential fields generated by ensembles of particles of the type encountered in plasma physics, molecular dynamics, fluid dynamics (the vortex method), and celestial mechanics. The algorithm is applicable both in the context of dynamical simulations and Monte Carlo simulations, provided that the fields to be evaluated are Coulombic in nature. The asymptotic CPU time estimate for the algorithm of the present paper is of the order $O(N)$, where N is the number of particles in the simulation, and the numerical examples presented in Section 5 indicate that even very large-scale problems result in acceptable CPU time requirements. In the present paper, a two-dimensional version of the algorithm is described. Generalizing this result to three dimensions is fairly straightforward and will be reported at a later date.

Acknowledgments

It is the authors' pleasure to thank Professor M. H. Schultz for drawing their attention to the subject of this paper, and for his continuing interest and support.

References

1. C. R. Anderson, *J. Comput. Phys.* **62**, 111 (1986).
2. A. W. Appel, *SIAM J. Sci. Stat. Comput.* **6**, 85 (1985).
3. J. Carrier, L. Greengard, and V. Rokhlin, "A Fast Adaptive Multipole Algorithm for Particle Simulations," Technical Report 496, Yale Computer Science Department, 1986 (unpublished).
4. A. J. Chorin, *J. Fluid. Mech.* **57**, 785 (1973).
5. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw–Hill, New York, 1981).
6. G. Polya and G. Latta, *Complex Variables* (Wiley, New York, 1974).
7. V. Rokhlin, *J. Comput. Phys.* **60**, 187 (1985).

Journal of Research of the National Bureau of Standards Vol. 49, No. 6, December 1952 Research Paper 2379

Methods of Conjugate Gradients for Solving Linear Systems[1]

Magnus R. Hestenes [2] and Eduard Stiefel [3]

An iterative algorithm is given for solving a system $Ax=k$ of n linear equations in n unknowns. The solution is given in n steps. It is shown that this method is a special case of a very general method which also includes Gaussian elimination. These general algorithms are essentially algorithms for finding an n dimensional ellipsoid. Connections are made with the theory of orthogonal polynomials and continued fractions.

1. Introduction

One of the major problems in machine computations is to find an effective method of solving a system of n simultaneous equations in n unknowns, particularly if n is large. There is, of course, no best method for all problems because the goodness of a method depends to some extent upon the particular system to be solved. In judging the goodness of a method for machine computations, one should bear in mind that criteria for a good machine method may be different from those for a hand method. By a hand method, we shall mean one in which a desk calculator may be used. By a machine method, we shall mean one in which sequence-controlled machines are used.

A machine method should have the following properties:

(1) The method should be simple, composed of a repetition of elementary routines requiring a minimum of storage space.

(2) The method should insure rapid convergence if the number of steps required for the solution is infinite. A method which—if no rounding-off errors occur—will yield the solution in a finite number of steps is to be preferred.

(3) The procedure should be stable with respect to rounding-off errors. If needed, a subroutine should be available to insure this stability. It should be possible to diminish rounding-off errors by a repetition of the same routine, starting with the previous result as the new estimate of the solution.

(4) Each step should give information about the solution and should yield a new and better estimate than the previous one.

(5) As many of the original data as possible should be used during each step of the routine. Special properties of the given linear system—such as having many vanishing coefficients—should be preserved. (For example, in the Gauss elimination special properties of this type may be destroyed.)

In our opinion there are two methods that best fit these criteria, namely, (a) the Gauss elimination method; (b) the conjugate gradient method presented in the present monograph.

There are many variations of the elimination method, just as there are many variations of the conjugate gradient method here presented. In the present paper it will be shown that both methods are special cases of a method that we call the method of conjugate directions. This enables one to compare the two methods from a theoretical point of view.

In our opinion, the conjugate gradient method is superior to the elimination method as a machine method. Our reasons can be stated as follows:

(a) Like the Gauss elimination method, the method of conjugate gradients gives the solution in n steps if no rounding-off error occurs.

(b) The conjugate gradient method is simpler to code and requires less storage space.

(c) The given matrix is unaltered during the process, so that a maximum of the original data is used. The advantage of having many zeros in the matrix is preserved. The method is, therefore, especially suited to handle linear systems arising from difference equations approximating boundary value problems.

(d) At each step an estimate of the solution is given, which is an improvement over the one given in the preceding step.

(e) At any step one can start anew by a very simple device, keeping the estimate last obtained as the initial estimate.

In the present paper, the conjugate gradient routines are developed for the symmetric and nonsymmetric cases. The principal results are described in section 3. For most of the theoretical considerations, we restrict ourselves to the positive definite symmetric case. No generality is lost thereby. We deal only with real matrices. The extension to complex matrices is simple.

The method of conjugate gradients was developed independently by E. Stiefel of the Institute of Applied Mathematics at Zurich and by M. R. Hestenes with the cooperation of J. B. Rosser, G. Forsythe, and L. Paige of the Institute for Numerical Analysis, National Bureau of Standards. The present account was prepared jointly by M. R. Hestenes and E. Stiefel during the latter's stay at the National Bureau of Standards. The first papers on this method were

[1] This work was performed on a National Bureau of Standards contract with the University of California at Los Angeles, and was sponsored (in part) by the Office of Naval Research, United States Navy.

[2] National Bureau of Standards and University of California at Los Angeles.

[3] University of California at Los Angeles, and Eidgenössische Technische Hochschule, Zurich, Switzerland.

given by E. Stiefel [4] and by M. R. Hestenes.[5] Reports on this method were given by E. Stiefel [6] and J. B. Rosser [7] at a Symposium [8] on August 23–25, 1951. Recently, C. Lanczos [9] developed a closely related routine based on his earlier paper on eigenvalue problem.[10] Examples and numerical tests of the method have been by R. Hayes, U. Hochstrasser, and M. Stein.

2. Notations and Terminology

Throughout the following pages we shall be concerned with the problem of solving a system of linear equations

$$\begin{aligned} a_{11}x_1+a_{12}x_2+\ .\ .\ .\ +a_{1n}x_n&=k_1\\ a_{21}x_1+a_{22}x_2+\ .\ .\ .\ +a_{2n}x_n&=k_2\\ .\ .\ .\quad .\ .\ .\quad .\ .\ .&\\ a_{n1}x_1+a_{n2}x_2+\ .\ .\ .\ +a_{nn}x_n&=k_n. \end{aligned} \tag{2:1}$$

These equations will be written in the vector form $Ax=k$. Here A is the matrix of coefficients (a_{ij}), x and k are the vectors $(x_1,\ .\ .\ .,x_n)$ and $(k_1,\ .\ .\ .,k_n)$. It is assumed that A is nonsingular. Its *inverse* A^{-1} therefore exists. We denote the *transpose* of A by A^*.

Given two vectors $x=(x_1,\ .\ .\ .,x_n)$ and $y=(y_1,\ .\ .\ .,y_n)$, their sum $x+y$ is the vector $(x_1+y_1,\ .\ .\ .,x_n+y_n)$, and ax is the vector $(ax_1,\ .\ .\ .,ax_n)$, where a is a scalar. The sum

$$(x,y)=x_1y_1+x_2y_2+\ .\ .\ .+x_ny_n$$

is their *scalar product*. The *length* of x will be denoted by

$$|x|=(x_1^2+\ .\ .\ .+x_n^2)^{\frac{1}{2}}=(x,x)^{\frac{1}{2}}.$$

The *Cauchy-Schwarz inequality* states that for all x,y:

$$(x,y)^2\leq(x,x)(y,y) \quad \text{or} \quad |(x,y)|\leq|x|\,|y|. \tag{2:2}$$

The matrix A and its transpose A^* satisfy the relation

$$(x,Ay)=\sum_{i,j=1}^{n}a_{ij}x_iy_j=(A^*x,y).$$

If $a_{ij}=a_{ji}$, that is, if $A=A^*$, then A is said to be *symmetric*. A matrix A is said to be *positive definite* in case $(x,Ax)>0$ whenever $x\neq0$. If $(x,Ax)\geqq0$ for all x, then A is said to be *nonnegative*. If A is symmetric, then two vectors x and y are said to be *conjugate* or *A-orthogonal* if the relation $(x,Ay)=(Ax,y)=0$ holds. This is an extension of the orthogonality relation $(x,y)=0$.

By an *eigenvalue* of a matrix A is meant a number λ such that $Ay=\lambda y$ has a solution $y\neq0$, and y is called a corresponding *eigenvector*.

Unless otherwise expressly stated the matrix A, with which we are concerned, will be *assumed to be symmetric and positive definite*. Clearly no loss of generality is caused thereby from a theoretical point of view, because the system $Ax=k$ is equivalent to the system $Bx=l$, where $B=A^*A$, $l=A^*k$. From a numerical point of view, the two systems are different, because of rounding-off errors that occur in joining the product A^*A. Our applications to the nonsymmetric case do not involve the computation of A^*A.

In the sequel we shall not have occasion to refer to a particular coordinate of a vector. Accordingly we may use subscripts to distinguish vectors instead of components. Thus x_0 will denote the vector $(x_{01},\ .\ .\ .,x_{0n})$ and x_i the vector $(x_{i1},\ .\ .\ .,x_{in})$. In case a symbol is to be interpreted as a component, we shall call attention to this fact unless the interpretation is evident from the context.

The *solution of the system $Ax=k$ will be denoted by* h; that is, $Ah=k$. If x is an estimate of h, the difference $r=k-Ax$ will be called the *residual* of x as an estimate of h. The quantity $|r|^2$ will be called the *squared residual*. The vector $h-x$ will be called the *error vector* of x, as an estimate of h.

3. Method of Conjugate Gradients (cg-Method)

The present section will be devoted to a description of a method of solving a system of linear equations $Ax=k$. This method will be called the *conjugate gradient method* or, more briefly, the cg-method, for reasons which will unfold from the theory developed in later sections. For the moment, we shall limit ourselves to collecting in one place the basic formulas upon which the method is based and to describing briefly how these formulas are used.

The cg-method is an iterative method which terminates in at most n steps if no rounding-off errors are encountered. Starting with an initial estimate x_0 of the solution h, one determines successively new estimates $x_0, x_1, x_2, \ .\ .\ .$ of h, the estimate x_i being closer to h than x_{i+1}. At each step the residual $r_i=k-Ax_i$ is computed. Normally this vector can be used as a measure of the "goodness" of the estimate x_i. However, this measure is not a reliable one because, as will be seen in section 18, it is possible to construct cases in which the *squared residual* $|r_i|^2$ increases at each step (except for the last) while the length of the error vector $|h-x_i|$ decreases monotonically. If no rounding-off error is encountered, one will reach an estimate $x_m (m\leqq n)$ at which $r_m=0$. This estimate is the desired solution h. Normally, $m=n$. However, since rounding-

[4] E. Stiefel, Uebereinige Methoden der Relaxationsrechnung, Z. angew. Math. Physik (3) (1952).

[5] M. R. Hestenes, Iterative methods for solving linear equations, NAML Report 52-9, National Bureau of Standards (1951).

[6] E. Stiefel, Some special methods of relaxation techniques, to appear in the Proceedings of the symposium (see footnote 8).

[7] J. B. Rosser, Rapidly converging iterative methods for solving linear equations, to appear in the Proceedings of the symposium (see footnote 8).

[8] Symposium on simultaneous linear equations and the determination of eigenvalues, Institute for Numerical Analysis, National Bureau of Standards, held on the campus of the University of California at Los Angeles (August 23–25, 1951).

[9] C. Lanczos, Solution of systems of linear equations by minimized iterations, NAML Report 52-13, National Bureau of Standards (1951).

[10] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Research NBS 45, 255 (1950) RP2133; Proceedings of Second Symposium on Large-Scale Digital Calculating Machinery, Cambridge, Mass., 1951, pages 164–206.

off errors always occur except under very unusual circumstances, the estimate x_n in general will not be the solution h but will be a good approximation of h. If the residual r_n is too large, one may continue with the iteration to obtain better estimates of h. Our experience indicates that frequently x_{n+1} is considerably better than x_n. One should not continue too far beyond x_n but should start anew with the last estimate obtained as the initial estimate, so as to diminish the effects of rounding-off errors. As a matter of fact one can start anew at any step one chooses. This flexibility is one of the principal advantages of the method.

In case the matrix A is *symmetric* and *positive definite*, the following formulas are used in the conjugate gradient method:

$$p_0=r_0=k-Ax_0 \qquad (x_0 \text{ arbitrary}) \tag{3:1a}$$

$$a_i=\frac{|r_i|^2}{(p_i,Ap_i)}, \tag{3:1b}$$

$$x_{i+1}=x_i+a_ip_i, \tag{3:1c}$$

$$r_{i+1}=r_i-a_iAp_i, \tag{3:1d}$$

$$b_i=\frac{|r_{i+1}|^2}{|r_i|^2}, \tag{3:1e}$$

$$p_{i+1}=r_{i+1}+b_ip_i. \tag{3:1f}$$

In place of the formulas (3:1b) and (3:1e) one may use

$$a_i=\frac{(p_i,r_i)}{(p_i,Ap_i)}, \tag{3:2a}$$

$$b_i=-\frac{(r_{i+1},Ap_i)}{(p_i,Ap_i)}. \tag{3:2b}$$

Although these formulas are slightly more complicated than those given in (3:1), they have the advantage that scale factors (introduced to increase accuracy) are more easily changed during the course of the computation.

The *conjugate gradient method* (cg-method) is given by the following steps:

Initial step: Select an estimate x_0 of h and compute the residual r_0 and the direction p_0 by formulas (3:1a).

General routine: Having determined the estimate x_i of h, the residual r_i, and the direction p_i, compute x_{i-1}, r_{i+1}, and p_{i-1} by formulas (3:1b), . . ., (3:1f) successively.

As will be seen in section 5, the residuals r_0, r_1, . . . are mutually orthogonal, and the direction vectors p_0, p_1, . . . are mutually conjugate, that is,

$$(r_i, r_j)=0, \quad (p_i, Ap_j)=0 \qquad (i\neq j). \tag{3:3}$$

These relations can be used as checks.

Once one has obtained the set of n mutually conjugate vectors p_0, . . ., p_{n-1} the solution of

$$Ax=k' \tag{3:4}$$

can be obtained by the formula

$$x=\sum_{i=0}^{n-1}\frac{(p_i,k')}{(Ap_i,p_i)}\,p_i. \tag{3:5}$$

It follows that, if we denote by p_{ij} the jth component of p_i, then

$$\sum_{i=0}^{n-1}\frac{p_{ij}p_{ik}}{(p_i,Ap_i)}$$

is the element in the jth row and kth column of the inverse A^{-1} of A.

There are two objections to the use of formula (3:5). First, contrary to the procedure of the general routine (3:1), this would require the storage of the vectors p_0, p_1, ... This is impractical, particularly in large systems. Second, the results obtained by this method are much more influenced by rounding-off errors than those obtained by the step-by-step routine (3:1).

In the cg-method *the error vector $h-x$ is diminished in length* at each step. The quantity $f(x)=(h-x, A(h-x))$, called the *error function*, is also diminished at each step. But the *squared residual* $|r|^2=|k-Ax|^2$ normally oscillates and may even increase. There is a modification of the cg-method where all three quantities diminish at each step. This modification is given in section 7. It has an advantage and a disadvantage. Its disadvantage is that the error vector in each step is longer than in the original method. Moreover, the computation is complicated, since it is a routine superimposed upon the original one. However, in the special case where the given linear equation system arises from a difference approximation of a boundary-value problem, it can be shown that the estimates are smoother in the modified method than in the original. This may be an advantage if the desired solution is to be differentiated afterwards.

Concurrently with the solution of a given linear system, characteristic roots of its matrix may be obtained: compute the values of the polynomials R_0, R_1, . . . and P_0, P_1, . . . at λ by the iteration

$$R_0=P_0=1$$

$$R_{i+1}=R_i-\lambda a_iP_i$$

$$P_{i+1}=R_{i+1}+b_iP_i. \tag{3:6}$$

The last polynomial $R_m(\lambda)$ is a factor of the characteristic polynomial of A and coincides with it when $m=n$. The characteristic roots, which are the zeros of $R_m(\lambda)$, can be found by Newton's methods without actually computing the polynomial $R_m(\lambda)$ itself. One uses the formulas

$$\lambda_{k+1}=\lambda_k-\frac{R_m(\lambda_k)}{R'_m(\lambda_k)}, \tag{3:7}$$

where $R_m(\lambda_k)$, $R'_m(\lambda_k)$ are determined by the iteration (3:6) and,

$$R'_0=P'_0=0$$

$$R'_{i+1}=R'_i-\lambda a_i P'_i-a_i P_i$$

$$P'_{i+1}=R'_i+b_i P'_i$$

with $\lambda=\lambda_k$. In this connection, it is of interest to observe that if $m=n$, the determinant of A is given by the formula

$$\det(A)=\frac{1}{a_0 a_1 \cdots a_{n-1}}.$$

The cg-method can be extended to the case in which A is a *general nonsymmetric and nonsingular* matrix. In this case one replaces eq (3:1) by the set

$$\begin{aligned} r_0&=k-Ax_0, \qquad p_0=A^*r_0,\\ a_i&=\frac{|A^*r_i|^2}{|Ap_i|^2}\\ x_{i+1}&=x_i+a_ip_i,\\ r_{i+1}&=r_i-a_iAp_i,\\ b_i&=\frac{|A^*r_{i+1}|^2}{|A^*r_i|^2}\\ p_{i+1}&=A^*r_{i+1}+b_ip_i. \end{aligned} \tag{3:8}$$

This system is discussed in section 10.

4. Method of Conjugate Directions (cd-Method)[1]

The cg-method can be considered as a special case of a general method, which we shall call the *method of conjugate directions* or more briefly the cd-method. In this method, the vectors $p_0, p_1, \ldots$ are selected to be mutually conjugate but have no further restrictions. It consists of the following routine:

Initial step. Select an estimate x_0 of h (the solution), compute the residual $r_0=k-Ax_0$, and choose a direction p_0.

General routine. Having obtained the estimate x_i of h, the residual $r_i=k-Ax_i$ and the direction p_i, compute the new estimate x_{i+1} and its residual r_{i+1} by the formulas

$$a_i=\frac{(p_i,r_i)}{(p_i,Ap_i)}, \tag{4:1a}$$

$$x_{i+1}=x_i+a_ip_i, \tag{4:1b}$$

$$r_{i+1}=r_i-a_iAp_i. \tag{4:1c}$$

Next select a direction p_{i+1} conjugate to $p_0, \ldots, p_i$, that is, such that

$$(p_{i+1},Ap_j)=0 \qquad (j=0,1,\ldots,i). \tag{4:2}$$

In a sense the cd-method is not precise, in that no formulas are given for the computation of the directions $p_0, p_1, \ldots$. Various formulas can be given, each leading to a special method. The formula (3:1f) leads to the cg-method. It will be seen in section 12 that the case in which the p's are obtained by an A-orthogonalization of the basic vectors $(1,0,\ldots,0)$, $(0,1,0,\ldots)$, $\ldots$ leads essentially to the Gauss elimination method.

The basic properties of the cd-method are given by the following theorems.

Theorem 4:1. *The direction vectors $p_0, p_1, \cdots$ are mutually conjugate. The residual vector r_i is orthogonal to $p_0, p_1, \cdots, p_{i-1}$. The inner product of p_i with each of the residuals $r_0, r_1, \cdots, r_i$ is the same. That is,*

$$(p_i,Ap_j)=0 \qquad (i\neq j) \tag{4:3a}$$

$$(p_j,r_i)=0 \qquad (j=0,1,\cdots,i-1) \tag{4:3b}$$

$$(p_i,r_0)=(p_i,r_1)=\cdots=(p_i,r_i). \tag{4:3c}$$

The scalar a_i can be given by the formula

$$a_i=\frac{(p_i,r_0)}{(p_i,Ap_i)} \tag{4:4}$$

in place of (4:1a).

Equation (4:3a) follows from (4:2). Using (4:1c), we find that

$$(p_j,r_{k+1})=(p_j,r_k)-a_k(p_j,Ap_k).$$

If $j=k$ we have, by (4:1a), $(p_k,r_{k+1})=0$. Moreover, by (4:3a) $(p_j,r_{k+1})=(p_j,r_k)$, $(j\neq k)$. Equations (4:3b) and (4:3c) follow from these relations. The formula (4:4) follows from (4:3c) and (4:1a).

As a consequence of (4:4) the estimates $x_1, x_2, \cdots$ of h can be computed without computing the residuals $r_0, r_1, \cdots$, provided that the choice of the direction vectors $p_0, p_1, \cdots$ is independent of these residuals.

Theorem 4:2. *The cd-method is an m-step method ($m\leqq n$) in the sense that at the mth step the estimate x_m is the desired solution h.*

For let m be the first integer such that $y_0=h-x_0$ is in the subspace spanned by $p_0, \cdots, p_{m-1}$. Clearly, $m\leqq n$, since the vectors $p_0, p_1, \cdots$ are linearly independent. We may, accordingly, choose scalars $\alpha_0, \cdots, \alpha_{m-1}$ such that

$$y_0=\alpha_0p_0+\cdots+\alpha_{m-1}p_{m-1}.$$

Hence,

$$h=x_0+\alpha_0p_0+\cdots+\alpha_{m-1}p_{m-1}.$$

Moreover,

$$r_0=k-Ax_0=A(h-x_0)=\alpha_0Ap_0+\cdots+\alpha_{m-1}Ap_{m-1}.$$

[1] This method was presented from a different point of view by Fox, Huskey, and Wilkinson on p. 149 of a paper entitled "Notes on the solution of algebraic linear simultaneous equations," Quarterly Journal of Mechanics and Applied Mathematics c. 2, 149–173 (1948).

Computing the inner product (p_i, r_0) we find by (4:3a) and (4:4) that $\alpha_i = a_i$, and hence that $h = x_m$, as was to be proved.

The cd-method can be looked upon as a relaxation method. In order to establish this result, we introduce the function

$$f(x) = (h-x, A(h-x)) = (x, Ax) - 2(x,k) + (h,k). \quad (4:5)$$

Clearly, $f(x) \geqq 0$ and $f(x) = 0$ if, and only if, $x = h$. The function $f(x)$ can be used as a measure of the "goodness" of x as an estimate of h. Since it plays an important role in our considerations, it will be referred to as the *error function*. If p is a direction vector, we have the useful relation

$$f(x+\alpha p) = f(x) - 2\alpha(p,r) + \alpha^2(p, Ap), \quad (4:6)$$

where $r = k - Ax = A(h-x)$, as one readily verifies by substitution. Considered as a function of α, the function $f(x+\alpha p)$ has a minimum value at $\alpha = a$, where

$$a = \frac{(p,r)}{(p, Ap)}. \quad (4:7)$$

This minimum value differs from $f(x)$ by the quantity

$$f(x) - f(x+ap) = a^2(p, Ap) = \frac{(p,r)^2}{(p, Ap)}. \quad (4:8)$$

Comparing (4:7) with (4:1a), we obtain the first two sentences of the following result:

Theorem 4:3. *The point x_i minimizes $f(x)$ on the line $x = x_{i-1} + \alpha p_{i-1}$. At the i-th step the error $f(x_{i-1})$ is relaxed by the amount*

$$f(x_{i-1}) - f(x_i) = \frac{(p_{i-1}, r_{i-1})^2}{(p_{i-1}, Ap_{i-1})}. \quad (4:9)$$

In fact, the point x_i minimizes $f(x)$ on the i-dimensional plane P_i of points

$$x = x_0 + \alpha_0 p_0 + \ldots + \alpha_{i-1} p_{i-1}, \quad (4:10)$$

where $\alpha_0, \ldots, \alpha_{i-1}$ are parameters. This plane contains the points $x_0, x_1, \ldots, x_i$.

In view of this result the cd-method is a method of relaxation of the error function $f(x)$. An iteration of the routine may accordingly be referred to as a relaxation.

In order to prove the third sentence of the theorem observe that at the point (4:10)

$$f(x) = f(x_0) - \sum_{j=0}^{i-1} [2\alpha_j(p_j, r_0) - \alpha_j^2(p_j, Ap_j)].$$

At the minimum point we have

$$\alpha_j = \frac{(p_j, r_0)}{(p_j, Ap_j)},$$

and hence $\alpha_j = a_j$, by (4:4). The minimum point is accordingly the point x_i, as was to be proved.

Geometrically, the equation $f(x) = \text{const.}$ defines an ellipsoid of dimension $n-1$. The point at which $f(x)$ has a minimum is the center of the ellipsoid and is the solution of $Ax = k$. The i-dimensional plane P_i, described in the last theorem, cuts the ellipsoid $f(x) = f(x_0)$ in an ellipsoid E_i of dimension $i-1$, unless E_i is the point x_0 itself. (In the cg-method, E_i is never degenerate, unless $x_0 = h$.) The point x_i is the center of E_i. Hence we have the corollary:

Corollary 1. The point x_i is the center of the $(i-1)$-dimensional ellipsoid in which the i-dimensional plane P_i intersects the $(n-1)$-dimensional ellipsoid $f(x) = f(x_0)$.

Although the function $f(x)$ is the fundamental error function which decreases at each step of the relaxation, one is unable to compute $f(x)$ without knowing the solution h we are seeking. In order to obtain an estimate of the magnitude of $f(x)$ we may use the following:

Theorem 4:4. *The error vector $y = h - x$, the residual $r = k - Ax$, and the error function $f(x)$ satisfy the relations*

$$\frac{|r|^2}{\mu(r)} \leqq f(x) \leqq \frac{|r|^2}{\mu(y)}, \quad (4:11)$$

where $\mu(z)$ is the Rayleigh quotient

$$\mu(z) = \frac{(z, Az)}{|z|^2}. \quad (4:12)$$

The Rayleigh quotient of the error vector y does not exceed that of the residual r, that is,

$$\mu(y) \leqq \mu(r). \quad (4:13)$$

Moreover,

$$\frac{|r|}{\mu(r)} \leqq |y| \leqq \frac{|r|}{\mu(y)}. \quad (4:14)$$

The proof of this result is based on the Schwarzian quotients

$$\frac{(z, Az)}{(z, z)} \leqq \frac{(Az, Az)}{(z, Az)} \leqq \frac{(Az, A^2z)}{(Az, Az)}. \quad (4:15)$$

The first of these follows from the inequality of Schwarz

$$|(p,q)|^2 \leqq (p,p)(q,q) \quad (4:16)$$

by choosing $p = z$, $q = Az$. The second is obtained by selecting $p = Bz$, $q = B^3z$, where $B^2 = A$.

In order to prove theorem 4:4 recall that if we set $y = h - x$, then

$$r = k - Ax = A(h-x) = Ay$$

$$f(x) = (y, Ay)$$

by (4:5). Using the inequalities (4:15) with $z = y$, we see that

$$\mu(y) = \frac{(y, Ay)}{(y, y)} \leqq \frac{(Ay, Ay)}{(y, Ay)} = \frac{|r|^2}{f(x)} \leqq \frac{(Ay, A^2y)}{(Ay, Ay)}$$

$$= \frac{(r, Ar)}{(r, r)} = \mu(r).$$

This yields (4:11) and (4:13). Using (4:16) with $p=y$ and $q=r$ we find that

$$f(x)=(y,Ay)=(y,r)\leqq |y|\ |r|.$$

Hence

$$f(x)=\mu(y)|y|^2\leqq |y|\ |r|,$$

so that the second inquality in (4:14) holds. The first inequality is obtained from the relations

$$\frac{|r|^2}{\mu(r)}\leqq f(x)\leqq |y|\ |r|.$$

As is to be expected, any cd-method has within its routine a determination of the inverse A^{-1} of A. We have, in fact, the following:

Theorem 4:5. *Let $p_0, \ldots, p_{n-1}$ be n mutually conjugate nonzero vectors and let p_{ij} be the j-th component of p_i. The element in the j-th row and k-th column of A^{-1} is given by the sum*

$$\sum_{i=0}^{n-1}\frac{p_{ij}p_{ik}}{(p_i,Ap_i)}.$$

This result follows from the formula

$$h=\sum_{i=0}^{n-1}\frac{(p_i,k)}{(p_i,Ap_i)}\,p_i$$

for the solution h of $Ax=k$, obtained by selecting $x_0=0$.

We conclude this section with the following:

Theorem 4:6. *Let π_i be the $(n-i)$-dimensional plane through x_i conjugate to the vectors $p_0, p_1, \ldots, p_{i-1}$. The plane π_i contains the points $x_i, x_{i+1}, \ldots$ and intersects the $(n-1)$-dimensional ellipsoid $f(x)=f(x_i)$ in an ellipsoid E_i' of dimension $(n-i-1)$. The center of E_i' is the solution h of $Ax=k$. The point x_{i+1} is the midpoint of the chord C_i of E_i' through x_i, which is parallel to p_i. In the cg-method the chord C_i is normal to E_i' at x_i and hence is in the direction of the gradient of $f(x)$ at x_i in π_i.*

The last statement will be established at the end of section 6. The equations of the plane π_i is given by the system

$$(Ap_j,x-x_i)=0 \qquad (j=0,1,\ldots,i-1).$$

Since $p_i,p_{i+1},\ldots$ are conjugate to $p_0,\ldots,p_{i-1}$, so also is

$$x_k-x_i=a_ip_i+\ldots+a_{k-1}p_{k-1} \qquad (k>i).$$

The points $x_i,x_{i+1},\ldots,x_m=h$ are accordingly in π_i, and h is the center of E_i'. The chord C_i is defined by the equation $x=x_i+ta_ip_i$, where t is a parameter. As is easily seen,

$$f(x_i+ta_ip_i)=f(x_i)-(2t-t^2)a_i^2(p_i,Ap_i).$$

The second endpoint of the chord C_i is the point $x_i+2a_ip_i$ at which $t=2$. The midpoint corresponds to $t=1$, and hence is the point x_{i+1} as was to be proved.

In view of theorem 4:6, it is seen that at each step of the cd-routine the dimensionality of the space π_i in which we seek the solution h is reduced by unity. Beginning with x_0, we select an arbitrary chord C_0 of $f(x)=f(x_0)$ through x_0 and find its center. The plane π_1 through x_1 conjugate to C_0 contains the centers of all chords parallel to C_0. In the next step we restrict ourselves to π_1 and select an arbitrary chord C_1 of $f(x)=f(x_1)$ through x_1 and find its midpoint x_2 and the plane π_2 in π_1 conjugate to C_1 (and hence to C_0). This process when repeated will yield the answer in at most n steps. In the cg-method the chord C_i of $f(x)=f(x_i)$ is chosen to be the normal at x_i.

5. Basic Relations in the cg-Method

Recall that in the cg-method the following formulas are used

$$p_0=r_0=k-Ax_0 \tag{5:1a}$$

$$a_i=\frac{|r_i|^2}{(p_i,Ap_i)} \tag{5:1b}$$

$$x_{i+1}=x_i+a_ip_i \tag{5:1c}$$

$$r_{i+1}=r_i-a_iAp_i \tag{5:1d}$$

$$b_i=\frac{|r_{i+1}|^2}{|r_i|^2} \tag{5:1e}$$

$$p_{i+1}=r_{i+1}+b_ip_i. \tag{5:1f}$$

One should verify *that eq.* (5:1e) *and* (5:1f) *hold for* $i=0,1,2,\ldots$ *if, and only if,*

$$p_k=|r_k|^2\sum_{j=0}^{k}\frac{r_j}{|r_j|^2} \qquad k=0,1,2,\ldots. \tag{5:2}$$

The present section is devoted to consequences of these formulas. As a first result, we have

Theorem 5:1. *The residuals $r_0, r_1, \ldots$ and the direction vectors $p_0, p_1, \ldots$ generated by (5:1) satisfy the relations*

$$(r_i,r_j)=0 \qquad (i\neq j) \tag{5:3a}$$

$$(p_i,Ap_j)=0 \qquad (i\neq j) \tag{5:3b}$$

$$(p_i,r_j)=0 \quad (i<j), \qquad (p_i,r_j)=|r_i|^2 \quad (i\geqq j) \tag{5:3c}$$

$$(r_i,Ap_i)=(p_i,Ap_i),\ (r_i,Ap_j)=0\ (i\neq j, i\neq j+1) \tag{5:3d}$$

The residuals $r_0, r_1, \ldots$ are mutually orthogonal and the direction vectors $p_0, p_1, \ldots$ are mutually conjugate.

The proof of this result will be made by induction. The vectors r_0, $p_0=r_0$ and r_1 satisfy these relations since

$$(r_0,r_1)=(p_0,r_1)=|r_0|^2-a_0(r_0,Ap_0)=0$$

by (5:1b). Suppose that (5:3) holds for the vectors $r_0, \ldots, r_k$ and $p_0, \ldots, p_{k-1}$. To show that p_k can be adjoined to this set it is necessary to show that

$$(r_i,p_k)=|r_k|^2 \qquad (i\leqq k) \qquad (5:4a)$$

$$(p_i,Ap_k)=0 \qquad (i<k) \qquad (5:4b)$$

$$(r_k,Ap_i)=(p_k,Ap_i) \qquad (i\leqq k, i\neq k-1) \qquad (5:4c)$$

Equation (5:4a) follows at once from (5:2) and (5:3a). To prove (5:4b) we use (5:1d) and find that

$$(r_{i+1},p_k)=(r_i,p_k)-a_i(Ap_i,p_k).$$

By (5:4a) this becomes

$$|r_k|^2=|r_k|^2-a_i(Ap_i,p_k) \qquad (i<k).$$

Since $a_i>0$, eq (5:4b) holds. In order to establish (5:4c), we use (5:1f) to obtain

$$(p_k,Ap_i)=(r_k,Ap_i)+b_{k-1}(p_{k-1},Ap_i)=(r_k,Ap_i) \qquad (i\neq k-1).$$

It follows that (5:4c) holds and hence that (5:3) holds for the vectors $r_0, r_1 \ldots, r_k$ and $p_0, p_1, \ldots, p_k$.

It remains to show that r_{k+1} can be adjoined to this set. This will be done by showing that

$$(r_i,r_{k+1})=0 \qquad (i\leqq k) \qquad (5:5a)$$

$$(Ap_i,r_{k+1})=0 \qquad (i<k) \qquad (5:5b)$$

$$(p_i,r_{k+1})=0 \qquad (i\leqq k). \qquad (5:5c)$$

By (5:1d) we have

$$(r_i,r_{k+1})=(r_i,r_k)-a_k(r_i,Ap_k).$$

If $i<k$, the terms on the right are zero and (5:5a) holds. If $i=k$, the right member is zero by (5:1b) and (5:3d). Using (5:1d) again we have with $i<k$

$$0=(r_{k+1},r_{i+1})=(r_{k+1},r_i)-a_i(r_{k+1},Ap_i)=-a_i(r_{k+1},Ap_i).$$

Hence (5:5b) holds. The equation (5:5c) follows from (5:5a) and the formula (5:2) for p_i.

As a consequence of (5:3b) we have the first two sentences of the following:

Theorem 5:2. *The cg-method is a cd-method. It is the special case of the cd-method in which the p_i are obtained by A-orthogonalization of the residual vectors r_i. On the other hand, a cd-method in which the residuals $r_0, r_1, \ldots$ are mutually orthogonal is essentially a cg-method.*

The term "essentially" is used to designate that we disregard iterations that terminate in fewer than n steps, unless one adds the natural assumption that the formula for p_i in the routine depends continuously on the initial estimate x_0. To prove this result we accordingly suppose that the routine terminates at the n-th step. Since the r_i is orthogonal to r_{i+1} we have $x_i\neq x_{i+1}$ and hence $a_i\neq 0$. It follows that $(p_i,r_i)\neq 0$ by (4:1a). We may accordingly suppose the vectors p_i have been normalized so that $(p_i,r_i)=|r_i|^2$. In view of (4:3b) and (4:3c) eq (5:3c) holds. Select numbers α_{ij} such that

$$p_i=\sum_{j=0}^{n-1}\alpha_{ij}r_j.$$

Taking the inner product of p_i with r_j it is seen by (5:3c) that

$$\alpha_{ij}=\frac{|r_i|^2}{|r_j|^2}\,(i\geqq j), \quad \alpha_{ij}=0 \qquad (<j).$$

Consequently, (5:2) holds and the theorem is established.

Theorem 5:3. *The residual vectors $r_0, r_1, \ldots$ and the direction vectors $p_0, p_1, \ldots$ satisfy the further relations*

$$(p_i,p_j)=\frac{|r_j|^2|p_i|^2}{|r_i|^2} \qquad (i\leqq j) \qquad (5:6a)$$

$$|p_i|^2=|r_i|^2+b_{i-1}^2|p_{i-1}|^2=|r_i|^4\sum_{j=0}^{i}\frac{1}{|r_j|^2} \qquad (i>0) \qquad (5:6b)$$

$$(r_i,Ar_j)=0 \qquad |i-j|>1 \qquad (5:6c)$$

$$(r_i,Ar_i)=(p_i,Ap_i)+b_{i-1}^2(p_{i-1},Ap_{i-1}) \qquad (i>0). \qquad (5:6d)$$

The vector r_i is shorter than p_i. The vector p_i makes an acute angle with p_j.

The relations (5:6a) and (5:6b) follow readily from (5:1e), (5:1f), (5:2), and (5:3). Using (5:1f) and (5:3d), we see that

$$(r_i,Ar_j)=(r_i,Ap_j)-b_{j-1}(r_i,Ap_{j-1})=0 \qquad (i<j-1).$$

Hence (5:6c) holds. Equation (5:6d) is a consequence of (5:1f) and (5:3b). The final statements are interpretations of formula (5:6b) and (5:6a).

Theorem 5:4. *The direction vectors $p_0, p_1, \ldots$ satisfy the relations*

$$p_1=(1+b_0)p_0-a_0Ap_0 \qquad (5:7a)$$

$$p_{i+1}=(1+b_i)p_i-a_iAp_i-b_{i-1}p_{i-1} \qquad (i>0). \qquad (5:7b)$$

Similarly, the residuals $r_0, r_1, \ldots$ satisfy the relations

$$r_1=r_0-a_0Ar_0 \qquad (5:8a)$$

$$r_{i+1}=(1+b'_{i-1})r_i-a_iAr_i-b'_{i-1}r_{i-1}, \qquad (5:8b)$$

where

$$b_{i-1}=\frac{a_i}{a_{i-1}}\,b_{i-1}. \qquad (5:9)$$

Equation (5:7b) is obtained by eliminating r_{i+1} and r_i from the equations

$$p_{i+1}=r_{i+1}+b_ip_i$$

$$r_{i+1}=r_i-a_iAp_i$$

$$p_i=r_i+b_{i-1}p_{i-1}.$$

Equation (5:7a) follows similarly. In order to prove (5:8b), eliminate Ap_i and Ap_{i-1} from the equations

$$r_{i+1}=r_i-a_iAp_i$$

$$Ap_i=Ar_i+b_{i-1}Ap_{i-1}$$

$$r_i=r_{i-1}-a_{i-1}Ap_{i-1}.$$

Equation (5:8a) holds since $p_0=r_0$.

Theorem 5:5. *The scalars a_i and b_i are given by the several formulas*

$$a_i=\frac{|r_i|^2}{(p_i,Ap_i)}=\frac{(p_i,r_i)}{(p_i,Ap_i)}=\frac{(p_i,r_0)}{(p_i,Ap_i)} \qquad (5:10)$$

$$b_i=\frac{|r_{i+1}|^2}{|r_i|^2}=-\frac{(r_{i+1},Ap_i)}{(p_i,Ap_i)}=-\frac{(r_{i+1},Ar_i)}{(p_i,Ap_i)}. \qquad (5:11)$$

The scalar a_i satisfies the relation

$$\frac{1}{a_0}=\mu(r_0), \qquad \mu(p_i)<\frac{1}{a_i}<\mu(r_i) \quad (i>0), \qquad (5:12)$$

where $\mu(z)$ is the Rayleigh quotient (4:12). *The reciprocal of a_i lies between the smallest and largest characteristic roots of A.*

The formula (5:10) follows from (5:1b) and (5:3c), while (5:11) follows from (5:1e), (5:1f), (5:3b), and (5:3d). Since

$$|r_i|<|p_i|, \qquad (r_i,Ar_i)>(p_i,Ap_i)$$

by (5:6b) and (5:6d), we have

$$\frac{(p_i,Ap_i)}{|p_i|^2}<\frac{(p_i,Ap_i)}{|r_i|^2}<\frac{(r_i,Ar_i)}{|r_i|^2}.$$

The inequalities (5:12) accordingly hold. The last statement is immediate, since $\mu(z)$ lies between the smallest and largest characteristic roots of A.

6. Properties of the Estimates x_i of h in the cg-Method

Let now $x_0, x_1, \ldots, x_m=h$ be the estimates of h obtained by applying the cg-method. Let $r_0, r_1, \ldots, r_m=0$ be the corresponding residuals and $p_0, p_1, \ldots, p_{m-1}$ the direction vectors used. The present section will be devoted to the study of the properties of the points $x_0, x_1, \ldots, x_m$. As a first result we have

Theorem 6:1. *The estimates $x_0, x_1, \cdots, x_m$ of h are distinct. The point x_i minimizes the error function $f(x)=(h-x,A(h-x))$ on the i-dimensional plane P_i passing through the points $x_0, x_1, \cdots, x_i$. In the* ith *step of the cg-method, $f(x)$ is diminished by the amount*

$$f(x_{i-1})-f(x_i)=a_{i-1}|r_{i-1}|^2=\mu(p_{i-1})|x_{i-1}-x_i|^2, \quad (6:1)$$

where $\mu(z)$ is the Rayleigh quotient (4:12). *Hence,*

$$f(x_i)-f(x_j)=a_i|r_i|^2+\cdots+a_{j-1}|r_{j-1}|^2 \qquad (i<j). \qquad (6:2)$$

The point x_i is given by the formulas

$$x_i=x_0+\sum_{j=0}^{i-1}a_jp_j=x_0+\sum_{j=0}^{i-1}\frac{f(x_j)-f(x_i)}{|r_j|^2}r_j. \qquad (6:3)$$

This result is essentially a restatement of theorem 4:3. The formula (6:3) follows from (5:2) and (6:2). The second equation in (6:1) is readily verified.

Theorem 6:2. *Let S_i be the convex closure of the estimates $x_0, x_1, \cdots, x_i$. The point x_i is the point in S_i, whose error vector $h-x$ is the shortest.*

For a point $x\neq x_i$ in S_i is expressible in the form

$$x=\alpha_0x_0+\cdots+\alpha_ix_i,$$

where $\alpha_l\geqq 0$, $\alpha_0+\alpha_1+\cdots+\alpha_i=1$.

We have accordingly

$$x_i-x=\alpha_0(x_i-x_0)+\cdots+\alpha_{i-1}(x_i-x_{i-1})=\beta_0p_0 + \cdots+\beta_{i-1}p_{i-1},$$

where the β's are nonnegative. Inasmuch as all $(p_j,p_k)>0$ it follows that

$$(x_j-x_i,\ x_i-x)>0 \qquad (i<j).$$

Using the relation

$$|x_j-x|^2=|x_j-x_i|^2+2(x_j-x_i,\ x_i-x)+|x_i-x|^2 \quad (6:4)$$

we find that

$$|x_j-x_i|<|x_j-x| \qquad (i<j).$$

Setting $j=m$, we obtain theorem 6:2.

Incidentally, we have established the

Corollary. The point x_i is the point in S_i nearest to the point x_j $(j>i)$.

Theorem 6:3. *At each step of the cg-algorithm the error vector $y_i=h-x_i$ is reduced in length. In fact,*

$$|y_{i-1}|^2-|y_i|^2=\frac{f(x_i)+f(x_{i-1})}{\mu(p_{i-1})}, \qquad (6:5)$$

where $\mu(z)$ is the Rayleigh quotient (4:12).

In order to establish (6:5) observe that, by (5:6a),

$$(y_i, x_i - x_{i-1}) = (x_m - x_i, p_{i-1})\, a_{i-1}$$
$$= [a_i(p_i, p_{i-1}) + \ldots + a_{m-1}(p_{m-1}, p_{i-1})]\, a_{i-1}$$
$$= [a_i |r_i|^2 + \ldots + a_{m-1}|r_{m-1}|^2]\, \frac{a_{i-1}|p_{i-1}|^2}{|r_{i-1}|^2}.$$

In view of (6:2) and (5:1b) this becomes

$$(y_i, x_i - x_{i-1}) = \frac{f(x_i)}{\mu(p_{i-1})}. \tag{6:6}$$

Setting $x = x_{i-1}$ and $j = m$ in (6:4), we obtain (6:5) by the use of (6:6) and (6:1).

This result establishes the cg-method as a method of successive approximations and justifies the procedure of stopping the algorithm before the final-step is reached. If this is done, the estimate obtained can be improved by using the results given in the next two theorems.

Theorem 6:4. *Let $x_{i+1}^{(i)}, \ldots, x_m^{(i)}$ be the projections of the points $x_{i+1}, \ldots, x_m = h$ in the i-dimensional plane P_i passing through the points $x_0, x_1, \ldots, x_i$. The points $x_{i-1}, x_i, x_{i+1}^{(i)}, \ldots, x_m^{(i)}$ lie on a straight line in the order given by their enumeration. The point $x_k^{(i)}$ $(k > i)$ is given by the formulas*

$$x_k^{(i)} = x_{i-1} + \frac{f(x_{i-1}) - f(x_k)}{f(x_{i-1}) - f(x_i)}(x_i - x_{i-1}), \tag{6:7a}$$

$$x_k^{(i)} = x_{i-1} + \frac{f(x_i) - f(x_k)}{|r_{i-1}|^2}\, p_{i-1}. \tag{6:7b}$$

In order to prove this result, it is sufficient to establish (6:7). To this end observe first that the vector

$$p_j - \frac{|r_j|^2}{|r_{i-1}|^2}\, p_{i-1} \qquad (j \geqq i)$$

is orthogonal to each of the vectors $p_0, p_1, \ldots, p_{i-1}$. This can be seen by forming the inner product with $p_l (l < i)$, and using (5:6a). The result is

$$(p_l, p_j) - \frac{|r_j|^2}{|r_{i-1}|^2}(p_l, p_{i-1}) = \frac{|p_l|^2}{|r_l|^2}[|r_j|^2 - |r_j|^2] = 0$$

The projection of the point

$$x_k = x_{i-1} + a_{i-1}p_{i-1} + a_i p_i + \ldots + a_{k-1}p_{k-1}$$

in P_i is accordingly

$$x_k^{(i)} = x_{i-1} + \frac{a_{i-1}|r_{i-1}|^2 + \ldots + a_{k-1}|r_{k-1}|^2}{|r_{i-1}|^2}\, p_{i-1}.$$

Using (6:2), we obtain (6:7). The points lie in the designated order, since $f(x_k) > f(x_{k+1})$.

Since $f(x_m) = 0$, we have the first part of

Theorem 6:5. *The point*

$$x_m^{(i)} = x_i + \frac{f(x_i)}{|r_{i-1}|^2}\, p_{i-1} \tag{6:8}$$

is the point in P_i whose distance from the solution h is the least. It lies on the line $x_{i-1}x_i$ beyond x_i. Moreover,

$$\frac{1}{f(x_m^{(i)})} = \frac{1}{f(x_i)} - \frac{1}{f(x_{i-1})} \tag{6.9}$$

and

$$|h - x_i|^2 = |h - x_m^{(i)}|^2 + \frac{f(x_m^{(i)}) - f(x_i)}{\mu(p_{i-1})}. \tag{6.10}$$

In order to establish (6:9) and 6:10) we use the formula

$$f(x_i + \alpha p_{i-1}) = f(x_i) + \alpha^2(p_{i-1}, Ap_{i-1}),$$

which holds for all values of α in view of the fact that x_i minimizes $f(x)$ on P_i. Setting $\alpha = f(x_i)/|r_{i-1}|^2$ we have

$$f(x_m^{(i)}) = f(x_1) + \frac{f(x_i)^2}{a_{i-1}|r_{i-1}|^2} = f(x_i)^2 + \frac{f(x_i)^2}{f(x_{i-1}) - f(x_i)}.$$

An algebraic reduction yields (6:9). Inasmuch as

$$|h - x_i|^2 - |h - x_m^{(i)}|^2 = \frac{f(x_i)^2 |p_{i-1}|^2}{|r_{i-1}|^4}$$
$$= \frac{f(x_i)^2}{f(x_{i-1}) - f(x_i)}\, \frac{a_{i-1}|p_{i-1}|^2}{|r_{i-1}|^2},$$

we obtain (6:10) from (6:9) and (5:1b).

As a further result we have

Theorem 6:6. *Let $x_1', \ldots, x_{m-1}'$ be the projections of the points $x_1, \ldots, x_{m-1}'$ on the line joining the initial point x_0 to the solution $x_m = h$. The points $x_0, x_1', \ldots, x_{m-1}', x_m = h$ lie in the order of enumeration.*

Thus, it is seen that we proceed towards the solution without oscillation. To prove this fact we need only observe that

$$(x_m - x_0, x_i - x_{i-1}) = (x_m - x_0, a_{i-1}, p_{i-1})$$
$$= a_{i-1} \sum_{j=0}^{m-1} a_j(p_j, p_{i-1}) > 0$$

by (5:6a). A similar result holds for the line joining x_i to $x_j (i < j)$.

Let π_i be the $(n-i)$-dimensional plane through x_i conjugate to $p_0, p_1, \ldots, p_{i-1}$. It consists of the set of points x satisfying the equation

$$(Ap_j, x - x_i) = 0 \qquad (j = 0, 1, \ldots, i-1).$$

This plane contains the points $x_{i+1}, \ldots, x_m$ and hence the solution h.

Theorem 6:7. *The gradient of the function $f(x)$ at x_i in the plane π_i is a scalar multiple of the vector p_i.*

The gradient of $f(x)$ at x_i is the vector $-r_i$. The gradient q_i of $f(x)$ at x_i in π_i is the orthogonal projection of $-r_i$ in the plane π_i. Hence q_i is of the form

$$q_i = -r_i + \alpha_0 Ap_0 + \ldots + \alpha_{i-1}Ap_{i-1},$$

where $\alpha_0, \ldots, \alpha_{i-1}$ are chosen so that q_i is orthogonal to $Ap_0, \ldots, Ap_{i-1}$. Since

$$p_i = |r_i|^2 \sum_{j=0}^{i} \frac{r_j}{|r_j|^2}$$

$$r_{j+1} = r_j - a_j A p_j \qquad (j=0,1,\ldots,i-1),$$

it is seen upon elimination of $r_0, r_1, \ldots, r_{i-1}$ successively that p_i is also a linear combination of $r_i, Ap_0, \ldots, Ap_{i-1}$. Inasmuch as p_i is conjugate to $p_0, \ldots, p_{i-1}$, it is orthogonal to $Ap_0, \ldots, Ap_{i-1}$. The vector p_i accordingly is a scalar multiple of the gradient q_i of $f(x)$ at x_i in π_i, as was to be proved.

In view of the result obtained in theorem 6:7 it is seen that the name "method of conjugate gradients" is an appropriate name for the method given in section 3. In the first step the relaxation is made in the direction p_0 of the gradient of $f(x)$ at x_0, obtaining a minimum value of $f(x)$ at x_1. Since the solution h lies in π_1, it is sufficient to restrict x to the plane π_1. Accordingly, in the next step, we relax in the direction p_1 of the gradient of $f(x)$ in π_1 at x_1, obtaining the point x_2 at which $f(x)$ is least. The problem is then reduced to relaxing $f(x)$ in the plane π_2, conjugate to p_0 and p_1. At the next step the gradient in x_2 in π_2 is used, and so on. The dimensionality of the space in which the relaxation is to take place is reduced by unity at each step. Accordingly, after at most n steps, the desired solution is attained.

7. Properties of the Estimates $\bar{x}_i$ of h in the cg-Method

In the cg-method there is a second set of estimates $\bar{x}_0 = x_0, \bar{x}_1, \bar{x}_2, \ldots$ of h that can be computed, and that are of significance in application to linear systems arising from difference equations approximating boundary-valve problems. In these applications, the function defined by $\bar{x}_i$ is smoother than that of x_i, and from this point of view is a better approximation of the solution h. The point $\bar{x}_i$ has its residual proportional to the conjugate gradient p_i. The points $\bar{x}_0, \bar{x}_1, \bar{x}_2, \ldots$ can be computed by the iteration (7:2) given in the following:

Theorem 7:1. *The conjugate gradient p_i is expressible in the form*

$$p_i = c_i(k - A\bar{x}_i), \tag{7:1}$$

where c_i and $\bar{x}_i$ are defined by the recursion formulas

$$c_0 = 1, \; c_{i+1} = 1 + b_i c_i \tag{7:2a}$$

$$\bar{x}_0 = x_0, \; \bar{x}_{i+1} = \frac{x_{i+1} + b_i c_i \bar{x}_i}{c_{i+1}}. \tag{7:2b}$$

We have the relations

$$c_i = |r_i|^2 \sum_{j=0}^{i} \frac{1}{|r_j|^2} = \frac{|p_i|^2}{|r_i|^2} \tag{7:3a}$$

$$\bar{x}_i = \frac{|r_i|^2}{c_i} \sum_{j=0}^{i} \frac{x_j}{|r_j|^2} \tag{7:3b}$$

$$\bar{r}_i = k - A\bar{x}_i = \frac{1}{c_i} p_i = \frac{|r_i|^2}{c_i} \sum_{j=0}^{i} \frac{r_j}{|r_j|^2}. \tag{7:3c}$$

The sum of the coefficients of $x_0, x_1, \ldots, x_i$ in (7:3b) (and hence of $r_0, r_1, \ldots, r_i$ in (7:3c)) is unity.

The relation (7:1) can be established by induction. It holds for $i=0$. If it holds for i, then

$$p_{i+1} = r_{i+1} + b_i p_i = (1 + b_i c_i)k - A(x_{i+1} + b_i c_i \bar{x}_i)$$
$$= c_{i+1}(k - A\bar{x}_{i+1}).$$

The formula (7:3a) follows from (7:2a), (5:1c) and (5:6b). Formula (7:3b) is an easy consequence of (7:2b). To prove (7:3c) one can use (5:2) or (7:3b), as one wishes. The final statement is a consequence of (7:3a).

Theorem 7:2. *The point $\bar{x}_i$ given by (7:2) lies in the convex closure S_i of the points $x_0, x_1, \ldots, x_i$. It is the point x in the i-dimensional plane P_i through $x_0, x_1, \ldots, x_i$ at which the squared residual $|k - Ax|^2$ has its minimum value. This minimum value is given by the formula*

$$|\bar{r}_i|^2 = \frac{|r_i|^2}{c_i} = \frac{|r_i|^4}{|p_i|^2}. \tag{7:4}$$

The squared residuals $|r_0|^2, |\bar{r}_i|^2, \ldots$ diminish monotonically during the cg-method. At the ith step the squared residual is reduced by the amount

$$|\bar{r}_{i-1}|^2 - |\bar{r}_i|^2 = \frac{|\bar{r}_{i-1}|^2}{c_i}. \tag{7:5}$$

The first statement follows from (7:3b), since the coefficients of $x_0, x_1, \ldots, x_i$ are positive and have unity as their sum. In order to show that the squared residual has a minimum on P_i at $\bar{x}_i$, observe that a point x in P_i differs from $\bar{x}_i$ by a vector z_i of the form

$$x - \bar{x}_i = z_i = \alpha_0 p_0 + \cdots + \alpha_{i-1} p_{i-1}.$$

The residual $r = k - Ax$ is accordingly given by

$$r = \bar{r}_i - Az_i$$

$$Az_i = \alpha_0 Ap_0 + \cdots - \alpha_{i-1} Ap_{i-1}.$$

Inasmuch as, by (7:3c), $\bar{r}_i = p_i/c_i$, we have

$$(\bar{r}_i, Ap_j) = \frac{1}{c_i}(p_i, Ap_j) = 0 \qquad (j < i).$$

Consequently, $(\bar{r}_i, Az_i) = 0$ and

$$|r|^2 = |\bar{r}_i|^2 + |Az_i|^2 > |\bar{r}_i|^2 \qquad (x \neq \bar{x}_i).$$

It follows that $\bar{x}_i$ affords a proper minimum to $|r|^2$ on P_i. Using (7:3c) and (7:3a) and the orthogonality of r_j's, it is seen that the minimum value of $|r|^2$ on P_i is given by (7:4). By (7:4) and (7:2a)

$$|\bar{r}_{i-1}|^2-|\bar{r}_i|^2=|\bar{r}_{i-1}|^2\left(1-\frac{c_{i-1}}{c_i}\frac{|r_i|^2}{|r_{i-1}|^2}\right)$$
$$=|\bar{r}_{i-1}|^2\left(1-\frac{b_{i-1}c_{i-1}}{c_i}\right)=\frac{|\bar{r}_{i-1}|^2}{c_i}.$$

This completes the proof of theorem 7:1.

Theorem 7:3. *The Rayleigh quotients of* r_0, r_1, . . . *and* $\bar{r}_0$, $\bar{r}_1$, . . . *are connected by the formulas*

$$\frac{\mu(r_i)}{|r_i|^2}=\frac{\mu(\bar{r}_i)}{|\bar{r}_i|^2}+\frac{\mu(\bar{r}_{i-1})}{|\bar{r}_{i-1}|^2} \tag{7:6a}$$

$$\frac{\mu(\bar{r}_i)}{|\bar{r}_i|^2}=\frac{\mu(r_i)}{|r_i|^2}-\frac{\mu(r_{i-1})}{|r_{i-1}|^2}+\cdots+(-1)^i\frac{\mu(r_0)}{|r_0|^2}. \tag{7:6b}$$

The Rayleigh quotient of $\bar{r}_i(i>0)$ *is smaller than that of* r_i, *that is,* $\mu(\bar{r}_i)<\mu(r_i)$.

In order to prove this result we use (5:6d) and obtain

$$\frac{(r_i,Ar_i)}{|r_i|^4}=\frac{(p_i,Ap_i)}{|r_i|^4}+\frac{(p_{i-1},Ap_{i-1})}{|r_{i-1}|^4}.$$

Since $|r_i|^4=|p_i|^2|\bar{r}_i|^2$ and $\mu(p_i)=\mu(\bar{r}_i)$, this relation yields (7:6a). The eq (7:6b) follows from (7:6a). The last statement follows from (5:12).

In the applications to linear systems arising from difference equations approximating boundary value problems, $\mu(r_i)$ can be taken as a measure of the smoothness of x_i. The smaller $\mu(r_i)$ is, the smoother x_i is. *Hence* $\bar{x}_i$ *is smoother than* x_i.

Theorem 7:4. *At the point* $\bar{x}_i$ *the error function* $f(x)$ *has the value*

$$f(\bar{x}_i)=f(x_i)+|\bar{r}_i|^4\sum_{j=0}^{i-1}\frac{f(x_j)-f(x_{j+1})}{|\bar{r}_j|^4} \tag{7:7}$$

and we have

$$f(x_i)<f(\bar{x}_i)<f(\bar{x}_{i-1}). \tag{7:8}$$

The sequence $f(\bar{x}_0)$, $f(\bar{x}_1)$, $f(\bar{x}_2)$, . . . *decreases monotonically.*

In order to prove this result, it is convenient to set

$$\bar{b}_{i-1}=\frac{|\bar{r}_i|^2}{|\bar{r}_{i-1}|^2}=\frac{c_{i-1}|r_i|^2}{c_i|r_{i-1}|^2}=\frac{b_{i-1}c_{i-1}}{c_i}. \tag{7:9}$$

By (7:2) we have the relation

$$\bar{x}_i-x_i=\bar{b}_{i-1}(\bar{x}_{i-1}-x_i). \tag{7:10}$$

Using the formula

$$f(x)-f(x_i)=(x-x_i,A(x-x_i)),$$

which holds for any x in P_i, we see that

$$f(\bar{x}_i)-f(x_i)=\bar{b}_{i-1}^2(\bar{x}_{i-1}-x_i,A(\bar{x}_{i-1}-x_i)),$$

that is,

$$f(\bar{x}_i)-f(x_i)=\bar{b}_{i-1}^2[f(\bar{x}_{i-1})-f(x_i)]. \tag{7:11}$$

By (7:9) it is seen that this result can be put in the form

$$\frac{f(\bar{x}_i)-f(x_i)}{|\bar{r}_i|^4}=\frac{f(x_{i-1})-f(x_i)}{|\bar{r}_{i-1}|^4}+\frac{f(\bar{x}_{i-1})-f(x_{i-1})}{|\bar{r}_{i-1}|^4}.$$

Since $\bar{x}_0=x_0$ this formula yields the desired relation (7:7). Since $\bar{b}_{i-1}<1$, it follows from (7:11) and (7:7) that (7:8) holds. This proves the theorem.

Theorem 7:5. *The error vector* $y_i=h-x_i$ *is shorter than the error vector* $\bar{y}_i=h-\bar{x}_i$. *Moreover,* $\bar{y}_i$ *is shorter than* y_{i-1}.

The first statement follows from (7:2). It also follows from theorem 6:2, since $\bar{x}_i$ is in S_i. By (7:2) the point $\bar{x}_i$ lies in the line segment joining $\bar{x}_i$ to $\bar{x}_{i-1}$. The distance from h to $\bar{x}_i$ exceeds the distance from h to $\bar{x}_i$. It follows that as we move from $\bar{x}_i$ to $\bar{x}_{i-1}$ the distance from h is increased, as was to be proved.

8. Propagation of Rounding-Off Errors in the cg-Method

In this section we take as basic relations between the vectors r_0, r_1, · · · and p_0, p_1, · · · in the cg-method the following:

$$p_0=r_0, \tag{8:1a}$$

$$a_i=\frac{|r_i|^2}{(p_i,Ap_i)} \tag{8:1b}$$

$$r_{i+1}=r_i-a_iAp_i \tag{8:1c}$$

$$b_1=\frac{|r_{i+1}|^2}{|r_i|^2} \tag{8:1d}$$

$$p_{i+1}=r_{i+1}+b_ip_i. \tag{8:1e}$$

As a consequence, we have the orthogonality relations

$$(r_i,r_k)=0,\quad (Ap_i,p_k)=0\qquad (i\neq k). \tag{8:2}$$

Because of rounding-off errors during a numerical calculation (routine), these relations will not be satisfied exactly. As the difference $|k-i|$ increases, the error in (8:2) may increase so rapidly that x_n will not be as good an estimate of h as desired. This error can be lessened in two ways: first, by introducing a subsidiary calculation to reduce rounding-off errors; and second, by repeating the iteration so as to obtain a new estimate. This section will be concerned with the first of these and with a study of the propagation of the rounding-off errors. To this end it is convenient to divide the section in four parts, the first of which is the following:

8.1. Basic propagation formulas

In this part we derive simple formulas showing how errors in scalar products of the type

$$(r_{i-1},r_i), \qquad (Ap_{i-1},p_i) \tag{8:3}$$

are propagated during the next step of the computation. From (8:1e) follows

$$(r_i,r_{i+1})=(p_i,r_{i+1})-b_{i-1}(p_{i-1},r_{i+1}).$$

Inserting (8:1c) in both terms on the right yields

$$(r_i,r_{i+1})=(p_i,r_i)-a_i(p_i,Ap_i)-b_{i-1}(p_{i-1},r_i)+b_{i-1}a_i(Ap_{i-1},p_i).$$

Applying (8:1e) to the first and third terms gives

$$(r_i,r_{i+1})=(r_i,r_i)-a_i(p_i,Ap_i)+b_{i-1}a_i(Ap_{i-1},p_i), \tag{8:4}$$

which by (8:1b) becomes

$$(r_i,r_{i+1})=b_{i-1}a_i(Ap_{i-1},p_i). \tag{8:5}$$

This is our first propagation formula.

Using (8:1e) again,

$$(Ap_i,p_{i+1})=Ap_i,r_{i+1})+b_i(Ap_i,p_i).$$

Inserting (7:1c) in the first term,

$$(Ap_i,p_{i+1})=-\frac{1}{a_i}|r_{i+1}|^2+\frac{1}{a_i}(r_i,r_{i+1})+b_i(Ap_i,p_i). \tag{8:6}$$

But in view of (8:1b) and (8:1d)

$$|r_{i+1}|^2=a_ib_i(Ap_i,p_i). \tag{8:7}$$

Therefore,

$$(Ap_i,p_{i+1})=\frac{1}{a_i}(r_i,r_{i+1}). \tag{8:8}$$

This is our *second propagation formula.*

Putting (8:5) and (8:8) together yields the *third and fourth propagation formulas*

$$(r_i,r_{i+1})=\frac{b_{i+1}a_i}{a_{i-1}}(r_{i-1},r_i) \tag{8:9a}$$

$$(Ap_i,p_{i+1})=b_{i-1}(Ap_{i-1},p_i), \tag{8:9b}$$

which can be written in the alternate form

$$\frac{(r_i,r_{i+1})}{|r_i|^2}=\frac{a_i}{a_{i-1}}\frac{(r_{i-1},r_i)}{|r_{i-1}|^2} \tag{8:10a}$$

$$\frac{(Ap_i,p_{i+1})}{(Ap_i,p_i)}=\frac{a_i}{a_{i-1}}\frac{(Ap_{i-1},p_i)}{(Ap_{i-1},p_{i-1})} \tag{8:10b}$$

by virtue of (8:1b) and (8:1d). Each of these propagation formulas, and in particular the simple formulas (8:9), can be used to check whether nonvanishing products (8:3) are due to normal rounding-off errors or to errors of the computer. The formulas (8:10) have the following meaning. If we build the symmetric matrix P having the elements (Ap_i,p_k), the left side of (8:10b) is the ratio of two consecutive elements in the same line, one located in the main diagonal and one on its right hand side. The formula (8:10b) gives the change of this ratio as we go down the main diagonal.

8.2. A Stability Condition

Even if the scalar products (8:2) are not all zero, so that the vectors $p_0, p_1, \cdots, p_{n-1}$ are not exactly conjugate, we may use these vectors for solving $Ax=k$ in the following way. The solution h may be written in the form

$$h=x_0+a_0'p_0+a_1'p_1+\cdots+a_{n-1}'p_{n-1}. \tag{8:11}$$

Taking the scalar product with Ap_i, we obtain

$$(x_0,Ap_i)+\sum_k(Ap_i,p_k)a_k'=(h,Ap_i)=(Ah,p_i)=(k,p_i)$$

or

$$\sum_k(Ap_i,p_k)a_k'=(r_0,p_i). \tag{8:12}$$

The system $Ax=k$ may be replaced by this linear system for $a_0',\cdots,a_{n-1}'$. Therefore, because of rounding-off errors we have certainly not solved the given system exactly, but we have reached a more modest goal, namely, we have transformed the given system into the system (8:12), which has a dominating main diagonal if rounding-off errors have not accumulated too fast. The cg-algorithm gives an approximate solution

$$h'=x_0+a_0p_0+\cdots+a_{n-1}p_{n-1}. \tag{8:13}$$

A comparison of (8:11) and (8:13) shows that the number a_k computed during the cg-process is an approximate value of a_k'.

In order to have a dominating main diagonal in the matrix of the system (8:12) the quotients

$$\frac{(Ap_i,p_k)}{(Ap_i,p_i)} \qquad (i\neq k) \tag{8:14}$$

must be small. In particular this must be true for $k=i+1$. In this special case we learn from (8:10b) that increasing numbers $a_0, a_1,\cdots$ during the cg-process lead to accumulation of rounding-off errors, because then these quotients increase also. We have accordingly the following stability condition.

The larger the ratios a_i/a_{i-1}, the more rapidly the rounding-off errors accumulate.

A more elaborate discussion of the general quotient (8:14) gives essentially the same result.

By theorem 5:5, the scalars a_i lie on the range

$$\frac{1}{\lambda_{max}}<a_i<\frac{1}{\lambda_{min}},$$

where λ_{min}, λ_{max} are the least and the largest eigenvalues of A. Accordingly, the ratio $\rho=\lambda_{max}/\lambda_{min}$ is an upper bound of the critical ratio a_i/a_{i-1}, which determines the stability of the process. When ρ is near one, that is, when A is near a multiple of the identity, the cg-method is relatively stable. It will be shown in section 18 that examples can be constructed in which the ratios a_i/a_{i-1} $(i=1,\cdots,n-1)$ are any set of preassigned positive numbers. Thus the stability may be low. However, this instability can be compensated to a certain extent by starting the cg-process with a vector x_0 whose residual vector r_0 is near to the eigenvector of A corresponding to λ_{min}. In this event a_0 is near to the upper bound $1/\lambda_{min}$ of the a_i. This result is brought out in the following theorem:

For a given symmetric and positive definite matrix A, which has distinct eigenvalues, there exists always an initial residual vector r_0 such that $(a_i/a_{i-1})<1$ and hence such that the algorithm is stable with respect to the propagation of rounding-off errors.

In order to prove this we introduce the eigenvalues

$$\lambda_{min}=\lambda_0<\lambda_1<\lambda_2<\ldots<\lambda_{n-1}=\lambda_{max}$$

of A, and we take the corresponding normalized eigenvectors as a coordinate system. Let α_0, α_1, ..., α_{n-1} be real numbers not equal to zero and ϵ a small quantity. Then we start with a residual vector

$$r_0=(\alpha_0,\alpha_1\epsilon,\alpha_2\epsilon^2,\ldots,\alpha_{n-1}\epsilon^{n-1}). \qquad (8:14a)$$

Expanding everything in a power series, one finds that

$$a_i=\frac{1}{\lambda_i}+\epsilon^2(*). \qquad (8:14b)$$

Hence

$$\frac{a_i}{a_{i-1}}=\frac{\lambda_{i-1}}{\lambda_i}+\epsilon^2(*)<1$$

if ϵ is small enough.

As a by-product of such a choice of r_0 we get by (8:14b) approximations of the eigenvalues of A. Moreover, it turns out that in this case the successive residual-vectors r_0, r_1, ..., r_{n-1} are approximations of the eigenvectors.

These results suggest the following rule:

The cg-process should start with a smooth residual distribution, that is, one for which $\mu(r_0)$ is close to λ_{min}. If needed, the first estimate can be smoothed by some relaxation process.

Of course, we may use for this preparing relaxation the cg-process itself, computing the estimates $\bar{x}_i$ given in section 7. A simpler method is to modify the cg-process by setting $b_i=0$ so that $p_i=r_i$ and selecting a_i of the form $a_i=\alpha/\mu(r_i)$, where α is a small constant in the range $0<\alpha<1$.

8.3. The End-Correction

The system (8:12) can be solved by ordinary relaxation processes. Introducing the numbers a_k as approximations of the solutions a_k', we get the residuals

$$(r_0,p_i)-\sum_k(Ap_i,p_k)a_k. \qquad (8:15)$$

Inasmuch as $r_0=r_{i+1}+a_0Ap_0+\ldots+a_iAp_i$ by (8:1c), we have

$$(r_0,p_i)=(r_{i+1},p_i)+a_0(Ap_i,p_0)+\ldots+a_i(Ap_i,p_i). \qquad (8:16)$$

It follows that the residual (8:15) is reduced to

$$(r_{i+1},p_i)-(Ap_i,p_{i+1})a_{i+1}-(Ap_i,p_{i+2})a_{i+2}-\ldots$$
$$-(Ap_i,p_{n-1})a_{n-1}. \qquad (8:17)$$

This leads to the correction of a_i

$$\Delta a_i=\frac{1}{(Ap_i,p_i)}\{(r_{i+1},p_i)-(Ap_i,p_{i+1})a_{i+1}$$
$$-(Ap_i,p_{i+2})a_{i+2}-\ldots-(Ap_i,p_{n-1})a_{n-1}\}. \qquad (8:18)$$

A first approximation of a_i' is accordingly

$$a_i'\sim a_i+\Delta a_i.$$

In order to discuss this result, suppose that the numbers a_i have been computed accordingly to the formula

$$a_i=\frac{(p_i,r_i)}{(p_i,Ap_i)} \qquad (8:19)$$

(theorem 5:5). From (8:1c) it follows that $(r_{i+1},p_i)=0$, and therefore this term drops out in (8:18). In this case the correction Δa_i depends only on the products (Ap_i,p_k) with $i<k$. That is to say, that this correction is influenced only by the rounding-off errors *after* the i-th step. If, for instance, the rounding-off errors in the last 10 steps of a cg-process are small enough to be neglected, the last 10 values a_i need not to be corrected. Hence, generally, the Δa_i decrease rather rapidly.

From (8:18) we learn that in order to have a good rounding-off behavior, it is not only necessary to keep the products (p_k,Ap_i) $(i\neq k)$ small, but also to satisfy $(r_{i+1},p_i)=0$ as well as possible. Therefore, it may be better to compute the a_i from the formulas (8:19) rather than from (8:1b). We see this immediately, if we compare (8:19) with (8:1b); by (8:19) and (8:1e) we have

$$a_i=\frac{1}{(p_i,Ap_i)}\{|r_i|^2+b_{i-1}(r_i,p_{i-1})\}.$$

For ill-conditioned matrices, where a_i and b_i may become considerably larger than 1, the omitting of the second summand may cause additional errors. For the same reason, it is at least as important in

these cases to use formula (3:2b) rather than (8:1d) for determining b_i, since by (3:2b) and (8:1c)

$$b_i=\frac{1}{a_i(p_i,Ap_i)}\{|r_{i+1}|^2-(r_{i+1},r_i)\}.$$

Here the second summand is not directly made zero by any of the two sets of formulas for a_i and b_i. The only orthogonality relations, which are directly fulfilled in the scope of exactitude of the numerical computation by the choice of a_i and b_i, are the following:

$$(r_{i+1},p_i)=0, \quad (p_{i+1},Ap_i)=0.$$

Therefore, we have to represent (r_{i+1},r_i) in terms of these scalar products:

$$(r_{i+1},r_i)=(r_{i+1},p_i)-b_{i-1}(r_i,p_{i-1})+a_ib_{i-1}(p_i,Ap_{i-1}).$$

From this expression we see that for large b_i and a_i the second and third terms may cause considerable rounding-off errors, which affect also the relation $(p_{i+1},Ap_i)=0$, if we use formula (8:1d) for b_i. This is confirmed by our numerical experiments (section 19).

From a practical point of view, the following formula is more advantageous because it avoids the computation of all the products (Ap_i,p_k). From (8:1c) follows

$$r_n=r_{i+1}-a_{i+1}Ap_{i+1}-a_{i+2}Ap_{i+2}-\ldots-a_{n-1}Ap_{n-1}$$

$$(r_n,p_i)=(r_{i+1},p_i)-a_{i+1}(Ap_i,p_{i+1})$$
$$-\ldots-a_{n+1}(Ap_i,p_{n-1}).$$

and we have the result

$$\Delta a_i=\frac{(r_n,p_i)}{(Ap_i,p_i)}. \tag{8:20}$$

This formula gives corrections of the a_i if, because of rounding-off errors, the residual r_n is not small enough.

8.4. Refinement of the cg-algorithm

In order to diminish rounding-off errors in the orthogonality of the residuals r_i we refine our general routine (8:1). After the ith step in the routine we compute (Ap_{i-1},p_i), which should be small. Going then to the $(i+1)$st step we replace a_i by a slightly different quantity $\bar{a}_i$ chosen so that $(r_i,r_{i+1})=0$. In order to perform this, we may use (8:4), which now must be written

$$(r_i,r_{i+1})=(r_i,r_i)-\bar{a}_i(Ap_i,p_i)+b_{i-1}\bar{a}_i(Ap_{i-1},p_i)=0$$

yielding

$$\bar{a}_i=\frac{|r_i|^2}{(Ap_i,p_i)-b_{i-1}(Ap_{i-1},p_i)}.$$

Introducing the correction factor

$$d_i=1-b_{i-1}\frac{(Ap_{i-1},p_i)}{(Ap_i,p_i)} \tag{8:21}$$

and taking into account the old value (8:1b) of a_i, this can be written in the form

$$\bar{a}_i=\frac{a_i}{d_i}. \tag{8:22}$$

Continuing in the general routine of the $(i+1)$st step we replace b_i by a number $\bar{b}_i$ in such a way that $(Ap_i,p_{i+1})=0$. We use (8:6), which now must be written in the form

$$-|r_{i+1}|^2+(r_i,r_{i+1})+\bar{a}_i\bar{b}_i(Ap_i,p_i)=0.$$

The term (r_i,r_{i+1}) vanishes by virtue of our choice of $\bar{a}_i$. Using (8:7), we see that $a_ib_i=\bar{a}_i\bar{b}_i$ and from (8:22)

$$\bar{b}_i=b_id_i. \tag{8:23}$$

Since rounding-off errors occur again, this subroutine can be used in the same way to improve the results in the $(i+1)$th step.

The corrections just described can be incorporated automatically in the general routine by replacing the formulas (3:1) by the following refinement:

$$\begin{aligned}
&p_0=r_0=k-Ax_0, \qquad d_0=1\\
&a_i=\frac{|r_i|^2}{(p_i,Ap_i)}\frac{1}{d_i}\\
&x_{i+1}=x_i+a_ip_i\\
&r_{i+1}=r_i-a_iAp_i\\
&b_i=\frac{|r_{i+1}|^2}{|r_i|^2}d_i\\
&p_{i+1}=r_{i+1}+b_ip_i\\
&d_{i+1}=1-b_i\frac{(Ap_{i+1},p_i)}{(Ap_{i+1},p_{i+1})}.
\end{aligned} \tag{8:24}$$

Another quite obvious, but numerically more laborious method of refinement goes along the following lines. After finishing the ith step, compute a product of the type (Ap_i,p_k) with $k<i$. Then replace p_i by

$$\bar{p}_i=p_i-\frac{(Ap_i,p_k)}{(Ap_k,p_k)}p_k. \tag{8:25}$$

The vector p_i is exactly conjugate to p_k. This method may be used in place of (8:24) in case $k=i-1$. It has the disadvantage that a *vector* must be corrected and not just *numbers*, as in (8:24).

9. Modifications of the cg-Process

In the cg-method given by (3:1), the lengths of the vectors $p_0, p_1, \ldots$ are at our disposal. In order to preserve significant figures in computations, it is desirable to have all the p_i of about the same magnitude. In order to aim at this goal, we normalized the p's by the formulas

$$p_0=r_0, \quad p_i=r_i+b_{i-1}p_{i-1} \qquad (i>0).$$

However other normalizations can be made. In order to see how this normalization appears, we replace p_i by d_ip_i in eq (3:1), where d_i is a scalar factor. This factor d_i is not the same as that given in section 8 but plays a similar role. The result is

$$\begin{aligned} r_0&=k-Ax_0, \qquad p_0=\frac{r_0}{d_0}\\ x_{i+1}&=x_i+a_ip_i\\ r_{i+1}&=r_i-a_iAp_i\\ p_{i+1}&=\frac{r_{i+1}+b_ip_i}{d_{i+1}}\\ a_i&=\frac{|r_i|^2}{(p_i,Ap_i)d_i}=\frac{(p_i,r_i)}{(p_i,Ap_i)}\\ b_i&=\frac{|r_{i+1}|^2d_i}{|r_i|^2}=-\frac{(r_{i+1},Ap_i)}{(p_i,Ap_i)}. \end{aligned} \tag{9:1}$$

The connections between a_i, b_i, d_i are given by the equation

$$\begin{aligned} &\mu(r_0)-\frac{d_0}{a_0}=0\\ &\mu(r_i)-\frac{d_i}{a_i}=\frac{b_{i-1}}{a_{i-1}}=\frac{|r_i|^2}{|r_{i-1}|^2}\frac{d_{i-1}}{a_{i-1}} \qquad (i>0), \end{aligned} \tag{9:2}$$

where $\mu(r)$ is the Rayleigh quotient (4:12). In order to establish these relations we use the fact that r_i and r_{i+1} are orthogonal. This yields

$$\begin{aligned} |r_i|^2&=a_i(r_i,Ap_i)\\ |r_{i+1}|^2&=-a_i(r_{i+1},Ap_i) \end{aligned} \qquad (i\geqq 0)$$

by virtue of the formula $r_{i+1}=r_i-a_iAp_i$. From the connection between p_i and r_i, we find that

$$\begin{aligned} \frac{d_i}{a_i}|r_i|^2&=d_i(r_i,Ap_i)\\ &=(r_i,Ar_i)+b_{i-1}(r_i,Ap_{i-1})\\ &=(r_i,Ar_i)-\frac{b_{i-1}}{a_{i-1}}|r_i|^2. \end{aligned}$$

This yields (9:2) in case $i>0$. The formula, when $i=0$, follows similarly.

In the formulas (9:1) the scalar factor d_i is an arbitrary positive number determining the length of p_i. The case $d_i=1$ is discussed in sections 3 and 5. The following cases are of interest.

I. *The vector p_i can be chosen to be the residual vector $\bar{r}_i$ described in section 7.*

In this event we select

$$d_0=1, \qquad d_{i+1}=1+b_i. \tag{9:3}$$

The formula (7:2b) for $\bar{x}_{i+1}$ becomes

$$\bar{x}_{i+1}=\frac{x_{i+1}+b_i\bar{x}_i}{1+b_i}. \tag{9:4}$$

II. *The vector p_i can be chosen so that the formula*

$$p_i=\sum_{j=0}^{j}\frac{r_j}{|r_j|^2}$$

holds.

In this event the basic formulas (9:1) take the simple form

$$\begin{aligned} r_0&=k-Ax_0, \qquad p_0=\frac{r_0}{|r_0|^2}\\ x_{i+1}&=x_i+\frac{p_i}{(p,Ap_i)}\\ r_{i+1}&=r_i-\frac{Ap_i}{(p_i,Ap_i)}\\ p_{i+1}&=p_i+\frac{r_{i+1}}{|r_{i+1}|^2}. \end{aligned} \tag{9:5}$$

This result is obtained from (9:1) by choosing

$$d_i=|r_i|^2.$$

In this case the formulas (9:5) are very simple and are particularly adaptable to computation. It has the disadvantage that the vectors p_i may grow considerably in length, as can be seen from the relations

$$|p_{i+1}|^2=|p_i|^2+\frac{1}{|r_{i+1}|^2}.$$

However, if "floating" operations are used, this should present no difficulty.

III. *The vector p_i can be chosen to be the correction to be added to x_i in the $(i+1)$st relaxation.*

In this event, $a_i=1$ and the formulas (9:1) take the form

$$\begin{aligned} r_0&=k-Ax_0, \qquad p_0=\frac{r_0}{d_0}\\ x_{i+1}&=x_i+p_i\\ r_{i+1}&=r_i-Ap_i\\ p_{i+1}&=\frac{r_{i+1}+b_ip_i}{d_{i+1}}\\ d_0&=\mu(r_0), \qquad d_{i+1}=\mu(r_{i+1})-b_i\\ b_i&=\frac{|r_{i+1}|^2}{|r_i|^2}d_i. \end{aligned} \tag{9:6}$$

These relations are obtained from (9:1) and (9:2) by setting $a_i=1$.

IV. *The vector p_i can be chosen so that a_i is the reciprocal of the Rayleigh quotient of r_i.*

The formulas for a_i, b_i and d_i in (9:1) then become

$$a_i=\frac{|r_i|^2}{(r_i,Ar_i)}$$

$$b_i=\frac{|r_{i+1}|^2}{|r_i|^2}d_i$$

$$d_0=1, \qquad d_{i+1}=1-\frac{b_i a_{i+1}}{a_i}.$$

This is sufficient to indicate the variety of choices that can be made for the scalar factor $d_0, d_1, \ldots$. For purposes of computation the choice $d_i=1$ appears to be the simplest, all things considered.

10. Extensions of the cg-Method

In the preceding pages we have assumed that the matrix A is a positive definite symmetric matrix. The algorithm (3:1) still holds when A is nonnegative and symmetric. The routine will terminate when one of the following situations is met:

(1) The residual r_m is zero. In this event x_m is a solution of $Ax=k$, and the problem is solved.

(2) The residual r_m is different from zero but $(Ap_m,p_m)=0$, and hence $Ap_m=0$. Since $p_i=c_i\bar{r}_i$, it follows that $A\bar{r}_m=0$, where r_m is the residual of the vector $\bar{x}_m$ defined in section 7. The point $\bar{x}_m$ is accordingly a point at which $|k-Ax|^2$ attains its minimum. In other words, $\bar{x}_m$ is a least-square solution. One should observe that $p_m\neq 0$ (and hence $\bar{r}_m\neq 0$). Otherwise, we would have $r_m=-b_{m-1}p_{m-1}$, contrary to the fact that r_m is orthogonal to p_{m-1}. The point x_m fails to minimize the function

$$g(x)=(x,Ax)-2(k,x),$$

for in this event

$$g(x_m+tp_m)=g(x_m)-2t|r_m|^2.$$

In fact, $g(x)$ fails to have a minimum value.

It remains to consider the case when A is a general nonsingular matrix. In this event we observe that the matrix A^*A is symmetric and that the system $Ax=k$ is equivalent to the system

$$A^*Ax=A^*k. \tag{10:1}$$

Applying the eq (3:1) to this last system, we obtain the following iteration,

$$\begin{aligned} r_0&=k-Ax_0, \qquad p_0=A^*r_0,\\ a_i&=\frac{|A^*r_i|^2}{|Ap_i|^2}\\ x_{i+1}&=x_i+a_ip_i\\ r_{i+1}&=r_i-a_iAp_i\\ b_i&=\frac{|A^*r_{i+1}|^2}{|A^*r_i|^2}\\ p_{i+1}&=A^*r_{i+1}+b_ip_i. \end{aligned} \tag{10:2}$$

If one does not wish to use any properties of the cg-method in the computation of a_i and b_i besides the defining relations, since they may be disturbed by rounding-off errors, one should use the formulas

$$a_i=\frac{(Ap_i,r_i)}{|Ap_i|^2}$$

$$b_i=-\frac{(Ap_i,AA^*r_{i+1})}{|Ap_i|^2}.$$

In this case the error function $f(x)$ is the function $f(x)=|k-Ax|^2$, and hence is the squared residual. It is a simple matter to interpret the results given above for this new system.

It should be emphasized that, even though the use of the system (10:2) is equivalent from a theoretical point of view to applying the cg-algorithm to the system (10:1), the two methods are not equivalent from a numerical point of view. This follows because rounding-off errors in the two methods are not the same. The system (10:2) is the better of the two, because at all times one uses the original matrix A instead of the computed matrix A^*A, which will contain rounding-off errors.

There is a slight generalization of the system (10:2) that is worthy of note. This generalization consists of selecting a matrix B such that BA is positive definite and symmetric. The matrix B is necessarily of the form A^*H, where H is positive definite and symmetric. We can apply the cg-algorithm to the system

$$BAx=Bk. \tag{10:3}$$

In place of (10:2) one obtains the algorithm

$$\begin{aligned} r_0&=k-Ax_0, \qquad p_0=Br_0,\\ a_i&=\frac{|Br_i|^2}{(p_i,BAp_i)},\\ x_{i+1}&=x_i+a_ip_i,\\ r_{i+1}&=r_i-a_iAp_i,\\ b_i&=\frac{|Br_{i+1}|^2}{|Br_i|^2},\\ p_{i+1}&=Br_{i+1}+b_ip_i. \end{aligned} \tag{10:4}$$

Again the formulas for a_i and b_i, which are given directly by the defining relations, are

$$a_i=\frac{(p_i,Br_i)}{(p_i,BAp_i)}$$

$$b_i=-\frac{(Br_{i+1},BAp_i)}{(p_i,BAp_i)}$$

When $B=A^*$, this system reduces to (10:2). If A is symmetric and positive definite, the choice $B=I$ gives the original cg-algorithm.

There is a generalization of the cd-algorithm concerning which a few remarks should be made. In this method we select vectors $p_0, \ldots, p_{n-1}$ and $q_0, \ldots, q_{n-1}$ such that

$$(q_i, Ap_j)=0 \qquad (i\neq j),$$
$$(q_i, Ap_j)>0. \tag{10:5}$$

The solution can be obtained by the recursion formulas

$$r_0=k-Ax_0,$$
$$a_i=\frac{(q_i, r_i)}{(q_i, Ap_i)}=\frac{(q_i, r_0)}{(q_i, Ap_i)},$$
$$x_{i+1}=x_i+a_ip_i, \tag{10:6}$$
$$r_{i+1}=r_i-a_iAp_i.$$

The problem is then reduced to finding the vectors p_i, q_i such that (10:5) holds. We shall show in a moment that q_i is of the form

$$q_i=B^*p_i, \tag{10:7}$$

where B has the property that BA is symmetric and positive definite. The algorithm (10:6) is accordingly equivalent to applying the cd-algorithm to (10:3). To see that q_i is of the form (10:7), let P be the matrix whose column vectors are $p_0, \ldots, p_{n-1}$ and Q be the matrix whose column vectors are $q_0, \ldots, q_{n-1}$. The condition (10:5) is equivalent to the statement that the matrix $D=Q^*AP$ is a diagonal matrix whose diagonal terms are positive. Select B so that $Q=B^*P$. Then $D=P^*BAP$ from which we conclude that BA is a positive definite symmetric matrix, as was to be proved.

In view of the results just obtained, we see that the algorithm (10:4) is the *most general cg-algorithm for any linear system*. Similarly, the most general cd-algorithm is obtained: by (i) selecting a matrix B such that BA is symmetric and positive definite, (ii) selecting nonzero vectors $p_0, \ldots, p_{n-1}$ such that

$$(p_i, BAp_j)=0, \qquad (i\neq j)$$

and (iii), using the recursion formulas

$$r_0=k-Ax_0$$
$$a_i=\frac{(p_i, Br_i)}{(p_i, BAp_i)}=\frac{(p_i, Br_0)}{(p_i, BAp_i)}$$
$$x_{i+1}=x_i+a_ip_i$$
$$r_{i+1}=r_i-a_iAp_i.$$

11. Construction of Mutually Conjugate Systems

As was remarked in section 4 the cd-method is not complete until a method of constructing a set of mutually conjugate vectors $p_0, p_1, \ldots$ has been given. In the cg-method the choice of the vector p_i depended on the result obtained in the previous step. The vectors $p_0, p_1, \ldots$ are accordingly determined by the starting point x_0 and vary with the point x_0.

Assume again that A is a positive definite, symmetric matrix. In a cd-method the vectors $p_0, p_1, \ldots$ can be chosen to be independent of the starting point. This can be done, for example, by starting with a set of n linearly independent vectors $u_0, u_1, \ldots, u_{n-1}$ and constructing conjugate vectors by a successive A-orthogeonalization process. For example, we may use the formulas

$$\begin{aligned}
p_0&=u_0,\\
p_1&=u_1-\alpha_{10}p_0,\\
p_2&=u_2-\alpha_{20}p_0-\alpha_{21}p_1,\\
&\vdots\\
p_i&=u_i-\alpha_{i0}p_0-\alpha_{i1}p_1-\ldots-\alpha_{i,i-1}p_{i-1}.\\
&\vdots
\end{aligned} \tag{11:1}$$

The coefficient $\alpha_{ij}(i>j)$ is to be chosen so that p_i is conjugate to p_j. The formula for α_{ij} is evidently

$$\alpha_{ij}=\frac{(u_i, Ap_j)}{(p_j, Ap_j)} \qquad (j<i). \tag{11:2}$$

Observe that

$$(p_i, Au_j)=0 \qquad (j<i)$$
$$(p_i, Au_i)=(p_i, Ap_i). \tag{11:3}$$

Using (11:3) we see that alternately

$$\alpha_{ij}=\frac{(Au_i, p_j)}{(Au_j, p_j)}. \tag{11:4}$$

As described in section 4, the successive estimates of the solution are given by the recursion formula

$$x_0=0, \qquad x_{i+1}=x_i+a_ip_i, \tag{11:5}$$

where

$$a_i=\frac{(p_i, k)}{(p_i, Ap_i)}. \tag{11:6}$$

There is a second method of computing the vectors $p_0, p_1, \ldots, p_{n-1}$, given by the recursion formulas

$$u_i^{(0)}=u_i \tag{11:7a}$$
$$p_j=u_j^{(j)}, \tag{11:7b}$$
$$u_i^{(j+1)}=u_i^{(j)}-\alpha_{ij}p_j, \qquad (i=j+1, \ldots, n) \tag{11:7c}$$
$$\alpha_{ij}=\frac{(u_i^{(j)}, Au_j)}{(p_j, Au_j)} \qquad (i>j). \tag{11:7d}$$

We have the relations (11:3) and,

$$(u_i^{(k)}, Au_j)=0 \qquad (j<k) \tag{11:8a}$$

$$(u_i^{(k)}, Ap_j)=0 \qquad (j<k) \tag{11:8b}$$

$$u_i^{(k)}=u_i-\alpha_{i0}p_0-\ldots-\alpha_{i,j-1}p_{j-1} \qquad (i>j) \tag{11:8c}$$

$$(p_i, Ap_j)=0 \qquad (i\neq j). \tag{11:8d}$$

The eq (11:8a) hold when $k=j+1$ by virtue of (11:7c) and (11:7d). That they hold for other values of $j<k$ follows by induction. Equation (11:8c) follows from (11:7c).

If one selects successively $u_0=r_0$, $u_1=r_1, \ldots, u_{n-1}=r_{n-1}$, the procedure just described is equivalent to the cg-method described in section 3, in the sense that the same estimates $x_0, x_1, \ldots$ and the same direction vectors $p_0, p_1, \ldots$ are obtained. If one selects $u_0=k$, $u_1=Ak, \ldots, u_{n-1}=A^{n-1}k$, one again obtains the same estimates $x_0, x_1, \ldots$ as in the cg-method with $x_0=0$. However in this event the vectors $p_0, p_1, \ldots$ are multiplied by nonzero scalar factors. On the other hand if one selects $u_0=(1,0,\ldots,0)$, $u_1=(0,1,\ldots,0), \ldots, u_{n-1}=(0,\ldots,0,1)$ the cd-method is equivalent to the Gauss elimination method. This case will be discussed in the next section.

12. Connections With the Gauss Elimination Method [12]

In the present section it will be convenient to use the range $1, \ldots, n$ in place of $0, 1, \ldots, n-1$ used heretofore, except for the notations $x_0, x_1, \ldots, x_n$ describing the successive estimates of the solution. Let $e_1, \ldots, e_n$ be the unit vectors $(1,0,\ldots,0)$, $(0,1,0,\ldots,0), \ldots, (0,\ldots,0,1)$. These vectors will play the role of the vectors $u_0, \ldots, u_{n-1}$ of section 11. The eq (11:7), together with (11:4) and (11:5), yield the recursion formulas

$$u_i^{(1)}=e_i \qquad (i=1,\ldots,n) \tag{12:1a}$$

$$p_i=u_i^{(1)} \tag{12:1b}$$

$$u_i^{(j+1)}=u_i^{(j)}-\alpha_{ij}p_j \qquad (i=j+1,\ldots,n) \tag{12:1c}$$

$$\alpha_{ij}=\frac{(Au_i^{(j)}, e_j)}{(Ap_j, e_j)} \tag{12:1d}$$

$$x_0=0, \qquad x_i=x_{i-1}+a_ip_i \tag{12:1e}$$

$$a_i=\frac{(p_i,k)}{(Ap_i,e_i)}. \tag{12:1f}$$

These formulas generate mutually conjugate vectors $p_1, \ldots, p_n$ and corresponding estimates $x_1, \ldots, x_n$ of the solution of $Ax=k$. In particular x_n is the desired solution. The advantage of this method lies in the ease with which the inner products appearing in (12:1d) and (12:1f) can be computed. A systematic scheme for carrying out the computations will now be given. The scheme is that commonly used in elimination. In the presentation that we now give, extraneous entries will be kept so as to give the reader a clear picture of the results obtained.

We begin by writing the matrices A, I and the vector k as a single matrix

$$\begin{array}{cccccccccc}
a_{11} & a_{12} & a_{13} & \ldots & a_{1n} & 1 & 0 & 0 & \ldots & 0 & k_1 \\
a_{21} & a_{22} & a_{23} & \ldots & a_{2n} & 0 & 1 & 0 & \ldots & 0 & k_2 \\
a_{31} & a_{32} & a_{33} & \ldots & a_{3n} & 0 & 0 & 1 & \ldots & 0 & \cdot \\
\cdot & \cdot & \cdot & & \cdot & \cdot & & & & \cdot & \cdot \\
\cdot & \cdot & \cdot & & \cdot & \cdot & & & & \cdot & \cdot \\
\cdot & \cdot & \cdot & & \cdot & \cdot & & & & 0 & \cdot \\
a_{n1} & a_{n2} & a_{n3} & \ldots & a_{nn} & 0 & & \ldots & 0 & 1 & k_n.
\end{array} \tag{12:2}$$

The vector p_1 is the vector $(1,0,\ldots,0)$, and $a_1=k_1/a_{11}$ is defined by (12:1f). Hence,

$$x_1=a_1p_1=\left(\frac{k_1}{a_{11}}, 0, \ldots, 0\right)$$

is our first estimate. Observe also that

$$\alpha_{i1}=\frac{a_{i1}}{a_{11}}.$$

Multiplying the first row by α_{i1} and subtracting the result from the ith row $(i=2, \cdots, n)$, we obtain the new matrix

$$\begin{array}{ccccccccc}
a_{11} & a_{12} & a_{13} & \ldots & a_{1n} & p_{11} & \ldots & p_{1n} & k_1 \\
0 & a_{22}^{(2)} & a_{23}^{(2)} & \ldots & a_{2n}^{(2)} & p_{21} & \ldots & p_{2n} & k_2^{(2)} \\
0 & a_{32}^{(2)} & a_{33}^{(2)} & \ldots & a_{3n}^{(2)} & u_{31}^{(2)} & \ldots & u_{3n}^{(2)} & k_3^{(2)} \\
\cdot & & & & \cdot & \cdot & & \cdot & \cdot \\
\cdot & & & & \cdot & \cdot & & \cdot & \cdot \\
\cdot & & & & \cdot & \cdot & & \cdot & \cdot \\
0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \ldots & a_{nn}^{(2)} & u_{n1}^{(2)} & \ldots & u_{nn}^{(2)} & k_n^{(2)}
\end{array} \tag{12:3}$$

One should observe that $(0,k_2^{(2)},\ldots,k_n^{(2)})$ is the residual of x_1. By the procedure just described the ith row $(i>1)$ of the identity matrix has been replaced by $u_i^{(2)}$, the second row yielding the vector $p_2=u_2^{(2)}$. Observe also that

$$a_{i2}^{(2)}=(Au_i^{(2)}, e_i) \qquad (i=2,\ldots,n)$$

$$a_{22}^{(2)}=(Ap_2, e_2), \qquad k_2^{(2)}=(p_2, k).$$

Hence,

$$x_2=x_1+\frac{k_2^{(2)}}{a_{22}^{(2)}}p_2$$

is the next estimate of the solution. Moreover,

$$\alpha_{i2}=\frac{a_{i2}^{(2)}}{a_{22}^{(2)}} \qquad (i=3,\ldots,n).$$

[12] cf. Fox, Huskey, and Wilkinson, loc. cit.

Next multiply the 2nd row of (12:3) by α_{i2} and subtract the result from the ith row $(i=3,\cdots,n)$. We obtain

$$\begin{array}{ccccccccc}
a_{11} & a_{12} & a_{13} & \dots & a_{1n} & p_{11} & \dots & p_{1n} & k_1 \\
0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2n}^{(2)} & p_{21} & \dots & p_{2n} & k_2^{(2)} \\
0 & 0 & a_{33}^{(3)} & \dots & a_{3n}^{(3)} & p_{31} & \dots & p_{3n} & k_3^{(3)} \\
0 & 0 & a_{43}^{(3)} & \dots & a_{4n}^{(3)} & u_{41}^{(3)} & \dots & u_{4n}^{(3)} & k_4^{(3)} \\
\cdot & \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot \\
\cdot & \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot \\
\cdot & \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot \\
0 & 0 & a_{n3}^{(3)} & \dots & a_{nn}^{(3)} & u_{n1}^{(3)} & \dots & u_{nn}^{(3)} & k_n^{(3)}.
\end{array}$$

The vector $(0,0,k_3^{(3)},\dots,k_n^{(3)})$ is the residual of x_2. The elements $u_{i1}^{(3)},\dots,u_{in}^{(3)}$ form the vector $u_i^{(3)}$ $(i=3)$ and $p_3=u_3^{(3)}$. We have

$$a_{i3}^{(3)}=(Au_i^{(3)}, e_i) \qquad (i=3,\dots,n)$$

$$a_{33}^{(3)}=(Ap_3, e_3), \qquad k_3^{(3)}=(p_3, k).$$

We have accordingly

$$x_3=x_2+\frac{k_3^{(3)}}{a_{33}^{(3)}}\,p_3,$$

and

$$\alpha_{i3}=\frac{a_{i3}^{(3)}}{a_{33}^{(3)}} \qquad (i=4,\dots,n)$$

Proceeding in this manner, we finally obtain a matrix of the form

$$\begin{array}{ccccccccc}
a_{11} & a_{12} & a_{13} & \dots & a_{1n} & p_{11} & \dots & p_{1n} & k_1 \\
0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2n}^{(2)} & p_{21} & \dots & p_{2n} & k_1^{(2)} \\
0 & 0 & a_{33}^{(3)} & \dots & a_{3n}^{(3)} & p_{31} & \dots & p_{3n} & k_1^{(3)} \\
\cdot & \cdot & & & \cdot & \cdot & & \cdot & \cdot \\
\cdot & \cdot & & & \cdot & \cdot & & \cdot & \cdot \\
\cdot & \cdot & & & \cdot & \cdot & & \cdot & \cdot \\
0 & 0 & \dots & 0 & a_{nn}^{(n)} & p_{n1} & \dots & p_{nn} & k_n^{(n)}
\end{array} \qquad (12:4)$$

The elements $p_{i1},\cdots,p_{in}$ define a vector p_i. The vectors $p_1,\cdots,p_n$ are the mutually conjugate vectors defined by the iteration (12:1). At each stage

$$\alpha_{ij}=\frac{a_{ij}^{(j)}}{a_{jj}^{(j)}} \qquad (i=j+1,\dots,n)$$

$$a_{ii}^{(i)}=(p_i, Ap_i), \qquad k_i^{(i)}=(p_i, k)=(p_i, r_i).$$

Moreover the estimate x_i of the solution h is given by the formula

$$x_i=x_{i-1}+\frac{k_i^{(i)}}{a_{ii}^{(i)}}\,p_i.$$

The vector $0,\cdots,0,a_{ii}^{(i)},\cdots,a_{in}^{(i)}$ defined by the first n elements in the ith row of (12:4) is the vector Ap_i. If we denote by P the matrix whose column vectors are $p_1, p_2,\cdots,p_n$, then the matrix (12:4) is the matrix

$$||P^*A \quad P^* \quad P^*k||.$$

The matrices P^*A and P are triangular matrices with zeros below the diagonal. The matrix $D=P^*AP$ is the diagonal matrix whose diagonal elements are $a_{11}, a_{22}^{(2)},\dots,a_{nn}^{(n)}$. The determinant of P is unity and the determinant of A is the product

$$a_{11}a_{22}^{(2)}\dots a_{nn}^{(n)}.$$

As was seen in section 4, if we let

$$f(x)=(h-x, A(h-x)),$$

the sequence

$$f(x_0), f(x_1),\dots, f(x_{n-1}), f(x_n)=0$$

decreases monotonically. No general statement can be made for the sequence

$$|y_0|, |y_1|,\dots, |y_{n-1}|, |y_n|=0$$

of lengths of the error vectors $y_1=h-x_1$. In fact, we shall show that this sequence can increase monotonically, except for the last step. A situation of this type cannot arise when the cg-process is used.

If A is nonsymmetric, the interpretation given above must be modified somewhat. An analysis of the method will show that one finds implicitly two triangular matrices P and Q such that Q^*AP is a diagonal matrix. To carry out this process, it may be necessary to interchange rows of A. By virtue of the remarks in section 10, the matrix Q^* is of the form B^*P. The general procedure is therefore equivalent to application of the above process to the system (10:3).

13. An Example

In the cg-method the estimates $x_0, x_1,\dots$ of the solution h of $Ax=k$ have the property that the error vectors $y_0=h-x_0$, $y_1=h-x_1,\dots$ are decreased in length at each step. This property is not enjoyed by every cd-method. In this section we construct an example such that, for the estimates $x_0=0, x_1,\dots$, of the elimination method,

$$|h-x_{i-1}|<|h-x_i| \qquad (i=1,\dots,n-1).$$

If the order of elimination is changed, this property may not be preserved.

The example we shall give is geometrical instead of numerical. Start with an $(n-1)$-dimensional ellipsoid E_n with center $x_n=h$ and with axes of unequal length. Draw a chord C_n through x_n, which is not orthogonal to an axis of E_n. Select a point $x_{n-1}\neq x_n$ on this chord inside E_n, and pass a hyperplane P_{n-1} through x_{n-1} conjugate to C_n, that is, parallel to the plane determined by the midpoints of the chords of E_n parallel to C_n. Let e_n be a unit vector normal to P_{n-1}. It is clear that e_n is not

parallel to C_n. The plane P_{n-1} can be shown to cut E_n in an $(n-2)$-dimensional ellipsoid E_{n-1} with center at x_{n-1} and with axes of unequal length.

Next draw a chord C_{n-1} of E_{n-1} through x_{n-1} which is not orthogonal to an axis of E_{n-1}, and which is not perpendicular to $h-x_{n-1}$. One can then select a point x_{n-2} on C_{n-1} which is nearer to h than x_{n-1}. Let P_{n-2} be the hyperplane through x_{n-2} conjugate to C_{n-1}. It intersects E_{n-1} in an $(n-3)$-dimensional ellipsoid E_{n-2} with center at x_{n-2}. The axes of E_{n-2} can be shown to be of unequal lengths. Let e_{n-1} be a unit vector in P_{n-1} perpendicular to P_{n-2}.

We now repeat the construction made in the last paragraph. Select a chord C_{n-2} of E_{n-2} through x_{n-2} that is not orthogonal to an axis of E_{n-2} and that is not perpendicular to $h-x_{n-2}$. Select x_{n-3} on C_{n-2} nearer to h than x_{n-2}, and let P_{n-3} be a plane through x_{n-3} conjugate to C_{n-2}. It cuts E_{n-2} in an $(n-4)$-dimensional ellipsoid E_{n-3} with center at x_{n-3} with axes of unequal lengths. Let e_{n-2} be a unit vector in P_{n-1} and P_{n-2} perpendicular to P_{n-3}. Clearly, e_n, e_{n-1}, e_{n-2} are mutually perpendicular.

Proceeding in this manner, we can construct

(1) Chords $C_n, C_{n-1}, \ldots, C_1$, which are mutually conjugate.

(2) Planes $P_{n-1}, \ldots, P_1$ such that P_k is conjugate to C_{k+1}. The chords $C_1, \ldots, C_k$ lie in P_k.

(3) The intersection of the planes $P_{n-1}, \ldots, P_k$, which cuts E_n in a $(k-1)$-dimensional ellipsoid E_k with center x_k.

(4) The point x_i, which is closer to h than x_{i+1}, $i<n-1$.

(5) The unit vectors $e_n, \ldots, e_2, e_1$ (with e_1 in the direction of C_1), which are mutually orthogonal.

Let x_0 be an arbitrary point on C_1 that is nearer to h than x_1. Select a coordinate system with x_0 as the origin and with $e_1, \ldots, e_n$ as the axes. In this coordinate system the elimination method described in the last section will yield as successive estimates the points $x_1, \ldots, x_n$ described above. These estimates have the property that x_i is closer to $x_n=h$ than x_{i+1} if $i<n-1$.

As a consequence of the construction just made we see that, given a set of mutually conjugate vectors $p_1, \ldots, p_n$ and a starting point x_0, one can always choose a coordinate system such that the elimination method will generate the vectors $p_1, \ldots, p_n$ (apart from scalar factors) and will generate the same estimates $x_1, \ldots, x_n$ of h as the cd-method determined by these data. One needs only to select the origin at x_0, the vector e_1 parallel to p_1, the vector e_2 in the plane of p_1 and p_2 and perpendicular to e_1, the vector e_3 in the plane of p_1, p_2, p_3 and perpendicular to e_1 and e_2, and so on. This result may have no practicale value, but it does serve to clarify the relationship between the elimination method and the cd-method, and also the relationship between the elimination method and the cg-method.

14. A Duality Between Orthogonal Polynomials and n-Dimensional Geometry

The method of conjugate gradients is related to the theory of orthogonal polynomials and to continued fraction expansions. To develop this, we first study connections between orthogonal polynomials and n-dimensional geometry.

Let $m(\lambda)$ be a nonnegative and nondecreasing function on the interval $0\leq\lambda\leq l$. The (Riemann) Stieltjes integral

$$\int_0^l f(\lambda)dm(\lambda)$$

then exists for any continuous function $f(\lambda)$ on $0\leq\lambda\leq l$. We call $m(\lambda)$ a *mass distribution* on the positive λ-axis. The following two cases must be distinguished.

(a) The function $m(\lambda)$ has infinitely many points of increase on $0<\lambda<l$.

(b) There are only a finite number n of points of increase. In both cases we may construct by orthogonalization of the successive powers $1, \lambda, \lambda^2, \cdots, \lambda^n$ a set of $n+1$ orthogonal polynomials [13]

$$R_0(\lambda), R_1(\lambda), \cdots, R_n(\lambda) \qquad (14:1)$$

with respect to the mass distribution. One has

$$\int_0^l R_i(\lambda)R_k(\lambda)dm(\lambda)=0 \qquad (i\neq k). \qquad (14:2)$$

The polynomial $R_i(\lambda)$ is of degree i. In case (b), $R_n(\lambda)$ is a polynomial of degree n having its zeros at the n points of increase of $m(\lambda)$. In both cases the zeros of each of the polynomials (14:1) are real and distinct and located inside the interval $(0,l)$. Hence we may normalize the polynomials so that

$$R_i(0)=1 \qquad (i=1, \cdots, n). \qquad (14:3)$$

The polynomials (14:1) are then uniquely determined by the mass distribution.

During the following investigations we use the *Gauss mechanical quadrature as* a basic tool. It can be described as follows: If $\lambda_1, \cdots, \lambda_n$ denote the zeros of $R_n(\lambda)$, there exist *positive* weight coefficients $m_1, m_2, \cdots, m_n$ such that,

$$\int_0^l R(\lambda)dm(\lambda)=m_1R(\lambda_1)+m_2R(\lambda_2)+\ldots+m_nR(\lambda_n) \qquad (14:4)$$

whenever $R(\lambda)$ is a polynomial of degree at most $2n-1$. In the special case b) the λ_k are the abscissas where $m(\lambda)$ jumps and the m_k the corresponding jump.

In order to establish the duality mentioned in the title of this section, we construct a positive definite matrix A having $\lambda_1, \lambda_2, \cdots, \lambda_n$ as eigenvalues (for instance, the diagonal matrix having $\lambda_1, \cdots, \lambda_n$ in the main diagonal and zeros elsewhere). Furthermore, if $e_1, e_2, \cdots, e_n$ are the normalized eigenvectors of A, we introduce the vector

$$r_0=\alpha_1e_1+\alpha_2e_2+\cdots+\alpha_ne_n, \qquad (14:5)$$

[13] The various properties of orthogonal polynomials used in this chapter may be found in G. Szegö, *Orthogonal Polynomials*, American Mathematical Society Colloquium Publications **23** (1939).

where

$$\alpha_i^2 = m_i \qquad (i=1, \cdots, n). \tag{14:6}$$

We then have

$$A^k r_0 = \alpha_1 \lambda_1^k e_1 + \alpha_2 \lambda_2^k e_2 + \cdots + \alpha_n \lambda_n^k e_n \tag{14:7}$$

for $k=0,1, \cdots, n-1$. The vectors $r_0, Ar_0, \cdots, A^{n-1}r_0$ are linearly independent and will be used as a coordinate system. Indeed their determinant is up to the factor $\alpha_1\alpha_2 \cdots \alpha_n$ Van der Monde's determinant of $\lambda_1, \cdots, \lambda_n$. By the correspondence

$$\lambda_k \to A^k r_0 \qquad (k=0,1, \cdots, n-1) \tag{14:8}$$

every polynomial of *maximal degree* $n-1$ is mapped onto a vector of the n-dimensional space and a one-one correspondence between these polynomials and vectors is established. The correspondence has the following properties:

Theorem 14:1. *Let the space of polynomials* $R(\lambda)$ *of degree* $\leq n-1$ *be metrized by the norm*

$$\|R\| = \left[\int_0^l R(\lambda)^2 dm(\lambda)\right]^{\frac{1}{2}}.$$

Then the correspondence described above is isometric, that is,

$$\int_0^l R(\lambda) R'(\lambda)\, dm(\lambda) = (r, r'),$$

where $R(\lambda)$, $R'(\lambda)$ *are the polynomials corresponding to* r *and* r'.

It is sufficient to prove this for the powers $1, \lambda, \lambda^2, \ldots, \lambda^{n-1}$. Let λ^j, λ^k be two of these powers. From Gauss' formula (14:4) follows

$$\int_0^l \lambda^j \lambda^k dm(\lambda) = \int_0^l \lambda^{j+k} dm(\lambda)$$
$$= m_1 \lambda_1^{j+k} + m_2 \lambda_2^{j+k} + \ldots + m_n \lambda_n^{j+k}.$$

But (14:5), (14:6), and (14:7) show that this is exactly the scalar product $(A^j r_0, A^k r_0)$ of the corresponding vectors.

Theorem 14:2. *Let the space of polynomials* $R(\lambda)$ *of degree* $\leq n-1$ *be metrized by the norm*

$$\left[\int_0^l R(\lambda)^2 \lambda dm(\lambda)\right]^{\frac{1}{2}}.$$

Then for polynomials $R(\lambda), R'(\lambda)$ *corresponding to* r, r' *one has*

$$\int_0^l R(\lambda) R'(\lambda)\, \lambda dm(\lambda) = (Ar, r'), \tag{14:9}$$

that is, the correspondence is isometric with respect to the weight function $\lambda dm(\lambda)$ *and the metric, determined by the matrix* A.

Again we may restrict ourselves to the powers $1, \lambda, \ldots, \lambda^{n-1}$. That is, we must show that

$$\int_0^l \lambda^{j+1} \lambda^k dm(\lambda) = (A^{j+1} r_0, A^k r_0) \qquad (j, k \leq n-1). \tag{14:10}$$

If $j < n-1$, this has already been verified. The remaining case

$$\int_0^l \lambda^n \lambda^k dm(\lambda) = (A^n r_0, A^k r_0) \qquad (k \leq n-1) \tag{14:11}$$

follows in the same manner from Gauss' integration formula, since $n+k \leq 2n-1$.

Theorem 14:3. *Let* A *be a positive definite symmetric matrix with distinct eigenvalues and let* r_0 *be a vector that is not perpendicular to an eigenvector of* A. *There is a mass distribution* $m(\lambda)$ *related to* A *as described above.*

In order to prove this result let $e_1, \ldots, e_n$ be the normalized eigenvectors of A and let $\lambda_1, \ldots, \lambda_n$ be the corresponding (positive) eigenvalues. The vector r_0 is expressible in the form (14:5). According to our assumption no α_k vanishes. The desired mass distribution can be constructed as a step function that is constant on each of the intervals $0 < \lambda_1 < \lambda_2 < \ldots < \lambda_n < l$, and having a jump at λ_k of the amount $m_k = \alpha_k^2 > 0$, the number l being any number greater than λ_n.

We want to emphasize the following property of our correspondence. If A and r_0 are given, we are able to establish the corredence *without computing eigenvalues of* A. This follows immediately from the basic relation (14:8). Moreover, we are able to compute integrals of the type

$$\int_0^l R(\lambda) R'(\lambda)\, dm(\lambda), \qquad \int_0^l R(\lambda) R'(\lambda)\, \lambda dm(\lambda), \tag{14:12}$$

where R, R' are polynomials of maximal degree $n-1$ without constructing the mass distribution. Indeed, the integrals are equal to the corresponding scalar products (r, r'), (Ar, r') of the corresponding vectors, by virtue of theorems 14:1 and 14:2. Finally, the same is true for the construction of the orthogonal polynomials $R_0(\lambda), R_1(\lambda), \ldots, R_n(\lambda)$ because the construction only involves the computation of integrals of the type (14:12). The corresponding vectors $r_0, r_1, \ldots, r_{n-1}$ build *an orthogonal basis in the Euclidian n-space.*

15. An Algorithm for Orthogonalization

In order to obtain the orthogonalization of polynomials, the following method can be used. For any three consecutive orthogonal polynomials the recurrence relation holds:

$$R_{i+1}(\lambda) = (d_i - a_i \lambda) R_i(\lambda) - c_i R_{i-1}(\lambda) \qquad R_0 = 1,\ c_0 = 0, \tag{15:1}$$

where a_i, c_i, d_i are real numbers and $a_i \neq 0$. Taking into account the normalization (14:3), we have

$$1 = d_i - c_i. \tag{15:2}$$

Hence

$$R_{i+1}(\lambda) = (1 + c_i - a_i\lambda)R_i(\lambda) - c_i R_{i-1}(\lambda).$$

This relation can be written

$$\frac{R_{i+1} - R_i}{a_i\lambda} = -R_i + \frac{c_i}{a_i}\frac{R_i - R_{i-1}}{\lambda}.$$

From this equation it is seen by induction that

$$P_i(\lambda) = -\frac{R_{i+1} - R_i}{a_i\lambda} \tag{15:3}$$

are polynomials of degree i. Introducing the numbers

$$b_{i-1} = \frac{c_i a_{i-1}}{a_i}, \qquad b_{-1} = 0 \tag{15:4}$$

we have

$$P_i(\lambda) = R_i(\lambda) + b_{i-1}P_{i-1}(\lambda) \tag{15:5a}$$

$$R_{i+1}(\lambda) = R_i(\lambda) - a_i\lambda P_i(\lambda). \tag{15:5b}$$

Beginning with $R_0 = 1$, we are able to compute by (15:5) successively the polynomials $P_0 = R_0 = 1$, R_1, P_1, R_2, P_2, . . ., provided that we know the numbers a_i, b_i. In order to compute them, observe first the relation

$$\int_0^l P_i(\lambda)P_k(\lambda)\lambda dm(\lambda) = 0 \qquad (i \neq k). \tag{15:6}$$

Indeed this integral is up to a constant factor

$$\int_0^l (R_{i+1} - R_i)P_k dm(\lambda).$$

For $k < i$ this is zero, because the second factor is of degree $k < i$.

Using (15:5a) and (15:6), we obtain

$$\int_0^l R_i(\lambda)P_i(\lambda)\lambda dm(\lambda) = \int_0^l P_i(\lambda)^2\lambda dm(\lambda).$$

Combining this result with the orthogonality of R_{i+1} and R_i, we see, by (15:5b), that

$$a_i = \frac{\int_0^l R_i(\lambda)^2 dm(\lambda)}{\int_0^l P_i(\lambda)^2\lambda dm(\lambda)}. \tag{15:7}$$

Using (15:6) and (15:5a),

$$0 = \int_0^l R_i(\lambda)P_{i-1}(\lambda)\lambda dm(\lambda) + b_{i-1}\int_0^l P_{i-1}(\lambda)^2\lambda dm(\lambda).$$

Applying (15:3) to the first term yields

$$\frac{1}{a_{i-1}}\int_0^l R_i(\lambda)^2 dm(\lambda) = b_{i-1}\int_0^l P_{i-1}(\lambda)^2\lambda dm(\lambda).$$

Combining this result with (15:7), we obtain

$$b_{i-1} = \frac{\int_0^l R_i(\lambda)^2 dm(\lambda)}{\int_0^l R_{i-1}(\lambda)^2 dm(\lambda)}. \tag{15:8}$$

The formulas (15:5), (15:7), (15:8), together with $R_0 = 1$, $b_{-1} = 0$, completely determine the polynomials R_0, R_1, . . ., R_{n-1}.

16. A New Approach to the cg-Method, Eigenvalues

In order to solve the system $Ax = k$, we compute the residual vector $r_0 = k - Ax_0$ of an initial estimate x_0 of the solution h and establish the correspondence based on A, r_0 described in Theorem 14:3. Without computing the mass distribution, the orthogonalization process of the last section may be carried out by (15:5), (15:7) and (15:8) with $R_0 = 1$, $b_{-1} = 0$. The vectors r_i, p_i corresponding to the polynominals R_i, P_i are therefore determined by the recurrence relations

$$p_i = r_i + b_{i-1}p_{i-1}, \qquad r_{i+1} = r_i - a_i Ap_i. \tag{16:1}$$

Multiplication by λ in the domain of polynominals is mapped by our correspondence into applying A in the vector space according to (14:11). In fact,

$$p_i = P_i(A)r_0, \quad r_i = R_i(A)r_0 \qquad (i = 0, 1, \ldots, n-1).$$

The numbers a_i, b_i are computed by (15:7) and (15:8). Using the isometric properties described in theorems 14:1 and 14:2, we find that

$$a_i = \frac{|r_i|^2}{(Ap_i, p_i)}, \qquad b_{i-1} = \frac{|r_i|^2}{|r_{i-1}|^2}.$$

The vectors r_i are orthogonal, and the p_i are conjugate; the latter result follows from (15:6). Hence the basic formulas and properties of the cg-method listed in sections 3 and 5 are established. It remains to prove that the method gives the exact solution after n steps. If we set $x_{i+1} = x_i + a_i p_i$, the corresponding residual is r_{i+1} as follows by induction:

$$k - Ax_{i+1} = (k - Ax_i) - a_i Ap_i = r_i - a_i Ap_i = r_{i+1}.$$

For the last residual r_n we have $(i = 0, 1, \ldots, n-1)$

$$\begin{aligned}(r_n, r_i) &= (r_{n-1}, r_i) - a_{n-1}(Ap_{n-1}, r_i)\\ &= \int R_{n-1}R_i dm - a_{n-1}\int P_{n-1}R_i\lambda dm\\ &= \int R_n R_i dm = 0.\end{aligned}$$

Our basic method reestablishes also the methods of C. Lanczos for computing the *characteristic polynomial* of a given matrix A. Indeed the polynomials R_i, computed by the recurrence relation (15:5), lead finally to the polynomial $R_n(\lambda)$, which, by the basic definition of the correspondence in section 14, is the characteristic polynomial of A, provided that r_0 satisfies the conditions given in theorem 14:3. It may be remembered that orthogonal polynomials build a Sturmian sequence. Therefore, the polynomials $R_0, R_1, \ldots, R_n$ build a *Sturmian sequence for the eigenvalues of the given matrix* A.

Our correspondence allows us to translate every method or result in the vector-space into an analogous method or result for polynomials and vice versa. Let us take as an example the smoothing process in section 7. It is easy to show that the vector $\bar{r}_i$ introduced in that section corresponds to a polynomial $\overline{R}_i(\lambda)$ characterized by the following property: $\overline{R}_i(\lambda)$ is the polynomial of degree i with $\overline{R}_i(0)=1$ having the least-square integral on $(0,l)$. In other words, if r_0 is given by (14:5), then

$$\alpha_1^2\overline{R}_i(\lambda_1)^2+\alpha_2^2\overline{R}_2(\lambda_2)^2+\ \ldots\ +\alpha_n^2\overline{R}_i(\lambda_n)^2=\text{minimum}.$$

This result may be used to estimate a single eigenvalue of A. In order to compute, for instance, the lowest eigenvalue λ_1, we select r_0 near to the corresponding eigenvector. The first term in the expansion being dominant, the smallest root of $\overline{R}_i(\lambda)$ will be a good approximation of λ_1, provided that i is not too small. Hence the last residual vanishes, being orthogonal to $r_0, r_1, \ldots, r_{n-1}$. It follows that x_n is the desired solution.

17. Example, Legendre Polynomials

Any known set of orthogonal polynomials yields an example of a cg-algorithm. Take, for instance, the Legendre polynomials. Adapted to the interval (0,1), they satisfy the recurrence relation

$$R_{i+1}(\lambda)=\frac{2i+1}{i+1}(1-2\lambda)R_i(\lambda)-\frac{i}{i+1}R_{i-1}(\lambda),\quad R_i(0)=1.$$

From (15:1) and (15:4)

$$a_i=\frac{4i+2}{i+1},\qquad c_i=\frac{i}{i+1},\qquad b_{i-1}=\frac{2i-1}{2i+1}.\qquad (17:1)$$

This gives the following result, let A be a symmetric matrix having the roots of the Legendre polynomial $R_n(\lambda)$ as eigenvalues, and let

$$r_0=\alpha_1e_1+\alpha_2e_2+\ \ldots\ +\alpha_ne_n,$$

where $e_1, \ldots, e_n$ are the normalized eigenvectors of A, and $m_1=\alpha_1^2,\ m_2=\alpha_2^2, \ldots, m_n=\alpha_n^2$ are the weight-coefficients for the Gauss' mechanical quadrature with respect to R_n. The cg-algorithm applied to A_ir_0 yields the numbers a_i, b_i given by (17:1). Moreover,

$$(r_i,r_i)=b_{i-1}b_{i-2}\ \ldots\ b_0(r_0,r_0)=\frac{1}{2i+1}(r_0,r_0)\qquad (i<n).$$

Hence the residuals decrease during the alogrithm. It may be worth noting that the Rayleigh quotient of r_i is

$$\frac{(r_i,Ar_i)}{|r_i|^2}=\frac{1}{a_i}+\frac{b_{i-1}}{a_{i-1}}=\frac{1}{2}.$$

All residual vectors have the same Rayleigh quotient. This shows that, unlike many other relaxation methods, the cg-process does not necessarily have the tendency of smoothing residuals.

The Chebyshev polynomials yield an example where a_i, b_i are constant for $i>0$.

18. Continued Fractions

Suppose that we have given a mass distribution of type (b) as described in section 14. The function $m(\lambda)$ is a step function with jumps at $0<\lambda_1<\lambda_2<\ldots<\lambda_n<l$, the values of the jumps being $m_1, m_2, \ldots, m_n$, respectively. It is well known[14] that the orthogonal polynominals $R_0(\lambda), R_1(\lambda), \ldots, R_n(\lambda)$, corresponding to this mass distribution, can be constructed by expanding the rational function

$$F(\lambda)=\frac{m_1}{\lambda-\lambda_1}+\cdots+\frac{m_n}{\lambda-\lambda_n}\qquad (18:1)$$

in a continued fraction. The polynominal $R_i(\lambda)$ is the denominator of the ith convergent. For our purposes it is convenient to write the continued fraction in the form

$$F(\lambda)=\cfrac{-1}{d_0-a_0\lambda-\cfrac{c_1}{d_1-a_1\lambda-\cfrac{c_2}{d_2-a_2\lambda-\cdots-\cfrac{c_{n-1}}{d_{n-1}-a_{n-1}\lambda}}}}\qquad (18:2)$$

[14] H. S. Wall, Analytic Theory of Continued fractions, Van Norstrand (1948).

The denominators of the convergents are given by the recursion formulas

$$R_{i+1}=(d_i-a_i\lambda)R_i-c_iR_{i-1}, \quad R_0=1, \quad c_0=0. \qquad (18:3)$$

This coincides with (15:1). However, in order to satisfy (14:3), the expansion must be carried out so that $d_i=c_i+1$, by virtue of (15:2). The numbers b_i are then given by (15:4). It is clear that

$$F(\lambda)=\frac{Q_{n-1}(\lambda)}{R_n(\lambda)}$$

where

$$Q_{n-1}(\lambda)=\sum_{i=1}^{n} m_i \prod_{j\neq i}(\lambda-\lambda_j).$$

Let us translate these results into the n-dimensional space given by our correspondence. As before, we construct a positive definite symmetric matrix A with eigenvalues $\lambda_1, \ldots, \lambda_n$. Let $e_1, \ldots, e_n$ be corresponding eigenvectors of unit length and choose, as before,

$$r_0=\alpha_1e_1+ \ldots +\alpha_ne_n, \qquad \alpha_i^2=m_i.$$

The eigenvalues are the reciprocals of the squares of the semiaxis of the $(n-1)$-dimensional ellipsoid $(x, Ax)=1$. The hyperplane, $(r_0, x)=0$, cuts this ellipsoid in an $(n-2)$-dimensional ellipsoid, E_{n-2}, the squares of whose semiaxis are given by the reciprocals of the zeros of the numerator $Q_{n-1}(\lambda)$ of $F(\lambda)$.

This follows from the fact that if λ_0 is a number such that there is a vector $x_0\neq 0$ orthogonal to r_0 having the property that $(Ax_0, x)=\lambda_0(x_0, x)$ whenever $(r_0, x)=0$, then λ_0 is the square of the reciprocal of the semiaxis of E_{n-2} whose direction is given by x_0. If the coordinate system is chosen so that the axes are given by $e_1, \ldots, e_n$, respectively, then $\lambda=\lambda_0$ satisfies the equation

$$Q_{n-1}(\lambda)=\begin{vmatrix} \lambda_1-\lambda & 0 & 0 & \ldots & 0 & \alpha_1 \\ 0 & \lambda_2-\lambda & 0 & \ldots & 0 & \alpha_2 \\ 0 & 0 & & \ldots & & \\ \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ 0 & 0 & & \ldots & \lambda_n-\lambda & \alpha_n \\ \alpha_1 & \alpha_2 & & \ldots & \alpha_n & 0 \end{vmatrix}=0$$

as was to be proved.

Let us call the zeros of $Q_{n-1}(\lambda)$ the *eigenvalues of* A with respect to r_0 and the polynomial $Q_{n-1}(\lambda)$ the *characteristic polynomial of* A *with respect to* r_0. The rational function $F(\lambda)$ is accordingly the quotient of this polynomial and the characteristic polynomial of A. Hence we have,

Theorem 18:1. *The numbers a_i, b_i connected with the cg-process of a matrix A and a vector x_0 can be computed by expanding into a continued fraction the quotient built by the characteristic polynomial of A with respect to r_0 and the ordinary characteristic polynomial of A.*

This is the simplest form of the relation between a matrix A, a vector r_0 and the numbers a_i, b_i of the corresponding cg-process. The theorem may be used to investigate the behavior of the a_i, b_i if the eigenvalues of A and those with respect to r_0 are given. The following special case is worth recording. If $m_1=m_2= \ldots =m_n=1$, the rational function is the logarithmic derivative of the characteristic polynomial. From theorem (18:1) follows

Theorem 18:2. *If the vector r_0 of a cg-process is the sum of the normalized eigenvectors of A, the numbers a_i, b_i may be computed by expanding the logarithmic derivative of the characteristic polynomial of A into a continued fraction.*

Finally, we are able to prove

Theorem 18:3. *There is no restriction whatever on the positive constants a_i, b_i in the cg-process, that is, given two sequences of positive numbers $a_0, a_1, \ldots, a_{n-1}$ and $b_0, b_1, \ldots, b_{n-1}$, there is a symmetric positive definite matrix A and a vector r_0 such that the cg-algorithm applied to A, r_0 yield the given numbers.*

The demonstration goes along the following lines: From (15:2) and (15:4), we compute the numbers c_i, d_i, the c_i being again positive. Then we use the continued fraction (18:2) to compute $F(\lambda)$ which we decompose into partial fractions to obtain (18:1). We show next that the numbers λ_i, m_i appearing in (18:1) are positive. After this has been established, our correspondence finishes the proof.

In order to prove that $\lambda_i>0$, $m_i>0$ we observe that the ratio R_{i+1}/R_i is a decreasing function of λ, as can be seen from (18:3) by induction. Using this result, it is not too difficult to show that the polynomials $R_0(\lambda), R_1(\lambda), \ldots, R_n(\lambda)$ build a Sturmian sequence in the following sense. The number of zeros of $R_n(\lambda)$ in any interval $a\leq\lambda\leq b$ is equal to the increase of the number of variations in sign in going from a to b. At the point λ_0 there are no variations in sign since $R_i(0)=1$ for every i. At $\lambda=+\infty$, there are exactly n variations because the coefficient of the highest power of λ in $R_i(\lambda)$ is $(-1)^ia_0a_1 \ldots a_{i-1}$. Therefore, the roots $\lambda_1, \lambda_2, \ldots, \lambda_n$ of $R_n(\lambda)$ are real and positive.

That the function $F(\lambda)$ is itself a decreasing function of λ follows directly from (18:2). Therefore, its residues $m_1, m_2, \ldots, m_n$ are positive.

In view of theorem 18:3 the numbers a_i in a cg-process can increase as fast as desired. This result was used in section 8.2. Furthermore, the formula

$$b_i=\frac{|r_{i+1}|^2}{|r_i|^2}$$

shows that there is no restriction at all on the behavior of the length of the residual vector during the cg-process. Hence, there are certainly examples where the residual vectors increase in length during the computation, as was stated earlier. This holds in spite of the fact that the error vector $h-x$ decreases in length at each step.

19. Numerical Illustrations

A number of numerical experiments have been made with the processes described in the preceding sections. A preliminary report on these experiments will be given in this section. In carrying out these experiments, no attempt was made to select those which favored the method. Normally, we selected those which might lead to difficulties.

In carrying out these experiments three sets of formulas for a_i, b_i were used in the symmetric case, namely,

$$a_i=\frac{(p_i,r_i)}{(p_i,Ap_i)}, \qquad b_i=-\frac{(r_{i+1},Ap_i)}{(p_i,Ap_i)}, \tag{19:1}$$

$$a_i=\frac{|r_i|^2}{(p_i,Ap_i)}, \qquad \frac{|r_{i+1}|^2}{|r_i|^2}, \tag{19:2}$$

$$a_i=\frac{|r_i|^2}{(p_i,Ap_i)d_i}, \quad b_i=\frac{|r_{i+1}|^2}{|r_i|^2}d_i, \quad d_i=1-b_{i-1}\frac{(p_{i-1},Ap_i)}{(p_i,Ap_i)}. \tag{19:3}$$

In the nonsymmetric case, we have used only the formulas

$$a_i=\frac{|A^*r_i|^2}{|Ap_i|^2}, \qquad b_i=\frac{|A^*r_{i+1}|^2}{|A^*r_i|^2}. \tag{19:4}$$

Our experience thus far indicates that the best results are obtained by the use of (19:1). Formulas (19:2) were about as good as (19:1) except for very ill conditioned matrices. Most of our experiments were carried out with the use of (19:2) because they are somewhat simpler than (19:1). Formulas (19:3) were designed to improve the relations

$$(r_i,r_{i+1})=0, \qquad (p_i,Ap_{i+1})=0, \tag{19:5}$$

which they did. Unfortunately, they disturbed the first of the relations

$$(p_i,r_{i+1})=0, \qquad (p_i,Ap_{i+1})=0. \tag{19:6}$$

A reflection of the geometrical interpretation of the method will convince one that one should strive to satisfy the relations (19:6) rather than (19:5). It is for this reason that (19:1) appears to be considerably superior to (19:3). In place of (19:2), one can use the formulas

$$a_i=\frac{|r_i|^2}{(p_i,Ap_i)}, \qquad b_i=\frac{|r_{i+1}|^2-(r_{i+1},r_i)}{|r_i|^2} \tag{19:2'}$$

to correct rounding off errors. A preliminary experiment indicates that this choice is better than (19:2) and is perhaps as good as (19:1).

A sufficient number of experiments have not been carried out as yet so as to determine the "best" formulas to be used. Our experiments do indicate that floating operations should be used whenever possible. We have also observed that the results in the $(n+1)$st and $(n+2)$nd iterations are normally far superior to those obtained in the nth iteration.

Example 1. This example was selected to illustrate the method of conjugate gradients in case there are no rounding off errors. The matrix A was chosen to be the matrix

$$A=\begin{Vmatrix} 1 & 2 & -1 & 1 \\ 2 & 5 & 0 & 2 \\ -1 & 0 & 6 & 0 \\ 1 & 2 & 0 & 3 \end{Vmatrix}.$$

If we select k to be the vector $(0,2,-1,1)$, the computation is simple. The results at each step are given in table 1.

Normally, the computation is not as simple as indicated in the preceding case. For example, if one selects the solution h to be the vector $(1,1,1,1)$, then k is the vector $(3,9,5,6)$. The results with $(0,0,0,0)$ as the initial estimate is given by table 2.

TABLE 1.

Step	Vector	Components of the vector				a_i	b_{i-1}
		1	2	3	4		
0	x_0	1	0	0	0		
	r_0	−1	0	0	0		
	p_0	−1	0	0	0		
	Ap_0	−1	−2	1	−1	1	
1	x_1	0	0	0	0		
	r_1	0	2	−1	1		6
	p_1	−6	2	−1	1		
	Ap_1	0	0	0	1	6	
2	x_2	−36	+12	−6	6		
	r_2	0	2	−1	−5		5
	p_2	−30	12	−6	0		
	Ap_2	0	0	−6	−6	5/6	
3	x_3	−61	22	−11	6		
	r_3	0	2	4	0		2/3
	p_3	−20	10	0	0		
	Ap_3	0	10	20	0	1/5	
4	x_4	−65	24	−11	6		

TABLE 2.

Step	Vector	α times components of vector				α
		1	2	3	4	
0	x_0	0	0	0	0	1
	r_0	3	9	5	6	1
	p_0	3	9	5	6	1
	Ap_0	22	63	27	39	1
1	x_1	453	1359	755	906	β_1
	r_1	−316	−495	933	123	β_1
	p_1	−1935	−2799	6461	1140	$\beta_1\gamma_1$
	Ap_1	−12854	−15585	40701	−4113	$\beta_1\gamma_1$
2	x_2	131702	419553	298277	304149	β_2
	r_2	1689	−34360	−27345	73483	β_2
	p_2	−116022	−1684085	−381080	3066641	$\beta_2\gamma_2$
	Ap_2	−66471	−2579187	−2140458	5685731	$\beta_2\gamma_2$
3	x_3	27589274	84526651	62344884	73103513	β_3
	r_3	542343	−188185	92550	−66019	β_3
	p_3	41725242	−15212135	6969632	−3788997	$\beta_3\gamma_3$
	Ap_3	542343	−188185	92550	−66019	$\beta_3\gamma_3$
4	x_4	1	1	1	1	1
	r_4	0	0	0	0	1

$\beta_1=1002,\quad \beta_2=326123,\quad \beta_3=69314516,$

$\gamma_1=\beta_1/151,\quad \gamma_2=\beta_2/8149,\quad \gamma_3=\beta_3/899615$

$a_0=1/\gamma_1,\quad a_1=\gamma_1/\gamma_2,\quad a_2=\gamma_2/\gamma_3,\quad a_3=\gamma_3$

$b_0=8149/\beta_1^2,\quad b_1=899615\beta_1\gamma_1/\beta_2^2,\quad b_2=380689\beta_2\gamma_2/\beta_3^2.$

TABLE 3.

					k	x_i	r_i
1 2 −1 1	1 0 0 0				3	3	0
2 5 0 2	0 1 0 0				9	0	3
−1 0 6 0	0 0 1 0				5	0	8
1 2 0 3	0 0 0 1				6	0	3
1 2 −1 1	1 0 0 0				3	−3	0
0 1 2 0	−2 1 0 0				3	3	0
0 2 5 1	1 0 1 0				8	0	2
0 0 1 2	−1 0 0 1				3	0	3
1 2 −1 1	1 0 0 0				3	7	0
0 1 2 0	−2 1 0 0				3	−1	0
0 0 1 1	5 −2 1 0				2	2	0
0 0 1 2	−1 0 0 1				3	0	1
1 2 −1 1	1 0 0 0				3	1	0
0 1 2 0	−2 1 0 0				3	1	0
0 0 1 1	5 −2 1 0				2	1	0
0 0 0 1	−6 2 −1 1				1	1	0

The system just described is particularly well suited for elimination. In case k is the vector (3, 9, 5, 6) the procedure described in section 12 yields the results given in table 3. In this table, we start with the matrices A and I. These matrices are transformed into the matrices P^*A and P^* given at the bottom of the table.

It is of interest to compare the error vectors $y_i=h-x_i$ obtained by the two methods just described with $k=(3, 9, 5, 6)$. The error $|y_i|$ is given in the following table.

$\|y_i\|$	cg-method	Elimination method
$\|y_0\|$	2.0	2.00
$\|y_1\|$	0.7	2.65
$\|y_2\|$	.67	4.69
$\|y_3\|$	.65	6.48
$\|y_4\|$	.0	0.00

In the cg-method $|y_i|$ decreases monotonically, while in the elimination method $|y_i|$ increases except for the last step.

Example 2. In this case the matrix A was chosen to be the matrix

.263879	.014799	.016836	.079773	−.020052	.011463
−.014799	.249379	.028764	.057757	−.056648	−.134493
.016836	.028764	.263734	−.033628	−.012128	.084932
.079773	.057757	−.033628	.215331	.090696	−.037489
−.020052	−.056648	−.012128	.090696	.324486	−.022484
.011463	−.134493	.084932	−.037489	−.022484	.339271

This matrix is a well-conditioned matrix, its eigenvalues lying on the range $\lambda_1=.6035\leqq\lambda\leqq\lambda_6=4.7357$. The computations were carried out on an IBM card programmed calculator with about seven significant figures. The results for the case in which x_0 is the origin and h the vector (1,1,1,1,1,1) are given in table 4.

Example 3. A good illustration of the effects of rounding can be obtained by study of an ill-conditioned system of three equations with three unknowns, namely, the system

$$6x+13y-17z=1$$
$$13x+29y-38z=2$$
$$-17x-38y+50z=-3,$$

whose solution is $x=1$, $y=-3$, $z=-2$. The system was constructed by E. Stiefel. The eigenvalues of A are given by the set

$$\lambda_1=.0588,\qquad \lambda_2=.2007,\qquad \lambda_3=84.7405.$$

The ratio of the largest to the smallest eigenvalue is very large: $\lambda_3/\lambda_1=1441$. The formulas (19:1), (19:2), and (19:3) were used to compute the solution,

TABLE 4.

Starting vector $k=(3.371, 1.2996, 3.4851, 3.7244, 3.0387, 2.412)$

Step	x_i	r_i	p_i	a_i, b_i
0	0	3.37100	3.37100	
	0	1.29960	1.29960	$a_0=3.092387$
	0	3.48510	3.48510	
	0	3.72440	3.72440	$b_0=0.02360156$
	0	3.03870	3.03870	
	0	2.41200	2.41200	
1	1.042444	−0.3176047	−0.02380454	
	.4018866	1.011922	.1042594	$a_1=3.487517$
	1.077728	.2194351	.03016873	
	1.151729	−.02954774	.005835219	$b_1=0.1411714$
	.9396836	−.3199108	−.02481941	
	.7458837	.03016107	.008708692	
2	.9594250	−0.009951160	−0.1331168	
	.7654931	.004267497	.1898594	$a_2=5.448597$
	1.1829418	−.01781102	−.1355206	
	1.1720790	−.009187803	−.08364038	$b_2=0.3997728$
	.8531255	.01514192	.1163813	
	.7762554	.03244676	.3367617	
3	.8868953	.1476560	.009443967	
	.8689395	.1042268	.01801273	$a_3=4.580482$
	1.1091023	−.1643885	−.02185659	
	1.1265069	−.0091902	−.004262730	$b_3=0.3769145$
	.9165367	.0633472	.01098733	
	.9597427	−.0907231	.004390484	
4	.930153	.08593514	.1215308	
	.951447	.00050406	.06839666	$a_4=5.464933$
	−1.008989	.05108757	−.03129307	
	−1.106982	−.12107954	−.1371464	
	.966864	−.03445640	.006956427	$b_4=0.2541540$
	.979853	.03607941	.05262778	
5	.996569	−.002365634	.007231114	
	.988825	.000616167	.02354492	$a_5=4.742589$
	.991887	.002508661	.01713337	
	1.032032	.003267702	−.06753326	
	.970666	.006006834	.06183634	$b_5=0$
	1.008614	.003155791	−.01818237	
6	.999998	−.00000252		
	.999991	−.00000084		
	1.000013	−.00002271		
	1.000004	.00000645		
	.999992	.00001636		
	.999991	.00000825		

keeping five significant figures at all times. For comparison, the computation was carried out also with 10 digits, using (19:2). The results are given in table 5. In the third iteration, formula (19:1) gave the better result. In the fourth iteration, formulas (19:1) and (19:2) were equally good, and superior to (19:3). The solution was also carried out by the elimination method using only five significant figures. The results are

	cg-method (19:1)	Elimination
x	.99424	1.06603
y	−2.99518	−3.00506
z	−1.99328	−2.00180

TABLE 5

$x_0=(1, 0, 0)$

Case	1 Formula (19:2)	2 Formula (19:3)	3 Formula (19:1)	1 with 10 digits
p_0	5	5	5	5
r_0	11	11	11	11
	−14	−14	−14	−14
c_0	.011804	.011804	.011804	.01180409347
x_1	.94098	.94098	.94098	.9409795326
	−.12984	−.12984	−.12984	−.1298450282
	.16526	.16526	.16526	.1652573086
r_1	.14856	.14856	.14856	.1485175838
	.18754	.18754	.18754	.1874503815
	.20021	.20021	.20021	.2003244444
r_1^2	.097325	.097325	.097325	.09732500125
b_0	.00028458	.00028458	.00027639	.0002845760270
p_1	.14998	.14998	.14994	.1499404639
	.19067	.19067	.19058	.1905807178
	.19623	.19623	.19634	.1963403800
a_1	7.0058	7.0393	7.0059	7.006740263
x_2	−.10975	−.11477	−.10948	−.1096143529
	−1.46564	−1.47202	−1.46502	−1.4651946170
	−1.20949	−1.21606	−1.21028	−1.2104487372
r_2	−.15045	−.15188	−.12747	−.1275876043
	.030400	.029648	.081611	.0814215368
	.085455	.084906	.018197	.0184025802
r_2^2	.030682	.031156	.023240	.02324671838
b_1	.31710	.31860	.23870	.2388565947
p_2	−.10289	−.10387	−.091679	−.0917733357
	.090861	.090685	.12710	.1269429981
	.14768	.14772	.065039	.0652997748
a_2	.047688	.047713	12.039	12.09069098
x_3	−.10484	−.05079	.99424	.9999886893
	−1.46997	−1.34651	−2.99518	−3.000023179
	−1.21653	−1.38837	−1.99328	−1.999968135
r_3	−.057616	−.058572	−.086092	−.0009108898
	.23615	.23643	−.19036	−.0020300857
	−.18543	−.18733	.25063	.0026663150
r_3^2	.093471	.094422	.10646	.000012060204
b_2	3.0287	3.0306	4.5804	.000518791676
p_3	−.36924	−.37336	−.50602	−.0009585010
	.51134	.51126	.39181	−.0019642287
	.26185	.26035	.54853	.0027001920
a_3	2.9923	2.9762	.011854	.0118007358
x_4	1.00004	1.06040	1.00024	1.0000000003
	−3.00005	−2.86812	−2.99982	−2.9999999997
	−2.00006	−2.16322	−1.99978	−1.9999999993
r_4	.00064408	.00014843	.00005181	0
	.0014340	−.00035647	.0000152	.00000000008
	−.0018823	.00094441	.0000364	.00000000002

In this case the results by the cg-method and elimination method appear to be equally effective. The cg-method has the advantage that an improvement can be made by taking one additional step.

This example is also a good illustration for the fact that the size of the residuals is not a reliable criterion for how close one is to the solution. In step 3 the residuals in case 1 are smaller than those of case 3, although the estimate in case 1 is very far from the right solution, whereas in case 3 we are close to it.

Further examples. The largest system that has been solved by the cg-method is a linear, symmetric system of 106 difference equations. The computation was done on the Zuse relay-computer at the Institute for Applied Mathematics in Zurich. The estimate obtained in the 90th step was of sufficient accuracy to be acceptable. [15]

[15] See C. Hochstrasser, "Die Anwendung der Methode der konjugierten Gradienten und ihrer Modifikationen auf die Lösung linearer Randwertprobleme," Thesis E. T. H., Zurich, Switzerland, in manuscript.

Several symmetric systems, some involving as many as twelve unknowns, have been solved on the IBM card programed calculator. In one case, where the ratio of the largest to the smallest eigenvalue was 4.9, a satisfactory solution has been obtained already in the third step; in another case, where this ratio was 100, one had to carry out fifteen steps in order to get an estimate with six correct digits. In these computations floating operations were not used. At all times an attempt was made to keep six or seven significant figures.

The cg-method has also been applied to the solution of small nonsymmetric systems on the SWAC. The results indicate that the method is very suitable for high speed machines.

A report on these experiments is being prepared at the National Bureau of Standards, Los Angeles.

LOS ANGELES, May 8, 1952.

Unitary Triangularization of a Nonsymmetric Matrix*

ALSTON S. HOUSEHOLDER

Oak Ridge National Laboratory, Oak Ridge, Tennessee

A method for the inversion of a nonsymmetric matrix, due to J. W. Givens, has been in use at Oak Ridge National Laboratory and has proved to be highly stable numerically but to require a rather large number of arithmetic operations, including a total of $n(n-1)/2$ square roots. Strictly, the method achieves the triangularization of the matrix, after which any standard method may be employed for inverting the triangle. The triangular form is brought about by means of a sequence of $n(n-1)/2$ plane rotations, whose product is an orthogonal matrix. Each rotation requires the extraction of a square root. The advantage in using the method lies in the fact that an orthogonal matrix is perfectly conditioned. Hence the condition of the matrix cannot deteriorate through successive transformations. In fact, if one defines the condition number of a matrix A to be [1]

$$\gamma(A) = \| A \| \, \| A^{-1} \|,$$

where the norm is the spectral norm, then for any orthogonal matrix W, $\gamma(W) = 1$ and the condition of any matrix is preserved under multiplication by an orthogonal matrix:

$$\gamma(WA) = \gamma(A).$$

To look at the matter another way, if

$$WA = R,$$

where R is an upper triangle, then

$$A^T A = A^T W^T W A = R^T R,$$

so that R is precisely the triangle one would obtain from the application of the Choleski square-root method to the positive definite matrix $A^T A$. It is, in fact, the matrix to which von Neumann and Goldstine [2] are led in their study of Gaussian elimination as applied to a positive definite matrix. To obtain the precise triangle that would result from Gaussian elimination with $A^T A$, one has only to remove as a factor the diagonal of R:

$$R = DU,$$

where U has a unit diagonal.

The purpose of the present note is to point out that the same result can be obtained with fewer arithmetic operations, and, in particular, for inverting a square matrix of order n, at most $2(n-1)$ square roots are required, instead of $n(n-1)/2$. For $n > 4$, this is a saving of $(n-4)(n-1)/4$ square roots.

* Received June, 1958

It is as easy to discuss the general complex case, and the method is based upon the almost self-evident

LEMMA. *For any vector $a \neq 0$, and any unit vector v, a unit vector u exists such that*

$$(I - 2uu^*)a = \| a \| v,$$

where $\| a \|$ represents the Euclidean norm: $\| a \|^2 = a^*a$. The computation requires two square roots and a single reciprocation.

The proof exhibits the computation. Let

$$\alpha = \| a \| > 0. \tag{1}$$

This represents one of the necessary square roots. It is required that

$$a - 2(u^*a)u = \alpha v.$$

Let

$$\mu = 2u^*a. \tag{2}$$

Then

$$\mu u = a - \alpha v, \tag{3}$$

$$\mu^2 = 2\alpha(\alpha - v^*a), \tag{4}$$

which accounts for the other square root. Clearly

$$\alpha - v^*a \geqq 0,$$

since α is the Euclidean length of a, and v^*a the projection of a upon the unit vector v. Hence α and μ are both real and can be taken non-negative. If $\alpha = v^*a$, then the lemma is verified with $u = 0$; otherwise take $\mu > 0$ defined by (4), and one verifies easily that u defined by (3) is effective. The single reciprocation necessary is in μ^{-1}.

Now let a be the first column of A and take $v = e_1$, the first column of the identity. Application of the lemma provides a unitary matrix

$$U_1 = I - 2u_1u_1^*$$

such that the first column of U_1A is null except in the first element. The result is equivalent to the application of $n - 1$ plane rotations, with one slight difference, that

$$\det (I - 2uu^*) = -1,$$

whence this transformation reverses the orientation of the configuration. One continues after suppressing the first row and first column of the transformed matrix. After $n - 1$ steps, at most, the matrix A is triangularized:

$$U = U_{n-1}U_{n-2} \cdots U_1, \qquad UA = R.$$

Evidently

$$\det A = (-1)^{n-1} \det R.$$

An interesting byproduct of the reduction is a simple proof of the theorem of Hadamard, that

$$|\det A| \leqq \prod_1^n \|a_i\|,$$

where a_i is the ith column of A. This results from the fact that the Euclidean norm is preserved under multiplication by a unitary matrix, and that each diagonal element of R is the norm of the projection of that column upon a certain subspace.

It is evident that when the reduction is applied to a matrix A of n columns and $N > n$ rows, there results again an upper triangle of non-null elements, and a trapezoid of zeros, the triangle representing the factorization of the normal matrix A^*A required for a least squares solution.

Returning to the case of a square matrix, if the vectors u_i are stored as generated and applied to the successive columns of the matrix, there are required the $n - 1$ reciprocations, $2(n - 1)$ square roots, and

$$(n - 1)(2n^2 + 5n + 9)/3 \doteq 2n^3/3$$

multiplications for the formation of R. In this count it is assumed that one forms α, μ^{-1}, u, and then each u^*a_i and uu^*a_i, $i > 1$.

Inversion of R requires $\binom{n+2}{3}$ multiplications; and if R^{-1} is multiplied by the U_i in reverse order, then

$$(n - 1)(4n^2 + 7n + 12)/6$$

multiplications are required. Altogether, for the formation of A^{-1}, the number of multiplications is

$$(3n^3 + 4n^2 + 5n - 10)/2 \doteq 3n^3/2,$$

as compared with approximately $5n^3/2$ multiplications required if one were to form $(A^*A)^{-1}A^*$.

Ky Fan has pointed out that half the square roots and all the reciprocations are evaded in the triangularization proper if one forms

$$\mu^2 I - 2(\mu u)(\mu u)^*.$$

Instead of R, a scalar multiple of R is formed in this way. This is feasible for floating-point computations, but probably not for fixed point.

If the matrix

$$A = (a_1, a_2, \cdots, a_n)$$

is scaled at the outset so that $\|a_i\| \leqq 1$ for every i, then all elements remain within range throughout the triangularization, since a unitary transformation leaves the Euclidean norm invariant. Hence no scaling problems arise in the actual triangularization. Moreover, when R^{-1} is formed, if this is similarly

scaled no further scaling is required in multiplying by the matrices U_i to form A^{-1}. Only in forming R^{-1} itself may intermediate scaling be required.

The unitary matrix employed here is obviously suggested by a somewhat more general one previously used [1]. A similar form is used by Steenrod [3] for theoretical purposes.

REFERENCES

1. A. S. Householder (1958), A class of methods for inverting matrices. *J. Soc. Ind. Appl. Math.* *6*, 189-195.
2. John von Neumann and H. H. Goldstine (1947), Numerical inverting of matrices of high order, *Bull. Am. Math. Soc.* *53:* 1021–99.
3. Norman E. Steenrod (1951), The topology of fibre bundles Princeton

BIT 18 (1978), 475–489

ORDER STARS AND STABILITY THEOREMS

G. WANNER, E. HAIRER and S. P. NØRSETT

Abstract.

This paper clears up to the following three conjectures:

1. The conjecture of Ehle [1] on the A-acceptability of Padé approximations to e^z, which is true;
2. The conjecture of Nørsett [5] on the zeros of the "E-polynomial", which is false;
3. The conjecture of Daniel and Moore [2] on the highest attainable order of certain A-stable multistep methods, which is true, generalizing the well-known Theorem of Dahlquist.

We further give necessary as well as sufficient conditions for A-stable (acceptable) rational approximations, bounds for the highest order of "restricted" Padé approximations and prove the non-existence of A-acceptable restricted Padé approximations of order greater than 6.

The method of proof, just looking at "order stars" and counting their "fingers", is very natural and geometric and never uses very complicated formulas.

1. How we came to order stars.

In the A-stability analysis of many classes of one-step methods, such as implicit Runge–Kutta, Collocation, Rosenbrock type or multiderivative formulas, for the numerical integration of stiff differential equations, there is the question if certain rational approximations to the exponential function $R(z)=P_k(z)/Q_j(z)$ are bounded by 1 on the entire left half plane $\operatorname{Re} z \leqq 0$.

In many cases $R(z)$ is a Padé approximation of order $k+j$ where

$$P_k(z) = 1+\frac{k}{j+k}z+\frac{k(k-1)}{(j+k)(j+k-1)}\frac{z^2}{2!}+\ldots+\frac{k(k-1)\ldots 1}{(j+k)\ldots(j+1)}\frac{z^k}{k!}$$

$$Q_j(z) = 1-\frac{j}{k+j}z+\frac{j(j-1)}{(k+j)(k+j-1)}\frac{z^2}{2!}-+\ldots\pm\frac{j(j-1)\ldots 1}{(k+j)\ldots(k+1)}\frac{z^j}{j!}.$$

Nørsett [5] defined the "E-polynomial"

$$E(y) = |Q_j(iy)|^2-|P_k(iy)|^2$$

in order to study the boundedness $|R(iy)|\leqq 1$ on the imaginary axis. He conjectured that, apart from a multiple zero at the origin, $E(y)$ has only real single roots. We computed the general formula (not easily)

Received April 17, 1978.

$$E(y) = \left[\frac{k!}{(k+j)!}\right]^2 \sum_{r=r_0}^{j} \frac{(-1)^{j-r}}{(j-r)!} \left[\prod_{q=1}^{j-r} (j-q+1)(k+q)(r-k-q)\right] y^{2r}$$

$$r_0 = \left[\frac{k+j+2}{2}\right]$$

for the (k,j) Padé approximation $(j \geqq k)$. It turned out that the conjecture of Nørsett is false (first counter example $j=6$, $k=0$) and only true for $k \to \infty$, $k-j = \text{const}$.

Numerical computations showed a close connection between the number of complex pairs of zeros of $E(y)$ and the number of poles of $R(z)$ in the left half plane. See Table 1 below:

Table 1

	Number of complex pairs of zeros of $E(y)$ with $\mathrm{Re}\,(y)>0$													Number of complex pairs of poles of $R(z)$ with $\mathrm{Re}\,(z)<0$											
	$k=0$	1	2	3	4	5	6	7	8	9	10	11			$k=0$	1	2	3	4	5	6	7	8	9	10	11
$j=4$	0	0	0	0	0									$j=4$	0	0	0	0	0							
$j=5$	0	0	0	0	0	0								$j=5$	1	0	0	0	0	0						
$j=6$	1	0	0	0	0	0	0							$j=6$	1	0	0	0	0	0	0					
$j=7$	1	0	0	0	0	0	0	0						$j=7$	1	1	0	0	0	0	0	0				
$j=8$	1	1	0	0	0	0	0	0	0					$j=8$	1	1	1	0	0	0	0	0	0			
$j=9$	1	1	1	0	0	0	0	0	0	0				$j=9$	1	1	1	0	0	0	0	0	0	0		
$j=10$	1	1	1	0	0	0	0	0	0	0	0			$j=10$	2	1	1	1	0	0	0	0	0	0	0	
$j=11$	2	1	1	1	0	0	0	0	0	0	0	0		$j=11$	2	1	1	1	1	0	0	0	0	0	0	0
$j=12$	2	1	1	1	1	0	0	0	0	0	0	0		$j=12$	2	2	1	1	1	1	0	0	0	0	0	0
$j=13$	2	2	1	1	1	1	0	0	0	0	0	0		$j=13$	2	2	1	1	1	1	1	0	0	0	0	0
$j=14$	2	2	1	1	1	1	0	0	0	0	0	0		$j=14$	2	2	2	1	1	1	1	0	0	0	0	0
$j=15$	2	2	2	1	1	1	1	0	0	0	0	0		$j=15$	2	2	2	2	1	1	1	1	0	0	0	0
$j=16$	2	2	2	2	1	1	1	1	0	0	0	0		$j=16$	3	2	2	2	1	1	1	1	1	0	0	0
$j=17$	3	2	2	2	1	1	1	1	1	0	0	0		$j=17$	3	2	2	2	2	1	1	1	1	1	0	0
$j=18$	3	2	2	2	2	1	1	1	1	1	0	0		$j=18$	3	3	2	2	2	2	1	1	1	1	1	0
$j=19$	3	3	2	2	2	2	1	1	1	1	0	0		$j=19$	3	3	3	2	2	2	1	1	1	1	1	0
$j=20$	3	3	3	2	2	2	1	1	1	1	1	0		$j=20$	3	3	3	2	2	2	2	1	1	1	1	1
$j=21$	3	3	3	2	2	2	2	1	1	1	1	1		$j=21$	4	3	3	3	2	2	2	2	1	1	1	1

In order to understand this regularity, we searched for curves that link the zeros of $E(y)$ to the poles of $R(z)$ in some manner. This led us to the definition of the "set A" as given below.

2. Properties of order stars.

First we study rational approximations to the exponential function

$$R(z) = \frac{P_k(z)}{Q_j(z)} \tag{1}$$

where $P_k(z)$ and $Q_j(z)$ are real polynomials of degree k and j respectively. We assume that $Q_j(0) \neq 0$ and that the fraction is reduced, so that P_k and Q_j have no common zeros.

Our aim is to study the *stability region* of R, namely

$$D = \{z \in \mathbf{C} \; ; \; |R(z)| \leqq 1\} . \tag{2}$$

One says that R is *A-acceptable* (and hence the corresponding method is A-stable), if

$$D \supset \mathbf{C}^- = \{z \in \mathbf{C} \; ; \; \mathrm{Re}\,(z) \leqq 0\} .$$

The main tool of this paper is to study instead the region

$$A = \{z \in \mathbf{C} \; ; \; |R(z)| > |e^z|\} = \{z \in \mathbf{C} \; ; \; |S(z)| > 1\} \tag{3}$$

where

$$S(z) = R(z)/e^z . \tag{4}$$

Note that $R(z)$ and $S(z)$ have the same zeros and the same poles.

PROPOSITION 1. *R is A-acceptable if and only if*
(i) *A has no intersection with the imaginary axis, and*
(ii) *R has no poles in $\mathbf{C}^-$.*

PROOF. This follows from the fact that on the imaginary axis, where $|e^z| = 1$, D and A are complementary, and from the maximum principle. ∎

Examples of the set A for Padé approximations are illustrated in the following Figure 1. Looking at these figures, one immediately understands for which reason Padé approximations are A-acceptable exactly if $j-2 \leqq k \leqq j$ (see Theorem 7 below).

The following Propositions 2, 3, and 4 are very elementary but fundamental for the discussion of A.

PROPOSITION 2. *Let the set B_r be defined as $B_r = \{t \in S^1 ; re^{it} \in A\}$. Then there is a number r_0 such that for $r \geqq r_0$ B_r is just an interval in S^1, which for $r \to \infty$ tends to $[\pi/2, 3\pi/2]$. So the border ∂A possesses only two branches that go to infinity.*

PROOF. Since e^x increases quicker and e^{-x} decreases quicker than any rational function, for t fixed,

$$\lim_{r \to \infty} \left| \frac{R(re^{it})}{e^{re^{it}}} \right| = \begin{cases} 0 & \text{if } -\dfrac{\pi}{2} < t < \dfrac{\pi}{2} \\ \infty & \text{if } \dfrac{\pi}{2} < t < \dfrac{3\pi}{2} \end{cases} .$$

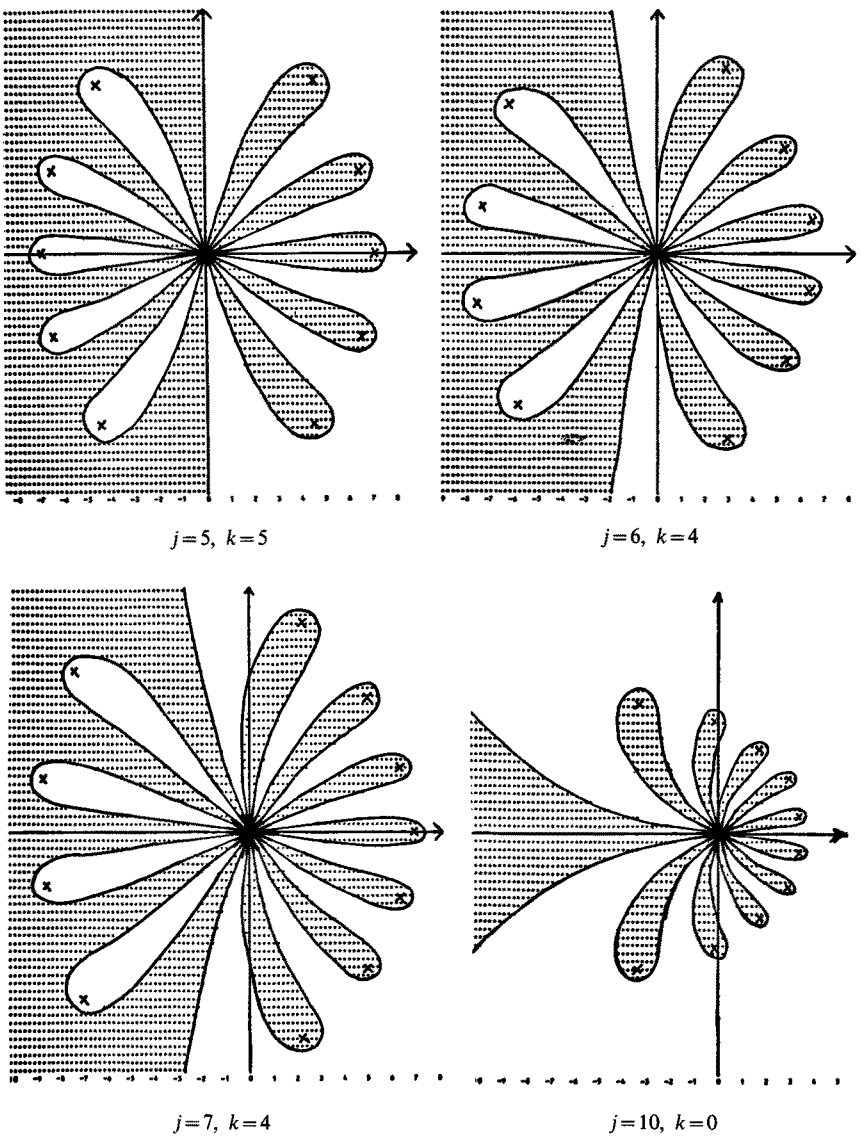


Figure 1. Order stars for Padé approximations.

Thus the boundary ∂A has at least two intersections with the circle $z=re^{it}$, $r>r_0$. In order to show that this circle has at most two intersections, we compute for $|R(re^{it})|=e^{r\cos t}$ the derivative

$$\frac{d}{dt}\left((e^{r\cos t})^2-|R(re^{it})|^2\right) = 2re^{2r\cos t}\left(-\sin t-\operatorname{Re}\left(ie^{it}\frac{R'(re^{it})}{R(re^{it})}\right)\right).$$

Since $|R'/R| \to 0$ for $r \to \infty$, this derivative is <0 for $0<t<\pi$ and >0 for $\pi<t<2\pi$ for large values of r. Hence there can only be two crossing points. ∎

The next Proposition relates the shape of A to the order of approximation.

One says that R is an *approximation of order p*, if there exists a constant $C \neq 0$ so that

$$e^z - R(z) = Cz^{p+1} + O(z^{p+2}) \quad \text{for } z \to 0 . \tag{5}$$

PROPOSITION 3. *R is an approximation of order p if and only if for $z \to 0$ A consists of $p+1$ sectors of width $\pi/(p+1)$, separated by $p+1$ sectors of the complement of A, each of the same width.*

PROOF. By (3) $z = re^{it}$ lies in A iff $|R(re^{it})e^{-r\cos t}| > 1$. We insert (5) to obtain for $r \to 0$ the condition

$$|1 - Ce^{-r\cos t} r^{p+1} e^{i(p+1)t}| > 1$$

which leads to

$$C \operatorname{Re}(e^{i(p+1)t}) = C\cos(p+1)t < 0 .$$

This is satisfied in consecutive intervals of length $\pi/(p+1)$. ∎

For this reason we use the name *order star* for the set A. We further call *fingers* the connected components of each of these sectors. If m sectors join together to one finger, we call it a *finger of multiplicity m*. The analogous sets for the complement CA we call *dual fingers* and *dual fingers of multiplicity m*.

PROPOSITION 4. *Each bounded finger of multiplicity m contains at least m poles of R (counted with their multiplicity); each bounded dual finger of multiplicity m contains at least m zeros of R.*

PROOF. Let $c(t)$, $t_0 \leqq t \leqq t_1$, be a parametrization of the positively oriented boundary of a finger F, $\boldsymbol{a} = (c_1'(t), c_2'(t))$ a tangent vector, $\boldsymbol{n} = (c_2'(t), -c_1'(t))$ an outside normal vector. We write $S(z) = r(x,y)e^{i\varphi(x,y)}$ $(z = x+iy)$ and since the modulus of S increases inside F, we have $\partial(\log r)/\partial \boldsymbol{n} < 0$. Now the Cauchy–Riemann differential equations in polar-coordinate-form

$$\frac{\partial(\log r)}{\partial x} = \frac{\partial \varphi}{\partial y}; \quad \frac{\partial(\log r)}{\partial y} = -\frac{\partial \varphi}{\partial x}$$

(see e.g. [13], p. 67) imply $\partial\varphi/\partial \boldsymbol{a} < 0$. Thus the argument of S decreases along c.

The difference between the number of zeros and the number of poles inside c is

$$Z - P = \frac{1}{2\pi i}\int_c \frac{S'(z)}{S(z)}\,dz = \text{number of rotations of } \arg(S) \text{ along } c .$$

If F is an m-fold finger, the boundary returns m times to the origin, thus at least m times arg (S) has the same direction, so the number of rotations is at least $-m$ and $P \geqq m$ (see Fig. 2 where $m=3$).

For dual fingers the argumentation is the same starting from $\partial(\log r)/\partial \boldsymbol{n} > 0$. ∎

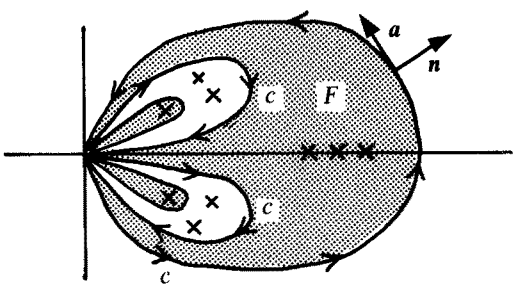


Figure 2.

3. Stability theorems for rational functions.

THEOREM 5. *If $R(z)=P_k(z)/Q_j(z)$ is A-acceptable and an approximation to e^z of order p, then*

$$p \leqq 2j \quad \text{and} \quad p \leqq 2k+2 .$$

PROOF. By Proposition 3 at least $[(p+1)/2]$ fingers of A start in the left half plane $\mathbf{C}^-$ (see Fig. 3 where $p+1=11$). These fingers cannot cross the imaginary axis (Prop. 1) and cannot be bounded (Prop. 4). So (Prop. 2) they all must collapse and include at least $[(p+1)/2]-1$ bounded dual fingers. So Prop. 4 gives that the total number of zeros of R satisfies $k \geqq [(p+1)/2]-1$ or $p \leqq 2k+2$.

The other inequality, which follows trivially from $p \leqq k+j$ and $k \leqq j$ could be proved similarly (see Theorem 12). ∎

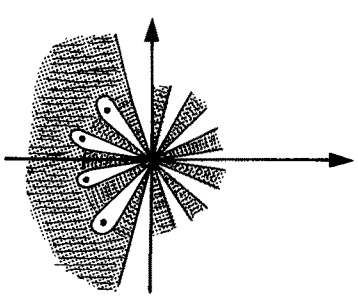


Figure 3.

Let us next give a simple proof of a result similar to that of Crouzeix and Ruamps [11].

THEOREM 6. *Suppose that for* $R(z)=P_k(z)/Q_j(z)$
(i) $p \geqq 2j-2$;
(ii) $\lim_{z\to\infty}|R(z)| \leqq 1$;
(iii) *the coefficients of* Q *have alternating signs.*
Then R *is A-acceptable.*

PROOF. It follows from (ii) that $k \leqq j$. Now the polynomial of degree $2j$ which is even because of symmetry

$$E(y) = |Q_j(iy)|^2 - |P_k(iy)|^2 = (|Q|+|P|)(|Q|-|P|)$$

satisfies $E(y)=O(y^{p+1})$ because of (5). Thus (i) gives us $E(y)=Ky^{2j}$ and (ii) implies $K \geqq 0$, so the order star can nowhere meet the imaginary axis.

At least $[(p+1)/2]$ fingers of A start in the right half plane $\mathbf{C}^+$ and must be bounded (Prop. 2). Hence there must be at least $[(p+1)/2]$ poles of S in $\mathbf{C}^+$.

Since from (i) $[(p+1)/2] \geqq j-1$, there can be at most one (and hence real) pole of S in $\mathbf{C}^-$, which is impossible because of (iii). ∎

THEOREM 7. ("*Theorem and Conjecture of Ehle*"). *Any Padé approximation* $R(z)=P_k(z)/Q_j(z)$ *to the exponential function is A-acceptable if and only if*

$$j-2 \leqq k \leqq j\,.$$

PROOF. Since Padé approximations have optimal order $p=k+j$, this is an immediate consequence of Theorems 5 and 6. ∎

4. The attainable order with real and multiple singularities.

For the treatment of large stiff systems it is preferable to use rational approximations for which the denominator can be factorized into real linear factors, since then the evaluation of $y_{n+1}=(Q_j(A))^{-1}P_k(A)y_n$ can be decomposed into a sequence of real linear equations. Another reason for the interest in these types of approximations is the fact that they are related to Rosenbrock type methods as well as semi-implicit or singly-implicit Runge–Kutta methods.

Let us give the following extension of a result of Nørsett and Wolfbrandt [6, 14].

THEOREM 8. *Let* $R(z)=P_k(z)/Q_j(z)$ *be such that* $Q_j(z)$ *has only* m *complex different zeros. If in addition* $Q_j(z)$ *possesses real zeros, then the order* p *satisfies*

$$p \leqq k+m+1\,.$$

If $Q_j(z)$ *has no real zeros at all, then we have*

$$p \leqq k+m\,.$$

PROOF. At most $\lambda = m+2$ of the $p+1$ dual fingers can be infinite (see Fig. 4). If there are no real singularities, $\lambda = m+1$. So at least $p+1-\lambda$ dual fingers are bounded and hence (Prop. 4) the number of zeros k must satisfy $k \geqq p+1-\lambda$. This gives the stated estimates. ∎

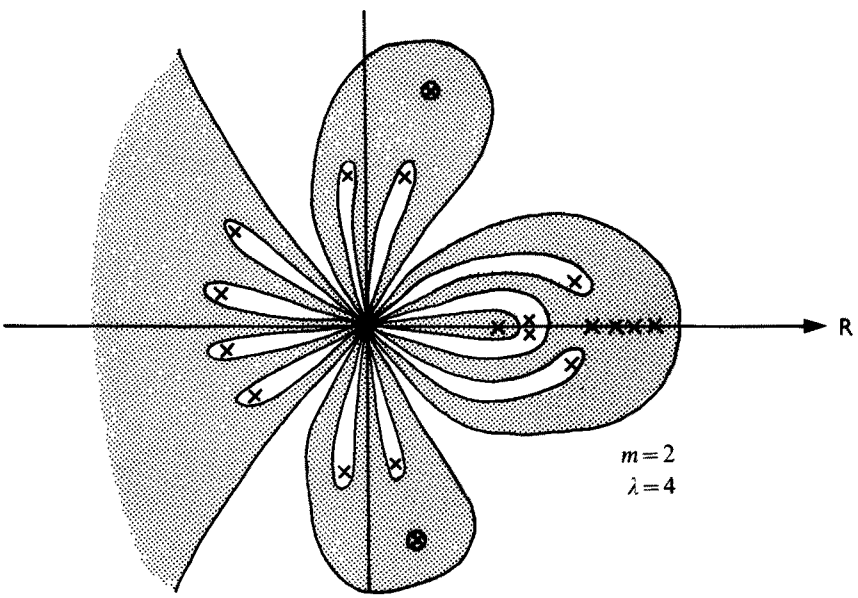


Figure 4.

5. A-acceptability of restricted Padé approximations.

In this section we study the particular class of approximations

$$R(z) = \left(\sum_{m=0}^{k} (-1)^k L_k^{(k-m)}(1/\gamma)(\gamma z)^m\right)/(1-\gamma z)^k \tag{7}$$

which are of order k for all $\gamma \in \mathbb{R}$ (see Nørsett [9] Corollary 2.1; L_k denotes the kth Laguerre polynomial). Here the denominator has just a k-fold zero. This makes them very useful for large sparse matrices.

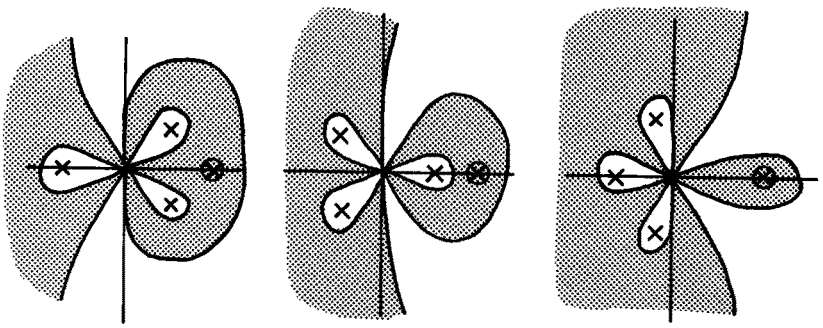
Since the error constant for this approximation is

$$C = (-1)^{k+1} \frac{\gamma^k}{k+1} L'_{k+1}\left(\frac{1}{\gamma}\right), \tag{8}$$

R has maximal order $k+1$ (see Theorem 8), when

$$\gamma = \gamma_\nu,\ L'_{k+1}\left(\frac{1}{\gamma_\nu}\right) = 0,\quad \frac{1}{\gamma_1} < \frac{1}{\gamma_2} < \ldots < \frac{1}{\gamma_k},\qquad \nu = 1,\ldots,k\,. \tag{9}$$

Figure 5 shows the order stars of these optimal approximations for the case $k=3$.



$\gamma=\gamma_1$ $\gamma=\gamma_2$ $\gamma=\gamma_3$

Figure 5.

PROPOSITION 9. *The order star for the restricted Padé approximation* (7) *with* γ *as in* (9) *contains just one bounded finger of multiplicity* $k-v+1$.

PROOF. Since there is only one k-fold singularity, all bounded fingers, say m, must collapse to one m-fold finger. Thus k of the $k+2$ dual fingers must be finite and each contains one zero of $P_k(z)$ (Prop. 4). So the order star is uniquely determined once we know how many zeros of $P_k(z)$ lie to the right of the two infinite branches of ∂A (Compare with Fig. 5 where this number, from left to right, is equal to 2, 1, and 0).

We write $P_k(z)$ by putting $\gamma z=w$ and $1/\gamma=\beta$ as

$$L_k(\beta)w^k+L_k'(\beta)w^{k-1}+L_k''(\beta)w^{k-2}+\ldots = 0\,. \tag{10}$$

When β increases say from $1/\gamma_v$ to $1/\gamma_{v+1}$, all w's depend continuously on β if $L_k(\beta)\neq 0$ and the zeros cannot change their position vis-à-vis ∂A. From properties of orthogonal polynomials there is exactly one $\tilde{\beta}$ in this interval where $L_k(\beta)$ has a single zero and thus exactly one solution of (10), say w_1, tends to infinity. To study its behavior near $\tilde{\beta}$ we write $L_k(\beta)=(\beta-\tilde{\beta})L_k'(\tilde{\beta})$ and neglect lower order terms in (10). Because $L_k'(\tilde{\beta})\neq 0$ this leads to $w_1\cong 1/(\tilde{\beta}-\beta)$, showing that for increasing β, one zero of $P_k(z)$ tends at the right to $+\infty$ and comes back at the left from $-\infty$. So m has decreased by one and there is a constant M such that $m=M-v$ ($v=1,\ldots,k$). Finally the overall inequality $1\leqq m\leqq k$ (Prop. 4) gives $M=k+1$. ∎

Figure 6 illustrates how the order star for $k=3$ changes when γ varies from γ_1 to γ_2.

THEOREM 10. *The restricted Padé approximation* (7) *with optimal order* (9) *can only be A-acceptable if*

$$q \leqq v \leqq q+1 \quad \textit{for } k=2q+1$$
$$q = v \quad \textit{for } k=2q\,.$$

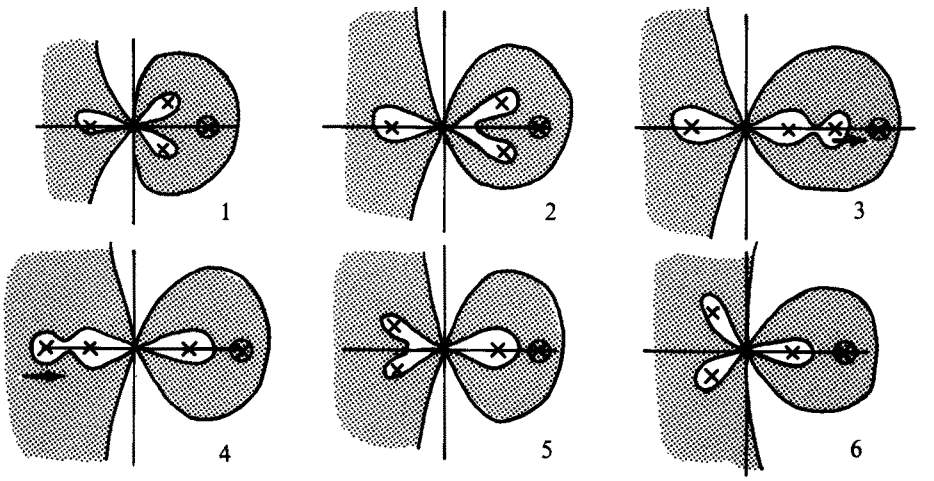


Figure 6.

PROOF. $k+2$ fingers start at the origin. According to proposition 9, $k-v+1$ go to the right, $v+1$ go to the left. At most $[(k+3)/2]$ fingers can start in $\mathbf{C}^+$ or in $\mathbf{C}^-$. So if not

$$\left[\frac{k+3}{2}\right] \geqq k-v+1 \quad \text{and} \quad \left[\frac{k+3}{2}\right] \geqq v+1\,,$$

some fingers must cross the imaginary axis and R cannot be A-acceptable. ∎

LEMMA. *The approximation* (7) *and* (9) *can only be A-acceptable if*

$$k \geqq \begin{cases} 1 & \text{if } v=1 \\ 5 & \text{if } v=2 \\ 9 & \text{if } v=3 \\ 6v-10 & \text{if } v \geqq 4\,. \end{cases}$$

PROOF. From (7) we have $\lim_{z\to\infty}|R(z)| = |L_k(1/\gamma_v)|$. So it is necessary that $|L_k(1/\gamma_v)| \leqq 1$. For small values of k the stated bounds are obtained by numerical computations (see Nørsett [9a], p. A.7). For large values we use Hilb's asymptotic formula (see Szegö [12], page 193)

$$L_k(x) = \frac{e^{x/2}}{(\pi)^{1/2}(xk)^{1/4}}\left(\cos\left(2(xk)^{1/2}-\pi/4\right)+(xk)^{-1/2}O(1)\right)$$

which gives for the vth extremum point of L_{k+1} the approximation

$$x_v = (\pi^2(v+1/4)^2)/4(k+1)\,.$$

By inserting this into the above formula for L_{k+1} and using $L_{k+1}(\gamma_\nu^{-1})=L_k(\gamma_\nu^{-1})$, some modifications show that $|L_k(x_\nu)|>1$, if not

$$k+1 \geqq \pi^2(\nu+1/4)^2/\big(4\log(\pi^2(\nu+1/4)/2)\big)\,.$$

Computing this expression for different value of ν, one obtains the above stated estimates except for the case $\nu=1$, where the asymptotic formula is not yet sufficiently close. ∎

THEOREM 11. *The restricted Padé approximation* (7) *with optimal order* (9) *is A-acceptable if and only if*

$$\begin{array}{lllll} k= & 1 & 2 & 3 & 5 \\ \nu= & 1 & 1 & 1 & 2 \end{array}\,.$$

PROOF. These are the only possible cases left by the two foregoing results. The A-acceptability of these remaining cases has been proved in [10]. ∎

6. The multistep case.

The stability analysis of multistep methods, multistep-multiderivative formulas, PC-schemes, composite or cyclic multistep methods, multistep Runge–Kutta methods etc. (see e.g. [3, 4, 7, 8]) leads to a characteristic algebraic equation

$$Q(z,R) := Q_0(z)R^k+Q_1(z)R^{k-1}+\ldots+Q_k(z) = 0 \tag{11}$$

for the eigenvalues of the resulting difference equation.

We suppose

$$Q(z,R) \text{ irreducible},\ Q_0(0)\neq 0,\ \frac{\partial Q}{\partial R}(0,1)\neq 0,\ \deg Q_r\leqq j \qquad (r=0,1,\ldots,k)\,. \tag{12}$$

For linear multistep methods all $Q_r(z)$ are linear, hence in this case $j=1$. For other classes of methods j indicates the number of used derivatives or stages while k is the number of steps involved in the method. For $k=1$ we obtain a rational R as considered in the foregoing sections.

For k greater than 1 the solution of (11) becomes a multivalued function. We thus introduce the corresponding Riemann surface M on which R becomes singlevalued again. With the exception of some pathological cases M can most easily be written as

$$M = \{(z,w)\in \mathbb{C}^2\ ;\ Q(z,w)=0\}$$

with the projections

$$\begin{array}{ccc} M & \xrightarrow{R} & \mathbb{C} \\ \pi\downarrow & & \\ \mathbb{C} & & \end{array} \qquad \begin{array}{ccc} (z,w) & \xmapsto{R} & w \\ \pi\downarrow & & \\ z & & \end{array}$$

For each $z \in \mathbf{C}$, the inverse $\pi^{-1}(z)$ of the covering projection consists in general of k points $(z, w_1), \ldots, (z, w_k)$ where $w_1, \ldots, w_k$ are the k solutions of (11). So M is an orientable surface consisting locally of k sheets lying above $\mathbf{C}$ and interacting in a finite number of branch points, i.e. the finite number of points where (11) has multiple solutions. (See e.g. [13], chapter V).

Again we define the *stability domain* as

$$D = \{z \in \mathbf{C} \ ; \ |R(\pi^{-1}(z))| \leqq 1 \text{ for all inverses and } <1 \text{ at branch points}\}$$

and call the function R *A-acceptable* if D contains $\mathbf{C}^-$.

The *order star* is now a subset of M

$$A = \{z \in M \ ; \ |R(z)| > |e^{\pi(z)}|\} = \{z \in M \ ; \ |S(z)| > 1\}$$

where

$$S(z) = R(z)/e^{\pi(z)} \quad \text{for } z \in M .$$

We suppose that the methods considered are *consistent*, so that for $z=0$ $R=1$ is a solution of (11) and by (12) this solution is simple in a neighborhood of the origin. Thus it can be continued in a neighbourhood of 0 to a *principal solution* $R_1(z)$ defined on its *principal sheet.*

The method is of *order p*, if there exists a constant $C \neq 0$ such that for the principal solution

$$e^z - R_1(z) = Cz^{p+1} + O(z^{p+2}) \quad \text{for } z \to 0 . \tag{13}$$

This is, because of

$$Q(z, e^z) = Q(z, e^z) - Q(z, R_1(z)) = \frac{\partial Q}{\partial R}(0,1) C z^{p+1} + O(z^{p+2})$$

and (12) equivalent to the usual definition.

Now the Propositions 1–4 remain true with the following modifications:

PROPOSITION 1. *R is A-acceptable if and only if*

(i) *A has no intersection with* $\pi^{-1}(i\mathbf{R})$ *and* $\bar{A}$ *never touches* $\pi^{-1}(i\mathbf{R})$ *in a branch point;*

(ii) *R has no poles in* $\pi^{-1}(\mathbf{C}^-)$, *i.e.* $Q_0(z)$ *has no zeros in* $\mathbf{C}^-$.

The proof is trivial in one direction (the one which is actually used below) and uses the maximum principle for Riemann surfaces for the other direction.

PROPOSITION 2. *The same statement now holds on each sheet. The sheets of M outside all finite branch points are either all separated or, if* ∞ *is itself a branch point, some of them spiral together in the usual way. The formula* $|R'/R| \to 0$ *for* $z \to \infty$, *used in the proof, is best seen from the expansion*

$$R(z) = \sum_{q=q_0}^{\infty} a_q z^{-q/m} .$$

PROPOSITION 3. *It remains true, but this time on the principal sheet only.*

We again say *fingers* and *multiple fingers* for the connected components of these sectors in M.

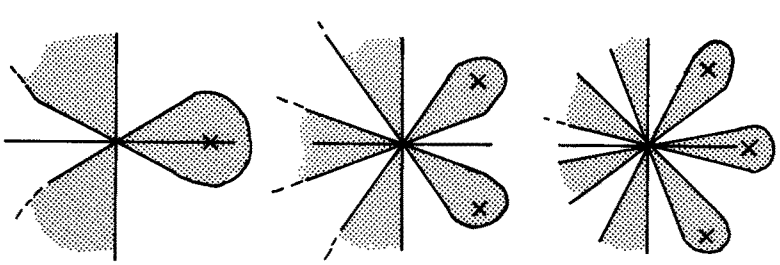
PROPOSITION 4. *It remains the same.*

For the proof one has to take care that the border of F can be composed of *several* closed loops, since M may be no longer simply connected. So the integration of $(1/2\pi i)\int_c S'(z)/S(z)\,dz$ has to be extended over all of these loops and the sum of the integrals is $\leqq -m$ since in total m times one of these loops visits the origin.

THEOREM 12 ("Conjecture of Daniel and Moore"). *If R is A-acceptable and satisfies* (11) *and* (12), *then $p \leqq 2j$ and* sign $(C) = (-1)^j$ *for $p = 2j$.*

PROOF. If R is of order p, at least $[(p+1)/2]$ sectors start in $\mathbf{C}^+$ on the principal sheet (Prop. 3, Fig. 7). Its fingers cannot cross $\pi^{-1}(i\mathbf{R})$ and thus must be bounded (Prop. 1 and 2). The total number of poles available on M is j, the degree of $Q_0(z)$. So by Propositon 4 $[(p+1)/2] \leqq j$ or $p \leqq 2j$.

The second assertion can be seen from Figure 7 and the fact (see the proof of Prop. 3) that the real positive axis (for z small) belongs to A iff $C < 0$. ∎



$p=2$ (the Dahlquist case) $p=4$ $p=6$

Figure 7.

THEOREM 13 (Second part of the conjecture). *The error constant C of an A-acceptable R of maximal order $p = 2j$ satisfies*

$$|C| \geqq |\tilde{C}|$$

where

$$\tilde{C} = (-1)^j \frac{(j!)^2}{(2j)!(2j+1)!}$$

is the error constant of $R_{jj}(z)$, the diagonal Padé approximation of order $2j$.

PROOF. Subtracting (5) from (13) we get

$$R_{jj}(z)-R_1(z) = (C-\tilde{C})z^{2j+1}+O(z^{2j+2}) . \tag{14}$$

So we consider $R_1(z)$ as approximation of order $2j$ to $R_{jj}(z)$ and look at the *relative* order star

$$B = \{z \in M \; ; \; |R_1(z)|>|R_{jj}(\pi(z))|\} = \{z \in M \; ; \; |S(z)|>1\}$$

where

$$S(z) = R_1(z)/R_{jj}(\pi(z)) .$$

Since $|R_{jj}(iy)|=1$ and R_1 is A-acceptable, B cannot cross $\pi^{-1}(i\mathbb{R})$ and since $R_{jj} \neq 0$ on $\mathbb{C}^+$ (see e.g. Fig. 1 or Theorem 7), $S(z)$ has no more poles on $\pi^{-1}(\mathbb{C}^+)$ than $R(z)$.

In spite of the fact that the fingers in $\mathbb{C}^+$ are no longer necessarily bounded, Proposition 4 applies as well, since both R_1 and R_{jj} are A-acceptable and the point ∞ is no longer a singularity.

Suppose now $|C|<|\tilde{C}|$ and, for example, j even. It follows from (14) that for z real, small and positive, $R_{jj}<R_1$, and hence the positive real axis for z small belongs to B. We thus have the situation contrary to that in Figure 7, so that now $j+1$ fingers start in $\mathbb{C}^+$ requiring $j+1$ poles of R_1, a contradiction. ∎

Remarks.

1. Several authors have proved parts of this conjecture. Genin [3] arrives at different results, since his methods are not stable in the usual sense. He has been corrected later by Jeltsch ([4] and other papers).

2. In the multistep case it is no longer easy to derive similar conditions on the number of zeros as in Theorem 5, since here M is in general not simply connected so that two collapsing fingers do not necessarily contain a bounded dual finger. A counter-example is Gear's 2nd order backward difference formula which leads to $(3/2-z)R^2-2R+1/2=0$. Here R has no zero at all.

3. The authors wish to acknowledge a long discussion with Mr. B. Kaup on Riemann surfaces.

REFERENCES

1. B. L. Ehle, *On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems*, Thesis 1969, CSRR2010, Department of Computer Science, University of Waterloo, Ontario, Canada, conjecture 3.1.
Also see a paper of similar title in SIAM J. Math. Anal. 4 (1973), 671–680.
2. J. W. Daniel and R. E. Moore, *Computation and theory in ordinary differential equations*, Freeman and Co., 1970, page 80.
3. Y. Genin, *An algebraic approach to A-stable linear multistep-multiderivative integration formulas*, BIT 14 (1974), 382–406.

4. R. Jeltsch, *Note on A-stability of multistep-multiderivative methods*, BIT 16 (1976), 74–78.
5. S. P. Nørsett, *C-polynomials for rational approximations to the exponential function*, Numer. Math. 25 (1975), 39–56.
6. S. P. Nørsett and A. Wolfbrandt, *Attainable order of rational approximations to the exponential function with only real poles*, BIT 17 (1977), 200–208.
7. W. B. Rubin, *A-stability and composite multistep methods*, Ph. D. Thesis, EE Dept., Syracuse University, New York 1973.
8. H. J. Stetter, *Analysis of discretization methods in ordinary differential equations*, Springer-Verlag 1973.
9. S. P. Nørsett, *Restricted Padé approximations to the exponential function*, to appear in SIAM J. Numer. Anal. 1978.

9a. *id.*, similar title, Math. and Computation, No. 474, Trondheim.

10. S. P. Nørsett, *One step methods of Hermite type for numerical integration of stiff systems*, BIT 14 (1974), 63–77.
11. M. Crouzeix and F. Ruamps, *On rational approximations to the exponential*, R.A.I.R.O. Analyse Numérique 11, n° 3 (1977), 241–243.
12. G. Szegö, *Orthogonal polynomials*, AMS 1939.
13. Behnke-Sommer, *Theorie der analytischen Funktionen*, Springer 1962.
14. A. Wolfbrandt, *A note on a recent result of rational approximations to the exponential function*, BIT 17 (1977), 367–368.

DÉPARTEMENT DE MATHÉMATIQUES
CASE POSTALE 124
2-4 RUE DU LIÈVRE
CH-1211 GENÈVE 24
SUISSE

A rapidly convergent descent method for minimization

By R. Fletcher and M. J. D. Powell

A powerful iterative descent method for finding a local minimum of a function of several variables is described. A number of theorems are proved to show that it always converges and that it converges rapidly. Numerical tests on a variety of functions confirm these theorems. The method has been used to solve a system of one hundred non-linear simultaneous equations.

1. Introduction

We are concerned in this paper with the general problem of finding an unrestricted local minimum of a function $f(x_1, x_2 \ldots, x_n)$ of several variables $x_1, x_2, \ldots, x_n$. We suppose that the function of interest can be calculated at all points. It is convenient to group functions into two main classes according to whether the gradient vector $g_i = \partial f/\partial x_i$ is defined analytically at each point or must be estimated from the differences of values of f. The method described in this paper is applicable to the case of a defined gradient. For the other case a useful method and general discussion are given by Rosenbrock (1960).

Methods using the gradient include the classical method of steepest descents (Courant, 1943; Curry, 1944; and Householder, 1953), Levenberg's modification of damped steepest descents (1944), a somewhat similar variation due to Booth (1957), the conjugate gradient method of Hestenes and Stiefel (1952), similar methods of Martin and Tee (1961), the "Partan" method of Shah, Buehler and Kempthorne (1961), and a method due to Powell (1962). In this paper we describe a powerful method with rapid convergence which is based upon a procedure described by Davidon (1959). Davidon's work has been little publicized, but in our opinion constitutes a considerable advance over current alternatives. We have made both a simplification by which certain orthogonality conditions which are important to the rate of attaining the solution are preserved, and also an improvement in the criterion of convergence.

Because, near the minimum, the second-order terms in the Taylor series expansion dominate, the only methods which will converge quickly for a general function are those which will guarantee to find the minimum of a general quadratic speedily. Only the latter four methods of the last paragraph do this, and the procedures of Hestenes and Stiefel and of Martin and Tee are not applicable to a general function. Of course the generalized Newton–Raphson method (Householder, 1953) has fast convergence eventually, but it requires second derivatives of the function to be evaluated, and frequently fails to converge from a poor approximation to the minimum. The method described has quadratic convergence and is superior to "Partan" and to Powell's method, both in that it makes use of information determined by previous iterations and also in that each iteration is quick and simple to carry out. Furthermore, it yields the curvature of the function at the minimum, so excellent tests for convergence and estimates of variance can be made.

The method is given an elegant theoretical basis, and proofs of stability and of the rate of convergence are included. The results of numerical tests with a variety of functions are also given. These confirm that the method is probably the most powerful general procedure for finding a local minimum which is known at the present time.

2. Notation

It is convenient to describe the method in terms of the Dirac bra-ket notation (Dirac, 1958) applied to real vectors. In this notation the column vector $(x_1, x_2 \ldots, x_n)$ is written as $|x\rangle$. The row vector with these same elements is denoted by $\langle x|$. The scalar product of $\langle x|$ and $|y\rangle$ is written $\langle x|y\rangle$ and we may note that

$$\langle x|y\rangle \equiv \sum_i x_i y_i \equiv \sum_i y_i x_i \equiv \langle y|x\rangle.$$

The construction $|x\rangle\langle y|$, however, denotes a linear operator with matrix elements $D_{ij} = x_i y_j$ so that $|x\rangle\langle y| \not\equiv |y\rangle\langle x|$. A general linear operator or matrix will be denoted by a capital letter in bold type. It then follows that say $\boldsymbol{H}|x\rangle$ is a column vector, $\langle x|\boldsymbol{H}$ is a row vector and $\langle x|\boldsymbol{H}|y\rangle$ is a scalar.

We reserve f to denote the function of interest, $|x\rangle$ to denote its arguments and $|g\rangle$ to denote its gradient.

Hence the standard quadratic form in n dimensions

$$f = f_0 + \sum_{i=1}^{n} a_i x_i + \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} G_{ij} x_i x_j$$

becomes in this notation

$$f = f_0 + \langle a|x\rangle + \tfrac{1}{2}\langle x|\boldsymbol{G}|x\rangle \tag{1}$$

and also

$$|g\rangle = |a\rangle + \boldsymbol{G}|x\rangle. \tag{2}$$

3. The method

If we consider the quadratic form (1) then, given the matrix $G_{ij} = \partial^2 f/\partial x_i \partial x_j$, we can calculate the displacement between the point $|x\rangle$ and the minimum $|x_0\rangle$ as

$$|x_0\rangle - |x\rangle = -\boldsymbol{G}^{-1}|g\rangle. \tag{3}$$

In this method the matrix $\boldsymbol{G}^{-1}$ is not evaluated directly;

Downloaded from https://academic.oup.com/comjnl/article-abstract/6/2/163/364776 by guest on 29 September 2019

instead a matrix $\boldsymbol{H}$ is used which may initially be chosen to be any positive definite symmetric matrix. This matrix is modified after the ith iteration using the information gained by moving down the direction

$$|s^i\rangle = -\boldsymbol{H}^i|g^i\rangle \tag{4}$$

in accordance with (3). The modification is such that $|\sigma^i\rangle$, the step to the minimum down the line

$$|x\rangle = |x^i\rangle + \lambda|s^i\rangle$$

is effectively an eigenvector of the matrix $\boldsymbol{H}^{i+1}\boldsymbol{G}$. This ensures that as the procedure converges $\boldsymbol{H}$ tends to $\boldsymbol{G}^{-1}$ evaluated at the minimum.

It is convenient to take the unit matrix initially for $\boldsymbol{H}$ so that the first direction is down the line of steepest descent.

Let the current point be $|x^i\rangle$ with gradient $|g^i\rangle$ and matrix $\boldsymbol{H}^i$. The iteration can then be stated as follows.

Set $|s^i\rangle = -\boldsymbol{H}^i|g^i\rangle$.

Obtain α^i such that $f(|x^i\rangle + \alpha^i|s^i\rangle)$ is a minimum with respect to λ along $|x^i\rangle + \lambda|s^i\rangle$ and $\alpha^i > 0$. We will prove that α^i can always be chosen to be positive.

Set $$|\sigma^i\rangle = \alpha^i|s^i\rangle. \tag{5}$$

Set $$|x^{i+1}\rangle = |x^i\rangle + |\sigma^i\rangle.$$

Evaluate $f(|x^{i+1}\rangle)$ and $|g^{i+1}\rangle$ noting that $|g^{i+1}\rangle$ is orthogonal to $|\sigma^i\rangle$, that is

$$\langle\sigma^i|g^{i+1}\rangle = 0. \tag{6}$$

Set $$|y^i\rangle = |g^{i+1}\rangle - |g^i\rangle.$$

Set $$\boldsymbol{H}^{i+1} = \boldsymbol{H}^i + \boldsymbol{A}^i + \boldsymbol{B}^i \tag{7}$$

where $$\boldsymbol{A}^i = \frac{|\sigma^i\rangle\langle\sigma^i|}{\langle\sigma^i|y^i\rangle}$$

and $$\boldsymbol{B}^i = \frac{-\boldsymbol{H}^i|y^i\rangle\langle y^i|\boldsymbol{H}^i}{\langle y^i|\boldsymbol{H}^i|y^i\rangle}.$$

Set $i = i + 1$ and repeat.

There are two obvious and very useful ways of terminating the procedure, and they arise because $|s^i\rangle$ tends to the correction to $|x^i\rangle$. One is to stop when the predicted absolute distance from the minimum $\langle s^i|s^i\rangle^{\frac{1}{2}}$ is less than a prescribed amount, and the other is to finish when every component of $|s^i\rangle$ is less than a prescribed accuracy. Two additional safeguards have been found necessary in automatic computer programs. The first is to work through at least n (the number of variables) iterations, and the second is to apply the tests to $|\sigma^i\rangle$ as well as to $|s^i\rangle$.

The method of obtaining the minimum along a line is not central to the theory. The suggested procedure given in the Appendix, which uses cubic interpolation, is based on that given by Davidon, and has been found satisfactory.

We shall now show that the process is stable, and demonstrate that if $f(|x\rangle)$ is the quadratic form (1) then the procedure terminates in n iterations. We shall also explain the theoretical justification for the manner in which the matrix $\boldsymbol{H}$ is modified.

4. Stability

It is usual for descent methods to be stable because one ensures that the function to be minimized is decreased by each step. It will be shown in this Section that the direction of search $|s^i\rangle$, defined by equation (4), is downhill, so α^i can always be chosen to be positive. Because $|g^i\rangle$ is the direction of steepest ascent, the direction $|s^i\rangle$ will be downhill if and only if

$$-\langle s^i|g^i\rangle = \langle g^i|\boldsymbol{H}|g^i\rangle$$

is positive. We wish the direction of search to be downhill for all possible $|g^i\rangle$ so we must prove that $\boldsymbol{H}^i$ is positive definite. Because $\boldsymbol{H}^0$ has been chosen to be positive definite an inductive argument will be used.

In the proof it is assumed that $\boldsymbol{H}^i$ is positive definite and consequently that α^i is positive. It is proved that, for any $|x\rangle$, $\langle x|\boldsymbol{H}^{i+1}|x\rangle \geqslant 0$. We may define $|p\rangle = (\boldsymbol{H}^i)^{\frac{1}{2}}|x\rangle$ and $|q\rangle = (\boldsymbol{H}^i)^{\frac{1}{2}}|y^i\rangle$ as the square root of a positive definite matrix exists.

From (7)

$$\begin{aligned}\langle x|\boldsymbol{H}^{i+1}|x\rangle &= \langle x|\boldsymbol{H}^i|x\rangle + \frac{\langle x|\sigma^i\rangle\langle\sigma^i|x\rangle}{\langle\sigma^i|y^i\rangle} \\ &\quad - \frac{\langle x|\boldsymbol{H}^i|y^i\rangle\langle y^i|\boldsymbol{H}^i|x\rangle}{\langle y^i|\boldsymbol{H}^i|y^i\rangle} \\ &= \frac{\langle p|p\rangle\langle q|q\rangle - \langle p|q\rangle^2}{\langle q|q\rangle} + \frac{\langle x|\sigma^i\rangle^2}{\langle\sigma^i|y^i\rangle} \\ &\geqslant \langle x|\sigma^i\rangle^2/\langle\sigma^i|y^i\rangle\end{aligned}$$

on account of Schwartz's inequality.

But
$$\begin{aligned}\langle\sigma^i|y^i\rangle &= \langle\sigma^i|g^{i+1}\rangle - \langle\sigma^i|g^i\rangle \\ &= -\langle\sigma^i|g^i\rangle \text{ from (6)} \\ &= \alpha^i\langle g^i|\boldsymbol{H}^i|g^i\rangle \text{ from (4) and (5)} \\ &> 0.\end{aligned}$$

Hence $\langle x|\boldsymbol{H}^{i+1}|x\rangle > 0$ for all non-trivial $|x\rangle$. Therefore $\boldsymbol{H}^{i+1}$ is positive definite and the procedure is stable.

5. Quadratic convergence

In this Section it is assumed that f is the quadratic form (1) and that f has a well defined minimum. It is proved that in this case the method finds the minimum in n iterations. The method of proof is to show that $|\sigma^0\rangle, |\sigma^1\rangle, \ldots, |\sigma^k\rangle$ are linearly independent eigenvectors of $\boldsymbol{H}^{k+1}\boldsymbol{G}$ with eigenvalue unity. Therefore it will follow that $\boldsymbol{H}^n\boldsymbol{G}$ is the unit matrix.

By definition

$$\begin{aligned}|y^i\rangle &= |g^{i+1}\rangle - |g^i\rangle \\ &= \boldsymbol{G}|x^{i+1}\rangle - \boldsymbol{G}|x^i\rangle \quad \text{from (2)} \\ &= \boldsymbol{G}|\sigma^i\rangle.\end{aligned} \tag{8}$$

Downloaded from https://academic.oup.com/comjnl/article-abstract/6/2/163/364776 by guest on 29 September 2019

Also from (8)

$$\begin{aligned} H^{i+1}G|\sigma^i\rangle &= H^{i+1}|y^i\rangle \\ &= H^i|y^i\rangle + |\sigma^i\rangle - H^i|y^i\rangle \text{ by using (7)} \\ &= |\sigma^i\rangle. \end{aligned} \tag{9}$$

The equations

$$\langle\sigma^i|G|\sigma^j\rangle = 0 \qquad 0 \leqslant i < j < k \tag{10}$$

and $$H^kG|\sigma^i\rangle = |\sigma^i\rangle \quad 0 \leqslant i < k \tag{11}$$

will now be considered. It is clear from (9) that they are true if $k = 1$. It will be proved that if they are true for k they are true for $k + 1$.

From (2)

$$\begin{aligned} |g^k\rangle &= |a\rangle + G|x^k\rangle \\ &= |a\rangle + G(|x^{i+1}\rangle + |\sigma^{i+1}\rangle + |\sigma^{i+2}\rangle + \ldots \\ &\qquad + |\sigma^{k-1}\rangle) \\ &= |g^{i+1}\rangle + G(|\sigma^{i+1}\rangle + \ldots + |\sigma^{k-1}\rangle). \end{aligned}$$

Therefore from (10) and (6)

$$\begin{aligned} \langle\sigma^i|g^k\rangle &= \langle\sigma^i|g^{i+1}\rangle \\ &= 0 \qquad 0 \leqslant i < k. \end{aligned} \tag{12}$$

Hence from (11)

$$\langle\sigma^i|GH^k|g^k\rangle = 0$$

so from (4) and (5)

$$-\alpha^i\langle\sigma^i|G|\sigma^k\rangle = 0.$$

Therefore

$$\langle\sigma^i|G|\sigma^k\rangle = 0 \qquad 0 \leqslant i < k. \tag{13}$$

Also from (8), (11) and (13)

$$\begin{aligned} \langle y^k|H^kG|\sigma^i\rangle &= \langle y^k|\sigma^i\rangle \\ &= \langle\sigma^k|G|\sigma^i\rangle \\ &= 0 \qquad 0 \leqslant i < k. \end{aligned}$$

Therefore using the above result and equations (7), (11) and (13)

$$\begin{aligned} H^{k+1}G|\sigma^i\rangle &= H^kG|\sigma^i\rangle \\ &= |\sigma^i\rangle \quad 0 \leqslant i < k. \end{aligned} \tag{14}$$

Equations (9), (13) and (14) prove the induction. Equation (10) proves that the vectors $|\sigma^0\rangle, |\sigma^1\rangle, \ldots, |\sigma^{n-1}\rangle$ are linearly independent and therefore $H^n = G^{-1}$.

That the minimum is found by n iterations is proved by equation (12). $|g^n\rangle$ must be orthogonal to $|\sigma^0\rangle, |\sigma^1\rangle, \ldots, |\sigma^{n-1}\rangle$ which is only possible if $|g^n\rangle$ is identically zero.

6. Improving the matrix H

The matrix H^i is modified by adding to it two terms A^i and B^i. A^i is the factor which makes H tend to G^{-1} in the sense that for a quadratic

$$G^{-1} = \sum_{i=0}^{n-1} A^i. \tag{15}$$

This result can be proved from the orthogonality conditions (10) because these imply that $S'GS = \Lambda$, where S is the matrix of vectors $|\sigma^i\rangle$, and Λ is a diagonal matrix with elements $\langle\sigma^i|G|\sigma^i\rangle$.

Hence by definition $$\begin{aligned} G &= (S')^{-1}\Lambda(S^{-1}) \\ &= (S\Lambda^{-1}S')^{-1}. \end{aligned}$$

Therefore $$G^{-1} = S\Lambda^{-1}S'$$

and as Λ is a diagonal matrix this reduces to

$$\begin{aligned} G^{-1} &= \sum_i (\Lambda^{-1})_{ii}|\sigma^i\rangle\langle\sigma^i| \\ &= \sum_i \langle\sigma^i|G|\sigma^i\rangle^{-1}|\sigma^i\rangle\langle\sigma^i|. \end{aligned}$$

Therefore from the definition of A^i and equation (8), equation (15) is proved.

The form of the term B^i can be deduced because equation (9) must be valid. For a quadratic we must have

$$\begin{aligned} H^{i+1}G|\sigma^i\rangle &= |\sigma^i\rangle \\ &= H^iG|\sigma^i\rangle + A^iG|\sigma^i\rangle + B^iG|\sigma^i\rangle. \end{aligned}$$

Therefore as $A^iG|\sigma^i\rangle = |\sigma^i\rangle$ the equation

$$B^iG|\sigma^i\rangle = -H^iG|\sigma^i\rangle = -H^i|y^i\rangle$$

must be satisfied.

This implies that the simplest form for B^i is

$$B^i = -\frac{H^i|y^i\rangle\langle z|}{\langle z|y^i\rangle}$$

and as B^i is to be symmetric this gives

$$B^i = -\frac{H^i|y^i\rangle\langle y^i|H^i}{\langle y^i|H^i|y^i\rangle}.$$

Although Davidon's method involves these relations, some of the other ideas used by him can cause H not to tend to G^{-1} even in the quadratic case. The effect in the non-quadratic case would depend upon the function in question but might well lead to slower convergence.

7. Numerical results—comparison with other procedures

As a comparison with other methods we use the function given by Rosenbrock

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

starting at $(-1{\cdot}2,\ 1{\cdot}0)$. This function is difficult to minimize on account of its having a steep sided valley following the curve $x_1^2 = x_2$. Eighteen iterations were required to reach the minimum, each one requiring the minimum to be calculated in only one direction. Table 1 shows how this procedure compares with the classical steepest descent method and Powell's method, one of the procedures with quadratic convergence. The table takes into account that the latter method requires minima to be found in three directions for each iteration. It will be seen that this method is considerably more efficient than that of Powell, both of these being far more efficient than steepest descents.

Downloaded from https://academic.oup.com/comjnl/article-abstract/6/2/163/364776 by guest on 29 September 2019

Table 1

A comparison in two dimensions

EQUIVALENT n	STEEPEST DESCENTS $f(x_1, x_2)$	POWELL'S METHOD $f(x_1, x_2)$	OUR METHOD $f(x_1, x_2)$
0	24·200	24·200	24·200
3	3·704	3·643	3·687
6	3·339	2·898	1·605
9	3·077	2·195	0·745
12	2·869	1·412	0·196
15	2·689	0·831	0·012
18	2·529	0·432	1×10^{-8}
21	2·383	0·182	—
24	2·247	0·052	—
27	2·118	0·004	—
30	1·994	5×10^{-5}	—
33	1·873	8×10^{-9}	—

Table 2

A quadratic function

ITERATION	0	1	2
x	$-4 \quad +2$	$-1{\cdot}69 \quad -1{\cdot}08$	$0 \quad 0$
f	$+40$	$+1{\cdot}54$	10^{-15}
H	$\begin{matrix} +1 & 0 \\ 0 & +1 \end{matrix}$	$\begin{matrix} +0{\cdot}781 & +0{\cdot}361 \\ +0{\cdot}361 & +0{\cdot}411 \end{matrix}$	$\begin{matrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{matrix}$
$\lvert\sigma\rangle$	$+2{\cdot}31 \quad -3{\cdot}08$	$+1{\cdot}69 \quad +1{\cdot}08$	—
A	$\begin{matrix} +0{\cdot}069 & -0{\cdot}092 \\ -0{\cdot}092 & +0{\cdot}123 \end{matrix}$	$\begin{matrix} +0{\cdot}931 & +0{\cdot}592 \\ +0{\cdot}592 & +0{\cdot}377 \end{matrix}$	$\begin{matrix} — \\ — \end{matrix}$

A similar comparison was made with the function given by Powell:

$$f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

starting at (3, −1, 0, 1). In six iterations the method reduced f from 215 to $2{\cdot}5 \times 10^{-8}$. Powell's method took the equivalent of seven iterations to reach 9×10^{-3}, whereas steepest descents only reached 6·36 in seven iterations. The method also brought out the singularity of $\boldsymbol{G}$ at the minimum of f, the elements of $\boldsymbol{H}$ becoming increasingly large.

To compare this variation of Davidon's method with his original method the simple quadratic

$$f(x_1, x_2) = x_1^2 - 2x_1x_2 + 2x_2^2$$

was used. The complete progress of the method described is given in Table 2, showing that it does terminate in two iterations and that $\boldsymbol{H}$ does converge to $\boldsymbol{G}^{-1}$ which for this function is

$$\begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

It will be noticed also, as proved, that $\boldsymbol{G}^{-1} = \sum_i \boldsymbol{A}^i$.

In Davidon's method, although a value of f of similar order of magnitude had been reached in two iterations, $\boldsymbol{H}$ had only reached

$$\begin{pmatrix} 0{\cdot}95 & 0{\cdot}47 \\ 0{\cdot}47 & 0{\cdot}48 \end{pmatrix}.$$

This was due to one of the alternatives allowed by Davidon. Also his procedure for terminating the process was unsatisfactory, and the computation had to be stopped manually.

A non-quadratic test in three dimensions was also made, by using a function with a steep sided helical valley. This function

$$f(x_1, x_2, x_3) = 100\{[x_3 - 10\theta(x_1, x_2)]^2 + [r(x_1, x_2) - 1]^2\} + x_3^2$$

where $\quad 2\pi\theta(x_1, x_2) = \arctan(x_2/x_1), \quad x_1 > 0$

$\qquad\qquad\qquad\qquad = \pi + \arctan(x_2/x_1), \quad x_1 < 0$

and $\quad r(x_1, x_2) = (x_1^2 + x_2^2)^{\frac{1}{2}}$

has a helical valley in the x_3 direction with pitch 10 and radius 1. It is only considered for

$$-\pi/2 < 2\pi\theta < 3\pi/2$$

that is $\quad -2{\cdot}5 < x_3 < 7{\cdot}5.$

It has a minimum at the point (1, 0, 0).

Both methods were started from (−1, 0, 0) and $\boldsymbol{H}^0$ set to the unit matrix. The method given in this paper converged in eighteen iterations, whereas Davidon's method required only ten. However, on account of the more complicated nature of Davidon's iterations, the minimum often being sought along more than one direction in a single iteration, the time taken by the two procedures was almost identical. The progress of this method on the function is given in Table 3.

8. Numerical results—functions of a large number of variables

Tests were also made to find out whether the method is suitable for finding the minimum of a function of a large number of variables. In these tests the Stretch computer was used to solve non-linear simultaneous equations in up to a hundred variables.

The equations were

$$\sum_{j=1}^{n} A_{ij} \sin \alpha_j + B_{ij} \cos \alpha_j = E_i \qquad i = 1, 2, \ldots, n$$

Downloaded from https://academic.oup.com/comjnl/article-abstract/6/2/163/364776 by guest on 29 September 2019

Table 3

A function with a steep-sided helical valley

n	x_1	x_2	x_3	f
0	−1·000	0·000	0·000	$2{\cdot}5 \times 10^4$
1	−1·000	2·278	1·431	$5{\cdot}2 \times 10^3$
2	−0·023	2·004	2·649	$1{\cdot}1 \times 10^3$
3	−0·856	1·559	3·429	74·080
4	−0·372	1·127	3·319	24·190
5	−0·499	0·908	3·285	10·942
6	−0·314	0·900	3·075	9·841
7	0·059	1·069	2·408	6·304
8	0·146	1·086	2·261	6·093
9	0·774	0·725	1·218	1·889
10	0·746	0·706	1·242	1·752
11	0·894	0·496	0·772	0·762
12	0·994	0·298	0·441	0·382
13	0·994	0·191	0·317	0·141
14	1·017	0·085	0·133	0·058
15	0·997	0·070	0·110	0·013
16	1·002	0·009	0·014	8×10^{-4}
17	1·000	0·002	0·040	3×10^{-6}
18	1·000	10^{-5}	10^{-5}	7×10^{-8}

so that the function to be minimized was

$$f = \sum_{i=1}^{n} \{E_i - \sum_{j=1}^{n} (A_{ij} \sin \alpha_j + B_{ij} \cos \alpha_j)\}^2.$$

The matrix elements of A and B were generated as random integers between −100 and +100, and the values of the variables α_i, $i = 1, 2, \ldots, n$ were generated randomly between $-\pi$ and π. For these values the right-hand sides of the equations, E_i, were worked out. The method of this paper was applied to find optimum values of α_i starting from $(\alpha_i + 0{\cdot}1\delta_i)$ where the δ_i's were also generated as random numbers between $-\pi$ and π. In each run the criterion for convergence was that every α should be found to accuracy 0·0001.

The method was entirely successful. Table 4 shows that the number of times f and its derivatives had to be calculated was approximately linear in the number of variables. The total time taken for *all* the runs was fifteen minutes, ten minutes of which was spent on the final case. That a different minimum was found on five occasions was not surprising because if $A \equiv B$ it may be shown that there are up to 2^n real solutions to the equations such that $|\alpha_i| \leqslant \pi$. This abundance of minima emphasizes the power of the method because in every case it converged to a reasonable solution.

The progress of these tests is interesting. For the first n iterations the changes in the function were similar to those experienced with the method of steepest descents, that is a substantial change occurred initially due to descending into a nearby valley, after which convergence was slow. However, after n iterations had been completed

Table 4

Application to a function of many variables

n	NO. OF TIMES f EVALUATED	WHETHER EXPECTED MINIMUM FOUND
5	19	Yes
5	23	Yes
10	36	Yes
10	29	Yes
20	89	No
20	84	Yes
20	68	Yes
20	121	Yes
30	86	No
30	92	Yes
30	118	No
30	113	Yes
50	169	No
50	119	Yes
100	318	No

a good approximation to the final matrix H had been accumulated, after which the function was decreased substantially at each iteration. For example in the hundred-variable trial the function to be minimized was decreased from 293407 to 54165 in the first ten iterations, and to 14686 after one hundred iterations. After 120 iterations it was down to 1342, and after 140 to 147. The function was reduced to 0·44 by 160 iterations, and the minimum was found on the 162nd. The second fifty-variable trial was even more striking. Ten iterations reduced the function from 205380 to 4264, fifty iterations reduced it to 3526, and a further ten iterations reduced it to 0·027.

The conclusion to be drawn from this behaviour is that for many applications of the method a substantial number of the iterations required will be spent on setting up the inverse of the matrix of second derivatives. Therefore, if a good positive definite approximation to H can be calculated initially, as is the case when the method is being applied to solving simultaneous equations, then this approximation should be chosen for H^0.

9. Conclusion

The numerical examples show clearly that the type of method given by Davidon is considerably superior to other methods previously available. The simplifications we have made enable programs to be written more easily, and they do not seem to impair the speed of convergence. It is obviously practicable to apply this method to find a local minimum of a general function of a large number of variables whose first derivatives can be evaluated quickly, even if only poor initial approximations to a solution are known.

Downloaded from https://academic.oup.com/comjnl/article-abstract/6/2/163/364776 by guest on 29 September 2019

10. Acknowledgement

One of us, R.F., is indebted to Dr. C. M. Reeves for his constant help and encouragement, and also to the D.S.I.R. for the provision of a research studentship.

Appendix

The minimum on a line

A simple algorithm is given for estimating the parameter α^i. A point $|y^i\rangle$ is chosen on $|x^i\rangle + \lambda|s^i\rangle$ with $\lambda > 0$. Let f_x, $|g_x\rangle$, f_y and $|g_y\rangle$ denote the values of the function and gradient at the points $|x^i\rangle$ and $|y^i\rangle$. Then an estimate of α^i can be formed by interpolating cubically, using the function values f_x and f_y and the components of the gradients along $|s^i\rangle$.

This is given by

$$\frac{\alpha^i}{\lambda} = 1 - \frac{\langle g_y|s^i\rangle + w - z}{\langle g_y|s^i\rangle - \langle g_x|s^i\rangle + 2w}$$

where $w = (z^2 - \langle g_x|s^i\rangle \langle g_y|s^i\rangle)^{\frac{1}{2}}$

and $z = \frac{3}{\lambda}(f_x - f_y) + \langle g_x|s^i\rangle + \langle g_y|s^i\rangle.$

A suitable choice of the point $|y^i\rangle$ is given by $|y^i\rangle = |x^i\rangle + \eta|s^i\rangle$ where

$$\eta = \text{MINIMUM OF} \left\{1, \frac{-2(f_x - f_0)}{\langle g_x|s^i\rangle}\right\}.$$

f_0 is the predicted lower bound of $f(|x\rangle)$, for example zero in least-squares calculations. This value of η ensures that the choice of $|y^i\rangle$ is reasonable.

It is necessary to check that $f(|x^i\rangle + \alpha^i|s^i\rangle)$ is less than both f_x and f_y. If it is not, the interpolation must be repeated over a smaller range. Davidon suggests one should ensure that the minimum is located between $|x^i\rangle$ and $|y^i\rangle$ by testing the sign of $\langle g_y|s^i\rangle$ and comparing f_x and f_y before interpolating. The reader is referred to Davidon's report for more extensive details of this stage.

References

BOOTH, A. D. (1957). *Numerical Methods*, London: Butterworths.

COURANT, R. (1943). "Variational methods for the solution of problems of equilibrium and vibrations," *Bull. Amer. Math. Soc.*, Vol. 49, p. 1.

CURRY, H. D. (1944). "The method of steepest descent for non-linear minimization problems;" *Qu. App. Maths.*, Vol. 2, p. 258.

DAVIDON, W. C. (1959). "Variable metric method for minimization," A.E.C. Research and Development Report, ANL-5990 (Rev.).

DIRAC, P. A. M. (1958). *The Principles of Quantum Mechanics*, Oxford: O.U.P.

HESTENES, M. R., and STIEFEL, E. (1952). "Methods of conjugate gradients for solving linear systems," *J. Res. N.B.S.*, Vol. 49, p. 409.

HOUSEHOLDER, A. S. (1953). *Principles of Numerical Analysis*, New York: McGraw-Hill.

LEVENBERG, K. (1944). "A method for the solution of certain non-linear problems in least squares," *Qu. App. Maths.*, Vol. 2, p. 164.

MARTIN, D. W., and TEE, G. J. (1961). "Iterative methods for linear equations with symmetric positive definite matrix," *The Computer Journal*, Vol. 4, p. 242.

POWELL, M. J. D. (1962). "An iterative method for finding stationary values of a function of several variables," *The Computer Journal*, Vol. 5, p. 147.

ROSENBROCK, H. H. (1960). "An automatic method for finding the greatest or least value of a function," *The Computer Journal*, Vol. 3, p. 175.

SHAH, B. V., BUEHLER, R. J., and KEMPTHORNE, O. (1961). "The method of parallel tangents (Partan) for finding an optimum," Office of Naval Research Report, NR-042-207 (No. 2).

Book Review (*continued from p.* 143)

here. It seems entirely wrong that modern source languages should be dominated by the sequential regimes of early machine codes, and any move towards conventional mathematical forms is to be welcomed. Again, the approach here seems rather tentative and some major benefits are lost. I think it is preferable to make sequential coding subordinate to definitions rather than the other way round: here lies the key to the very important problem of integrating the translator with a realistic operating system.

Amongst the other problems tackled are the handling of complex variables, recurrence relations, and direct transfer of control to parts of the program not similarly accessible in ALGOL. It may be regarded as a tribute to ALGOL that an attempt has been made to graft such a system onto the same tree. At the same time it must accept a measure of responsibility for the fact that the above ideas were not more fully developed and in use three years ago.

J. K. ILIFFE.

Downloaded from https://academic.oup.com/comjnl/article-abstract/6/2/163/364776 by guest on 29 September 2019

COMBINATORICA 4 (4) (1984) 373—395

A NEW POLYNOMIAL-TIME ALGORITHM FOR LINEAR PROGRAMMING

N. KARMARKAR

Received 20 August 1984
Revised 9 November 1984

We present a new polynomial-time algorithm for linear programming. In the worst case, the algorithm requires $O(n^{3.5}L)$ arithmetic operations on $O(L)$ bit numbers, where n is the number of variables and L is the number of bits in the input. The running-time of this algorithm is better than the ellipsoid algorithm by a factor of $O(n^{2.5})$. We prove that given a polytope P and a strictly interior point $\mathbf{a} \in P$, there is a projective transformation of the space that maps P, $\mathbf{a}$ to P', $\mathbf{a}'$ having the following property. The ratio of the radius of the smallest sphere with center $\mathbf{a}'$, containing P' to the radius of the largest sphere with center $\mathbf{a}'$ contained in P' is $O(n)$. The algorithm consists of repeated application of such projective transformations each followed by optimization over an inscribed sphere to create a sequence of points which converges to the optimal solution in polynomial time.

1. Informal outline

In this Section we give an informal discussion of the main ideas involved in the algorithm. A more rigorous description will be given in the next Section.

1.1. Problem definition

We shall reduce the general linear programming problem over the rationals to the following particular case:

$$\text{minimize} \quad \mathbf{c}^T\mathbf{x}$$

$$\text{subject to} \quad \begin{cases} A\mathbf{x} = \mathbf{0} \\ \sum x_i = 1 \\ \mathbf{x} \geqq \mathbf{0}. \end{cases}$$

Here $\mathbf{x}=(x_1, \ldots, x_n)^T \in \mathbf{R}^n$, $\mathbf{c} \in \mathbf{Z}^n$, $A \in \mathbf{Z}^{m \times n}$.

This is a substantially revised version of the paper presented at the Symposium on Theory of Computing, Washington D. C., April 1984.

AMS subject classification (1980): 90 C 05

Let Ω denote the subspace $\{\mathbf{x}|A\mathbf{x}=\mathbf{0}\}$. Let Δ denote the simplex $\Delta = \{x|x \geqq 0, \sum x_i = 1\}$ and let Π be the polytope defined by

$$\Pi = \Omega \cap \Delta.$$

Thus the problem we consider is

$$\begin{aligned} &\text{minimize} \quad \mathbf{c}^T\mathbf{x} \\ &\text{subject to} \quad \mathbf{x} \in \Pi. \end{aligned}$$

1.2. Optimization over a sphere

If we replace the constraint $\mathbf{x} \in \Delta$ by a constraint of the form $\mathbf{x} \in S$, where S is a sphere, then it is trivial to see that the optimum can be found by solving a linear system of equations.

The intersection of a sphere and an affine space is a lower dimensional sphere inside that affine space. Hence the problem becomes:

$$\min \mathbf{c}'^T\mathbf{x}, \quad \text{subject to} \quad \mathbf{x} \in S' \text{ (a sphere)}$$

where $\mathbf{c}'$ was obtained by projecting $\mathbf{c}$ orthogonally onto Ω.

But this problem is trivial: From the center of the sphere, take a step along the direction $-\mathbf{c}'$, of length equal to the radius of the sphere.

1.3. Bounds on the objective function

Let $\mathbf{a}_0$ be strictly interior point in the polytope P. Suppose we draw an ellipsoid E with center $\mathbf{a}_0$, that is contained in the polytope and solve the optimization problem over the restricted region E instead of P. How good a solution do we get as compared to the optimal solution? To derive a bound, we create another ellipsoid E' by magnifying E by a sufficiently large factor v so that E' contains P.

$$E \subset P \subset E', \quad E' = vE.$$

Let $f_E, f_P, f_{E'}$ denote the minimum values of objective function $f(\mathbf{x})$ on E, P, and E' respectively.

$$f(\mathbf{a}_0) - f_E \leqq f(\mathbf{a}_0) - f_P \leqq f(\mathbf{a}_0) - f_{E'} = v[f(\mathbf{a}_0) - f_E].$$

The last equation follows from the linearity of $f(\mathbf{x})$.

$$\frac{f(\mathbf{a}_0) - f_E}{f(\mathbf{a}_0) - f_P} \geqq \frac{1}{v}$$

$$\frac{f_E - f_P}{f(\mathbf{a}_0) - f_P} \leqq \left(1 - \frac{1}{v}\right).$$

Thus by going from $\mathbf{a}_0$ to the point, say $\mathbf{a}'$, that minimizes $f(\mathbf{x})$ over E we come closer to the minimum value of the objective function by a factor $1 - v^{-1}$. We can repeat the same process with $\mathbf{a}'$ as the center. The rate of convergence of this method depends on v, the smaller the value of v, the faster the convergence.

1.4. Projective transformations

Projective transformations of $\mathbf{R}^n$ are described by formulas of the form

$$\mathbf{x} \mapsto \frac{C\mathbf{x}+\mathbf{d}}{\mathbf{f}^T\mathbf{x}+g}$$

where $C \in \mathbf{R}^{n\times n}$, $\mathbf{d}, \mathbf{f} \in \mathbf{R}^n$, $g \in \mathbf{R}$ and the matrix

$$\begin{bmatrix} C & \mathbf{d} \\ \mathbf{f}^T & g \end{bmatrix}$$

is non-singular.

We shall be interested in projective transformations of the hyperplane

$$\Sigma = \{\mathbf{x} | \sum_i x_i = 1\}.$$

It is easy to see that these transformations are described by

$$\mathbf{x} \mapsto \frac{C\mathbf{x}}{\mathbf{e}^T C\mathbf{x}}$$

where $\mathbf{e}=(1, 1, \ldots, 1)^T$ and C is a non-singular $n\times n$ matrix.

For each interior point $\mathbf{a}$ of Δ there is a unique projective transformation $T(\mathbf{a}, \mathbf{a}_0)$ of the hyperplane Σ fixing all vertices of the simplex Δ and moving $\mathbf{a}=(a_1, \ldots, a_n)^T$ to the center $\mathbf{a}_0=(1/n)\mathbf{e}$. This transformation is given by

$$x_i' = \frac{x_i/a_i}{\sum_j x_j/a_j}.$$

Observe that the corresponding matrix C is now diagonal: $C=\operatorname{diag}\left(\frac{1}{a_1}, \ldots, \frac{1}{a_n}\right)$. The transformation $C(\mathbf{a}, \mathbf{a}_0)$ has the following properties:

1. T is one-to-one and maps the simplex onto itself. Its inverse is given by

$$x_i = \frac{a_i x_i'}{\sum_j a_j x_j'} \quad i = 1, 2, \ldots, n.$$

2. Each facet of the simplex given by $x_i=0$ is mapped onto the corresponding facet $x_i'=0$.

3. Image of the point $\mathbf{x}=\mathbf{a}$ is the center of the simplex given by $x_i'=\frac{1}{n}$.

4. Let A_i denote the ith column of A. Then the system of equations

$$\sum_i A_i x_i = \mathbf{0}$$

now becomes

$$\frac{\sum_i A_i a_i x_i'}{\sum_i a_i x_i'} = \mathbf{0}$$

or

$$\sum_i A_i' x_i' = \mathbf{0}$$

where

$$A_i' = a_i A_i.$$

We denote the affine subspace $\{\mathbf{x}' | A'\mathbf{x}' = \mathbf{0}\}$ by Ω'.

5. If $\mathbf{a} \in \Omega$ then the center of the simplex $\mathbf{a}_0 \in \Omega'$.

Let $B(\mathbf{a}_0, r)$ be the largest ball with center $\mathbf{a}_0$, the same as the center of the simplex, and radius r that is contained in the simplex. Similarly let $B(\mathbf{a}_0, R)$ be the smallest ball containing Δ. It is easy to show that $R/r = n-1$.

$$B(\mathbf{a}_0, r) \subseteqq \Delta \subseteqq B(\mathbf{a}_0, R)$$

$$B(\mathbf{a}_0, r) \cap \Omega' \subseteqq \Delta \cap \Omega' \subseteqq B(\mathbf{a}_0, R) \cap \Omega'.$$

But $\Delta \cap \Omega'$ is the image Π' of the polytope $\Pi = \Delta \cap \Omega$. The intersection of a ball and an affine subspace is a ball of lower dimension in that subspace and has the same radius if the center of the original ball lies in the affine subspace, which it does in our case. Hence we get

$$B'(\mathbf{a}_0, r) \subseteqq \Pi' \subseteqq B'(\mathbf{a}_0, R), \quad \frac{R}{r} = n-1$$

which proves that v = dimension of the space can always be achieved by this method. (Note: Δ is an $n-1$ dimensional simplex.)

1.5. Invariant potential functions

The algorithm creates a sequence of points $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}, \ldots$ having decreasing values of the objective function. In the kth step, the point $\mathbf{x}^{(k)}$ is brought into the center by a projective transformation. Then we optimize the objective function over the intersection of the inscribed ball and the affine subspace to find the next point $\mathbf{x}^{(k+1)}$. Based on the results of the previous Sections, we expect the objective function to be reduced by a factor of $(1-n^{-1})$ at least, in each step, but there is one more technicality we have to deal with. The set of linear functions is not invariant under a projective transformation, but ratios of linear functions are transformed into ratios of linear functions. With every linear objective function $l(\mathbf{x})$ we associate a "potential function" $g(\mathbf{x})$ expressed in terms of logarithms of ratios of linear functions, in order to measure the progress of the algorithm.

$$g(\mathbf{x}) = \sum_j \ln\left(\frac{l(\mathbf{x})}{x_j}\right) + k \quad (k \text{ constant}).$$

It has the following properties:

1. Any desired amount of reduction in the value of $l(\mathbf{x})$ can be achieved by sufficient reduction in the value of $g(\mathbf{x})$.
2. $g(\mathbf{x})$ is mapped into a function of the same form by projective transformations $T(\mathbf{a}, \mathbf{a}_0)$ described in 1.4.
3. Optimization of $g(\mathbf{x})$ in each step can be done approximately by optimizing a linear function (but a different linear function in different steps).

1.6. Complexity of the algorithm

The value of the objective function is reduced by a constant factor in $O(n)$ steps. As in the ellipsoid algorithm we define

$$L = \log(1+|D_{\max}|)+\log(1+\alpha)$$

where $D_{\max} = \max\{|\det(X)| \mid X$ is a square submatrix of constraint matrix $A\}$

$$\alpha = \max\{|c_i'|,\ |b_i| \quad i=1, \dots, n\}.$$

Clearly, L is not greater than the length of the input.

During the course of the algorithm, we periodically check for optimality by converting the interior point solution to an extreme point solution without degrading the objective function value, and testing the extreme point for optimality. The interval between two successive checks can be adjusted so that time spent between checks is comparable to time spent in checking for optimality. Any two distinct values of the objective function on the vertices of the polytope have to differ by at least 2^{-kL} for some constant k [3]. Therefore, in at most $O(nL)$ steps we reach a situation in which any extreme point better than the current interior point has to be an exact optimal solution.

In each step, we have to solve a linear system of equations which takes $O(n^3)$ arithmetic operations in the worst case. However, if we solve the equations at each step by modifying the solution to the previous step, we can save a factor of $\sqrt{n}$. This method is described in Section 6.

For each arithmetic operation, we need a precision of $O(L)$ bits. Note that this much precision is required for any method that solves the equations $A\mathbf{x}=\mathbf{b}$ if $\det(A)=O(2^L)$, e.g., the complexity of gaussian elimination is $O(n^3 L \ln L \ln\ln L)$, although it is normally stated as $O(n^3)$. This factor L must appear in the complexity of any algorithm for linear programming, since representing the output can require $O(L)$ bits/variable in the worst case.

The overall complexity of our algorithm is $O(n^{3.5}L^2 \ln L \ln\ln L)$ as compared to $O(n^6 L^2 \ln L \ln\ln L)$ for the ellipsoid algorithm.

2. Main algorithm

2.1. Problem definition

In order to focus on the ideas which are new and the most important in our algorithm, we make several simplifying assumptions. In Section 5 we will remove all these restrictions by showing how a linear programming problem in standard form can be transformed into our canonical form.

Canonical form of the problem

Input: $A \in \mathbf{Z}^{m\times n}$, $\mathbf{c} \in \mathbf{Z}^n$.

Let $\Omega = \{\mathbf{x} \mid A\mathbf{x}=\mathbf{0}\}$, $\Delta = \{\mathbf{x} \mid \mathbf{x} \geqq \mathbf{0},\ \sum x_i = 1\}$ and $a_0 = e/n$.

Assumptions: $\mathbf{c}^T\mathbf{x} \geqq 0$ for every $\mathbf{x} \in \Omega \cap \Delta$; $\mathbf{a}_0$ is a feasible starting point, i.e. $A\mathbf{a}_0 = 0$; $\mathbf{c}^T\mathbf{a}_0 \neq 0$.

Output: either $\mathbf{x}_0 \in \Omega \cap \Delta$ such that $\mathbf{c}^T\mathbf{x}_0 = 0$ or a proof that $\min\{\mathbf{c}^T\mathbf{x} \mid \mathbf{x} \in \Omega \cap \Delta\} > 0$.

2.2. Comments

1. Note that the feasible region is the intersection of an affine space with a simplex rather than the positive orthant. Initially it takes one projective transformation to bring a linear programming problem to this form.

2. The linear system of equations defining Ω is homogeneous i.e., the right hand side is zero. The projective transformation that transforms positive orthant into simplex also transforms inhomogeneous system of equations into an homogeneous one.

3. The target minimum value of the objective function is zero. This will be justified by combining primal and dual into a single problem.

2.3. Description of the Algorithm

The algorithm creates a sequence of points $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}$ by these steps:

Step 0. Initialize

$$\mathbf{x}^{(0)} = \text{center of the simplex}$$

Step 1. Compute the next point in the sequence

$$\mathbf{x}^{(k+1)} = \varphi(\mathbf{x}^{(k)}).$$

Step 2. Check for infeasibility

Step 3. Check for optimality

go to Step 1.

Now we describe the steps in more detail.

Step 1. The function $\mathbf{b}=\varphi(\mathbf{a})$ is defined by the following sequence of operations.

Let $D=\operatorname{diag}\{a_1, a_2, \ldots, a_n\}$ be the diagonal matrix whose i^{th} diagonal entry is a_i.

1. Let $B=\left[\dfrac{AD}{\mathbf{e}^T}\right]$

i.e., augment the matrix AD with a row of all 1's. This guarantees that $\operatorname{Ker} B \subseteqq \Sigma$.

2. Compute the orthogonal projection of $D\mathbf{c}$ into the null space of B.

$$\mathbf{c}_p = [I - B^T(BB^T)^{-1}B]\, D\mathbf{c}.$$

3. $\hat{\mathbf{c}}=\dfrac{\mathbf{c}_p}{|\mathbf{c}_p|}$, i.e. $\hat{\mathbf{c}}$ is the unit vector in the direction of $\mathbf{c}_p$.

4. $\mathbf{b}'=\mathbf{a}_0-\alpha r\hat{\mathbf{c}}$ i.e. take a step of length αr in the direction $\hat{\mathbf{c}}$, where r is the radius of the largest inscribed sphere

$$r = \frac{1}{\sqrt{n(n-1)}}$$

and $\alpha \in (0, 1)$ is a parameter which can be set equal to 1/4.

5. Apply inverse projective transformation to $\mathbf{b}'$

$$\mathbf{b} = \frac{D\mathbf{b}'}{\mathbf{e}^T D\mathbf{b}'}.$$

Return $\mathbf{b}$.

Step 2. *Check for infeasibility.*
We define a "potential" function by

$$f(\mathbf{x}) = \sum_i \ln \frac{\mathbf{c}^T\mathbf{x}}{x_i}.$$

We expect certain improvement δ in the potential function at each step. The value of δ depends on the choice of parameter α in Step 1.4. For example if $\alpha = 1/4$ then $\delta = 1/8$. If we don't observe the expected improvement i.e. if $f(\mathbf{x}^{(k+1)}) > f(\mathbf{x}^{(k)}) - \delta$ then we stop and conclude that the minimum value of the objective function must be strictly positive. When the canonical form of the problem was obtained by transformation on the standard linear program, then this situation corresponds to the case that the original problem does not have a finite optimum i.e. it is either infeasible or unbounded.

Step 3. *Check for optimality.*
This check is carried out periodically. It involves going from the current interior point to an extreme point without increasing the value of the objective function and testing the extreme point for optimality. This is done only when the time spent since the last check exceeds the time required for checking.

3. Underlying conceptual algorithm

In this Section we show how the computations in step 2 represent an underlying conceptual sequence of operations. We will also introduce more notation and state theorems about the performance of the algorithm. All proofs will be given in Section 4. (A note on notation: In choosing variable names we use the same philosophy as in the design of a large program: Some variables are local to a theorem while others are global, some theorems are stated in terms of "formal" parameters and in order to apply them to the algorithm, one has to substitute actual values for the formal parameters e.g., a result about one step of the algorithm may be stated in terms of input point $\mathbf{a}$ and output point $\mathbf{b}$, and can be applied to the kth step of the algorithm by substituting $\mathbf{a} = \mathbf{x}^{(k)}$ and $\mathbf{b} = \mathbf{x}^{(k+1)}$.)

3.1. The top level

Theorem 1. *In* $O(n(q+\log n))$ *steps the algorithm finds a feasible point* $\mathbf{x}$ *such that*

$$\frac{\mathbf{c}^T\mathbf{x}}{\mathbf{c}^T\mathbf{a}_0} \leqq 2^{-q}.$$

We associate the objective function $\mathbf{c}^T\mathbf{x}$ with a "potential function" $f(\mathbf{x})$

given by

$$f(\mathbf{x}) = \sum_j \ln\left(\frac{\mathbf{c}^T\mathbf{x}}{x_j}\right).$$

Theorem 2. *Either* (i) $f(\mathbf{x}^{(k+1)}) \leqq f(\mathbf{x}^{(k)}) - \delta$ *or*

(ii) *the minimum value of objective function is strictly positive, where* δ *is a constant and depends on the choice of the value of the parameter* α *of Step 1, Substep 4.*

A particular choice that works: If $\alpha = 1/4$, then $\delta \geqq 1/8$.

3.2. The basic step

One step of the algorithm is of the form $\mathbf{b} = \varphi(\mathbf{a})$ which consists of a sequence of three steps:

1. Perform the projective transformation $T(\mathbf{a}, \mathbf{a}_0)$ of the simplex Δ that maps input point $\mathbf{a}$ to the center $\mathbf{a}_0$.

2. Optimize (approximately) the transformed objective function over an inscribed sphere to find a point $\mathbf{b}'$.

3. Apply the inverse of T to point $\mathbf{b}'$ to obtain output $\mathbf{b}$.

We describe each of the steps in detail.

3.3. Projective transformation $T(\mathbf{a}, \mathbf{a}_0)$

Let $\mathbf{a} = (a_1, a_2, \ldots, a_n)$

Let $D = \operatorname{diag}\{a_1, a_2, \ldots, a_n\}$ be the diagonal matrix with diagonal entries $a_1, a_2, \ldots, a_n$. Then $T(\mathbf{a}, \mathbf{a}_0)$ is given by

$$\mathbf{x}' = \frac{D^{-1}\mathbf{x}}{\mathbf{e}^T D^{-1}\mathbf{x}} \quad \text{where} \quad \mathbf{e} = (1, 1, \ldots, 1).$$

Its inverse is given by

$$\mathbf{x} = \frac{D\mathbf{x}'}{\mathbf{e}^T D\mathbf{x}}.$$

Let f' be the transformed potential function defined by

$$f(\mathbf{x}) = f'(T(\mathbf{x})).$$

Then

$$f'(\mathbf{y}) = \sum_j \ln \frac{\mathbf{c}'^T\mathbf{y}}{y_j} - \sum_j \ln a_j$$

where $\mathbf{c}' = D\mathbf{c}$. Let Ω' be the transformed affine space Ω.

$$A\mathbf{x} = 0 \Leftrightarrow AD\,\mathbf{x}' = 0.$$

Thus Ω' is the null-space of AD.

3.4. Optimization over a ball

Let r be the radius of the largest ball with center $\mathbf{a}_0$ that can be inscribed in the simplex Δ. Then

$$r = \frac{1}{\sqrt{n(n-1)}}.$$

We optimize over a smaller ball $B(\mathbf{a}_0, \alpha r)$ $0<\alpha<1$ for two reasons:

1. It allows optimization of $f'(\mathbf{y})$ to be approximated very closely by optimization of a linear function.

2. If we wish to perform arithmetic operations approximately rather than by exact rational arithmetic, it gives us a margin to absorb round-off errors without going outside the simplex.

One choice of the value of α that works is $\alpha=1/4$ and corresponds to $\delta>1/8$ in Theorem 2. We restrict the affine space $\Omega'=\{\mathbf{y}\mid AD\mathbf{y}=\mathbf{0}\}$ by the equation $\sum y_i=1$. Let Ω'' be the resulting affine space. Let B be the matrix obtained by adding a row of all 1's to AD. Then any displacement in Ω'' is in the null space of B,

$$\text{i.e., for } u\in\Omega'' \quad \Omega''=u+\operatorname{Ker} B.$$

We are interested in optimizing $f'(\mathbf{y})$ over $B(\mathbf{a}_0, \alpha r)\cap\Omega''$. First, we prove the existence of a point that achieves a constant reduction in the potential function.

Theorem 3. *There exists a point* $\mathbf{b}'\in B(\mathbf{a}_0, \alpha r)\cap\Omega''$ *such that* $f'(\mathbf{b}')\leqq f'(\mathbf{a}_0)-\delta$, *where* δ *is a constant.*

Then we prove that minimization of $f'(\mathbf{x})$ can be approximated by minimization of the linear function $\mathbf{c}'^T\mathbf{x}$.

Theorem 4. *Let* $\mathbf{b}'$ *be the point that minimizes* $\mathbf{c}'^T\mathbf{x}$ *over* $B(\mathbf{a}_0, \alpha r)\cap\Omega''$. *Then* $f'(\mathbf{b}')\leqq f'(\mathbf{a}_0)-\delta$ *where* δ *is a positive constant depending on* α. *For* $\alpha=1/4$ *we may take* $\delta=1/8$.

Finally we describe an algorithm to minimize $\mathbf{c}'^T\mathbf{x}$, over $B(\mathbf{a}_0, \alpha^* r)$.

Algorithm A.

1. Project $\mathbf{c}'$ *orthogonally into the null space of* B:

$$\mathbf{c}_P = [I-B^T(BB^T)^{-1}B]\mathbf{c}'.$$

2. Normalize $\mathbf{c}_P$:

$$\hat{\mathbf{c}}_P = \frac{\mathbf{c}_P}{|\mathbf{c}_P|}.$$

3. Take a step of length αr *along the direction* $-\hat{\mathbf{c}}_P$

$$\mathbf{b}' = \mathbf{a}_0-\alpha r\hat{\mathbf{c}}_P.$$

Theorem 5. *The point* $\mathbf{b}'$ *returned by algorithm A minimizes* $\mathbf{c}'^T\mathbf{x}$ *over* $B(\mathbf{a}_0, \alpha r)\cap\Omega''$.

4. Analysis of the algorithm

Proof of Theorem 5. Let $\mathbf{z}$ be any point such that $\mathbf{z}\in B(\mathbf{a}_0, \alpha r)\cap \Omega''$

$$\mathbf{b}', \mathbf{z}\in\Omega'' \Rightarrow B(\mathbf{b}'-\mathbf{z}) = 0$$

$$\Rightarrow B^T(BB^T)^{-1}B(\mathbf{b}'-\mathbf{z}) = 0$$

$$\Rightarrow (\mathbf{c}'-\mathbf{c}_P)^T(\mathbf{b}'-\mathbf{z}) = 0$$

$$\Rightarrow \mathbf{c}'^T(\mathbf{b}'-\mathbf{z}) = \mathbf{c}_P^T(\mathbf{b}'-\mathbf{z})$$

$$\mathbf{c}_P^T(\mathbf{b}'-\mathbf{z}) = |\mathbf{c}_P^T|\hat{\mathbf{c}}_P^T[\mathbf{a}_0-\alpha r\hat{\mathbf{c}}_P-\mathbf{z}]$$

$$= |\mathbf{c}_P^T|\{\hat{\mathbf{c}}_P^T(\mathbf{a}_0-\mathbf{z})-\alpha r\}$$

$$\hat{\mathbf{c}}_P^T(\mathbf{a}_0-\mathbf{z}) \leqq |\hat{\mathbf{c}}_P||\mathbf{a}_0-\mathbf{z}| \leqq \alpha r \quad \text{since} \quad \mathbf{z}\in B(\mathbf{a}_0, \alpha r)$$

$$\mathbf{c}_P^T(\mathbf{b}'-\mathbf{z}) \leqq 0$$

$$\mathbf{c}'^T(\mathbf{b}'-\mathbf{z}) \leqq 0$$

$\mathbf{c}'^T\mathbf{b}' \leqq \mathbf{c}'^T\mathbf{z}$, for all $\mathbf{z}\in B(\mathbf{a}_0, \alpha r)\cap\Omega''$. ∎

Proof of Theorem 3. Let $\mathbf{x}^*$ be the point in $\Omega''\cap S$ where the minimum of $\mathbf{c}'^T\mathbf{x}$ is achieved. Clearly $\mathbf{x}^*\notin B(\mathbf{a}_0, \alpha r)$.

Let $\mathbf{b}'$ be the point where the line segment $\mathbf{a}_0\mathbf{x}^*$ intersects the boundary of the sphere $B(\mathbf{a}_0, \alpha r)$. $\mathbf{b}'=(1-\lambda)\mathbf{a}_0+\lambda\mathbf{x}^*$ for some $\lambda\in[0, 1]$. By definition, $\mathbf{b}'\in B(\mathbf{a}_0, \alpha r)$ and $\mathbf{a}_0, \mathbf{x}^*\in\Omega''$ imply $\mathbf{b}'\in\Omega''$.

Hence $\mathbf{b}'\in B(\mathbf{a}_0, \alpha r)\cap\Omega''$

$$\mathbf{c}'^T\mathbf{b}' = (1-\lambda)\mathbf{c}'^T\mathbf{a}_0+\lambda\mathbf{c}'^Tx^* = (1-\lambda)\mathbf{c}'^T\mathbf{a}_0$$

$$\frac{\mathbf{c}'^T\mathbf{a}_0}{\mathbf{c}'^T\mathbf{b}'} = \frac{1}{1-\lambda}$$

and

$$b_j' = (1-\lambda)a_{0j}+\lambda x_j^*.$$

Hence

$$f'(\mathbf{a}_0)-f'(\mathbf{b}') = \sum_j \ln\frac{\mathbf{c}'^T\mathbf{a}_0}{\mathbf{c}'^T\mathbf{b}'}\frac{b_j}{a_{0j}} = \sum_j \ln\frac{(1-\lambda)a_{0j}+\lambda x_j^*}{(1-\lambda)a_{0j}}$$

$$= \sum_j \ln\left[1+\frac{\lambda}{1-\lambda}\frac{x_j^*}{a_{0j}}\right].$$

Now we use the following inequality: If $P_i \geqq 0$ then $\prod_i (1+P_i) \geqq 1+\sum P_i$. Therefore, $\sum_i \ln(1+P_i) \geqq \ln(1+\sum P_i)$.

$$f'(\mathbf{a}_0)-f'(\mathbf{b}') \geqq \ln\left[1+\frac{\lambda}{1-\lambda}\frac{1}{a_{0j}}\sum_j x_j^*\right] = \ln\left[1+\frac{n\lambda}{1-\lambda}\right]$$

$$\mathbf{b}' = (1-\lambda)\mathbf{a}_0+\lambda\mathbf{x}^*$$

$$\mathbf{b}'-\mathbf{a}_0 = \lambda(\mathbf{x}^*-\mathbf{a}_0)$$

$$\alpha r = |\mathbf{b}'-\mathbf{a}_0| = \lambda|\mathbf{x}^*-\mathbf{a}_0| \leqq \lambda R$$

where R is the radius of the smallest circumscribing sphere and $R/r=n-1$.

$$\lambda \geqq \alpha\frac{r}{R} = \frac{\alpha}{n-1}$$

$$1+\frac{\lambda n}{1-\lambda} \geqq 1+\frac{n\alpha/(n-1)}{1-\alpha/(n-1)} = 1+\frac{n\alpha}{n-1-\alpha} \geqq 1+\alpha$$

$$f'(\mathbf{a}_0)-f'(\mathbf{b}') \geqq \ln(1+\alpha).$$

Taking $\delta=\ln(1+\alpha)$, $f'(\mathbf{b}') \leqq f'(\mathbf{a}_0)-\delta$. ∎

Proof of Theorem 4. First we prove a few lemmas.

Lemma 4.1. *If* $|x| \leqq \beta < 1$ *then* $|\ln(1+x)-x| \leqq \dfrac{x^2}{2(1-\beta)}$.

Proof. A simple exercise in calculus. ∎

Lemma 4.2. *Let* $\beta=\alpha\sqrt{\dfrac{n}{n-1}}$. *Then for any* $\mathbf{x} \in B(\mathbf{a}_0, \alpha r)$, $\left|\sum_j \ln\dfrac{x_j}{a_{0j}}\right| \leqq \dfrac{\beta^2}{2(1-\beta)}$.

Proof. $\mathbf{x} \in B(\mathbf{a}_0, \alpha r)$ implies $|\mathbf{x}-\mathbf{a}_0|^2 \leqq \alpha^2 r^2$. Therefore

$$\sum_j \left(\frac{x_j-a_{0j}}{a_{0j}}\right)^2 \leqq \frac{\alpha^2 r^2}{(1/n)^2} = \frac{\alpha^2 n^2}{n(n-1)} = \beta^2$$

$$\left|\frac{x_j-a_{0j}}{a_{0j}}\right| \leqq \beta \quad \text{for} \quad j = 1, \ldots, n$$

$$\left|\ln\left(1+\frac{x_j-a_{0j}}{a_{0j}}\right)-\frac{x_j-a_{0j}}{a_{0j}}\right| \leqq \frac{1}{2(1-\beta)}\left(\frac{x_j-a_{0j}}{a_{0j}}\right)^2 \quad \text{by Lemma 4.1}$$

$$\sum_j \ln\frac{x_j}{a_{0j}} = \sum_j \ln\left[1+\frac{x_j-a_{0j}}{a_{0j}}\right]$$

$$\sum_j \left|\ln\frac{x_j}{a_{0j}}-\frac{x_j-a_{0j}}{a_{0j}}\right| \leqq \frac{1}{2(1-\beta)}\sum_j\left(\frac{x_j-a_{0j}}{a_{0j}}\right)^2 = \frac{\beta^2}{2(1-\beta)}$$

but

$$\sum_j \frac{x_j-a_{0j}}{a_{0j}} = \frac{1}{a_{0j}}\left[\sum_j x_j-\sum_j a_{0j}\right] = \frac{1}{a_{0j}}[1-1] = 0$$

$$\left|\sum_j \ln\frac{x_j}{a_{0j}}\right| \leqq \frac{\beta^2}{2(1-\beta)}. \quad ∎$$

Now we prove Theorem 4.

Define $\tilde{f}(\mathbf{x})=n\ln\dfrac{\mathbf{c}'^T\mathbf{x}}{\mathbf{c}'^T\mathbf{a}_0}$. Let $\mathbf{b}_m$ be the point in $B(\mathbf{a}_0,\alpha r)\cap\Omega''$ where $f(\mathbf{x})$ achieves its minimum value.

$$\begin{aligned} f(\mathbf{a}_0)-f(\mathbf{b}') &= f(\mathbf{a}_0)-f(\mathbf{b}_m)+f(\mathbf{b}_m)-f(\mathbf{b}') \\ &= [f(\mathbf{a}_0)-f(\mathbf{b}_m)] \\ &\quad +[f(\mathbf{b}_m)-(f(\mathbf{a}_0)+\tilde{f}(\mathbf{b}_m))] \\ &\quad -[f(\mathbf{b}')-(f(\mathbf{a}_0)+\tilde{f}(\mathbf{b}'))] \\ &\quad +[\tilde{f}(\mathbf{b}_m)-\tilde{f}(\mathbf{b}')] \end{aligned}$$

By Theorem 3,

$$f(\mathbf{a}_0)-f(\mathbf{b}_m)\geqq \ln(1+\alpha). \tag{1}$$

For $\mathbf{x}\in B(\mathbf{a}_0,\alpha r)\cap\Omega'$,

$$\begin{aligned} f(\mathbf{x})-(f(\mathbf{a}_0)+\tilde{f}(\mathbf{x})) &= \sum_j \ln\frac{\mathbf{c}'^T\mathbf{x}}{x_j}-\sum_j\ln\frac{\mathbf{c}'^T\mathbf{a}_0}{a_{0j}}-n\sum\ln\frac{\mathbf{c}'^T\mathbf{x}}{\mathbf{c}'^T a_0} \\ &= -\sum_j \ln\frac{x_j}{a_{0j}} \end{aligned}$$

$$\left|f(\mathbf{x})-(f(\mathbf{a}_0)+\tilde{f}(\mathbf{x}))\right| = \left|\sum_j \ln\frac{x_j}{a_{0j}}\right| \leqq \frac{\beta^2}{2(1-\beta)} \tag{2}$$

by Lemma 4.2.

Since $\tilde{f}(\mathbf{x})$ depends on $\mathbf{c}'^T\mathbf{x}$ in a monotonically increasing manner, $\tilde{f}(\mathbf{x})$ and $\mathbf{c}'^T\mathbf{x}$ achieve their minimum values over $B(\mathbf{a}_0,\alpha r)\cap\Omega''$ at the same point, viz $\mathbf{b}'$,

$$\tilde{f}(\mathbf{b}_m)\geqq\tilde{f}(\mathbf{b}'). \tag{3}$$

By (1), (2) and (3),

$$f(\mathbf{a}_0)-f(\mathbf{b}')\geqq\ln(1+\alpha)-\frac{\beta^2}{(1-\beta)}\geqq\alpha-\frac{\alpha^2}{2}-\frac{\alpha^2 n^2}{n(n-1)\left[1-\alpha\sqrt{\dfrac{n}{n-1}}\right]}.$$

Define

$$\delta=\alpha-\frac{\alpha^2}{2}-\frac{\alpha^2 n}{(n-1)\left[1-\alpha\sqrt{\dfrac{n}{n-1}}\right]}$$

$$f(\mathbf{a}_0)-f(\mathbf{b}')\geqq\delta.$$

Observe that

1. $\lim\limits_{n\to\infty}\delta(n)=\alpha-\dfrac{\alpha^2}{2}-\dfrac{\alpha^2}{1-\alpha}$. If $\alpha=\dfrac{1}{4}$, then $\lim\limits_{n\to\infty}\delta(n)\geqq\dfrac{1}{8}$.

2. If $n\geqq 4$, $\alpha=\dfrac{1}{5}$ then $\delta\geqq\dfrac{1}{10}$. ∎

Proof of Theorem 2. By Theorem 4, $f'(\mathbf{b}')\leqq f'(\mathbf{a}_0)-\delta$. Applying the inverse transform $T^{-1}(\mathbf{a},\mathbf{a}_0)$ and substituting $\mathbf{x}^{(k)}=T^{-1}(\mathbf{a}_0)$, we obtain $\mathbf{x}^{(k+1)}=T^{-1}(\mathbf{b}')$, $f(\mathbf{x}^{(k+1)})\leqq f(\mathbf{x}^{(k)})-\delta$. ∎

Proof of Theorem 1. For each k we have

$$f(\mathbf{x}^{(k+1)}) \leqq f(\mathbf{x}^{(k)}) - \delta$$

$$f(\mathbf{x}^{(k)}) \leqq f(\mathbf{x}^{(0)}) - k\delta$$

$$\sum_j \ln \frac{\mathbf{c}^T \mathbf{x}^{(k)}}{x_j^{(k)}} \leqq \sum_j \ln \frac{\mathbf{c}^T \mathbf{a}_0}{a_{0j}} - k\delta$$

$$n \ln \frac{\mathbf{c}^T \mathbf{x}^{(k)}}{\mathbf{c}^T \mathbf{a}_0} \leqq \sum_j \ln (\mathbf{x}_j^{(k)}) - \sum_j \ln a_{0j} - k\delta.$$

But $x_j^{(k)} \leqq 1$, $a_{0j} = \frac{1}{n}$, so $\ln \frac{\mathbf{c}^T \mathbf{x}^{(k)}}{\mathbf{c}^T \mathbf{a}_0} \leqq \ln (n) - \frac{k}{n} \delta$. Therefore there exists a constant k' such that after $m = k'(n(q + \ln (n)))$ steps.

$$\frac{\mathbf{c}^T \mathbf{x}}{\mathbf{c}^T \mathbf{a}_0} \leqq 2^{-q}. \quad \blacksquare$$

5. Transformation of a linear program to canonical form

Consider a linear programming problem in the standard form:

$$\text{minimize} \quad \mathbf{c}^T \mathbf{x}$$

$$\text{subject to} \begin{cases} A\mathbf{x} \geqq \mathbf{b} \\ \mathbf{x} \geqq \mathbf{0}. \end{cases}$$

There are many ways by which this problem can be converted to our canonical form. We describe one such method.

Step 1. *Combine primal and dual problems.*

$$A\mathbf{x} \geqq \mathbf{b}$$

$$A^T \mathbf{u} \leqq \mathbf{c}$$

$$\mathbf{c}^T \mathbf{x} - \mathbf{b}^T \mathbf{u} = 0$$

$$\mathbf{x} \geqq \mathbf{0} \quad \mathbf{u} \geqq \mathbf{0}.$$

By duality theory, this combined problem is feasible if and only the original problem has finite optimum solution.

Step 2. *Introduce slack variables.*

$$A\mathbf{x} - \mathbf{y} = \mathbf{b}$$

$$A^T \mathbf{u} + \mathbf{v} = \mathbf{c}$$

$$\mathbf{c}^T \mathbf{x} - \mathbf{b}^T \mathbf{u} = 0$$

$$\mathbf{x} \geqq \mathbf{0} \quad \mathbf{u} \geqq \mathbf{0} \quad \mathbf{y} \geqq \mathbf{0} \quad \mathbf{v} \geqq \mathbf{0}.$$

Step 3. *Introduce an artificial variable to create an interior starting point.* Let $\mathbf{x}_0$, $\mathbf{y}_0$, $\mathbf{u}_0$, $\mathbf{v}_0$ be strictly interior points in the positive orthant

$$\text{minimize} \quad \lambda$$

$$\text{subject to} \begin{cases} A\mathbf{x}-\mathbf{y}+(\mathbf{b}-A\mathbf{x}_0+\mathbf{y}_0)\lambda = \mathbf{b} \\ A^T\mathbf{u}+\mathbf{v}+(\mathbf{c}-A^T\mathbf{u}_0)\lambda = \mathbf{c} \\ \mathbf{c}^T\mathbf{x}-\mathbf{b}^T\mathbf{u}+(-\mathbf{c}^T\mathbf{x}_0+\mathbf{b}^T\mathbf{v}_0)\lambda = 0 \\ \mathbf{x} \geqq \mathbf{0} \quad \mathbf{y} \geqq \mathbf{0}, \ \mathbf{u} \geqq \mathbf{0}, \ \mathbf{v} \geqq \mathbf{0}, \ \lambda \geqq 0. \end{cases}$$

Note that $\mathbf{x}=\mathbf{x}_0$, $\mathbf{y}=\mathbf{y}_0$, $\mathbf{u}=\mathbf{u}_0$, $\mathbf{v}=\mathbf{v}_0$, $\lambda=1$ is a strictly interior feasible solution which we can take as a starting point. The minimum value of λ is zero if and only if the problem in step 2 is feasible.

Step 3. *Change of Notation.* For convenience of description we rewrite the problem in Step 3 as

$$\text{minimize} \ \mathbf{c}^T\mathbf{x}, \quad \mathbf{c}, \mathbf{x} \in \mathbf{R}^n$$

$$\text{subject to} \ A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geqq \mathbf{0}$$

$\mathbf{x}=\mathbf{a}$ is a known strictly interior starting point, and the target value of the objective function we are interested in is zero.

Step 4. *A projective transformation of the positive orthant into a simplex.* Consider the transformation $\mathbf{x}'=T(\mathbf{x})$ where $\mathbf{x}\in\mathbf{R}^n$, $\mathbf{x}'\in\mathbf{R}^{n+1}$ are defined by

$$x_i' = \frac{x_i/a_i}{\sum_j (x_j/a_j)+1} \quad i = 1, 2, \ldots, n$$

$$x_{n+1}' = 1 - \sum_{i=1}^{n} x_i'.$$

Let P_+ denote the positive orthant

$$P_+ = \{\mathbf{x}\in\mathbf{R}^n | \mathbf{x} \geqq 0\}$$

and Δ denote the simplex

$$\Delta = \left\{\mathbf{x}\in\mathbf{R}^{n+1} | \mathbf{x} \geqq 0, \quad \sum_{i=1}^{n+1} x_i = 1\right\}.$$

The transformation $T(\mathbf{x})$ has the following properties:

1. It maps P_+ onto Δ.

2. It is one-to-one and the inverse transformation is given by

$$x_i = \frac{a_i x_i'}{x_{n+1}'} \quad i = 1, 2, \ldots, n.$$

3. The image of point $\mathbf{a}$ is the center of the simplex.

4. Let A_i denote ith column of A. If

$$\sum_{i=1}^{n} A_i x_i = \mathbf{b}$$

then

$$\frac{\sum A_i a_i x_i'}{x_{n+1}'} = \mathbf{b}$$

or

$$\sum_{i=1}^{n} A_i a_i x_i' - \mathbf{b} x_{n+1}' = \mathbf{0}.$$

We define the columns of a matrix A' by these equations:

$$A_i' = a_i A_i, \quad i = 1, \ldots, n$$

$$A_{n+1}' = -\mathbf{b}.$$

Then

$$\sum_{i=1}^{n+1} A_i' x_i' = \mathbf{0}.$$

Let Ω' denote this affine subspace, i.e. $\Omega' = \{\mathbf{x}' \in \mathbf{R}^{n+1} | A'\mathbf{x}' = \mathbf{0}\}$. Thus we get a system of homogeneous equations in the transformed space.

5. Since we are interested in target value of zero, we define Z to be the affine subspace corresponding to the zero set of the objective function in the original space

$$Z = \{\mathbf{x} \in \mathbf{R}^n | \mathbf{c}^T \mathbf{x} = 0\}.$$

Substituting

$$x_i = \frac{a_i x_i'}{x_{n+1}'}$$

we get

$$\sum_{i=1}^{n+1} \frac{c_i a_i x_i'}{x_{n+1}'} = 0.$$

Define $\mathbf{c}' \in \mathbf{R}^{n+1}$ by

$$c_i' = a_i c_i \quad i = 1, 2, \ldots, n$$

$$c_{n+1}' = 0.$$

Then $\mathbf{c}^T \mathbf{x} = 0$ implies $\mathbf{c}'^T \mathbf{x}' = 0$.

Therefore the image of Z is given by $Z' = \{\mathbf{x}' \in \mathbf{R}^{n+1} | \mathbf{c}'^T \mathbf{x}' = 0\}$. The transformed problem is

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'^T \mathbf{x}' \\ & A'\mathbf{x}' = \mathbf{0} \\ \text{subject to} \quad & \mathbf{x}' \geqq \mathbf{0}, \ \sum_{i=1}^{n+1} x_i' = 1. \end{aligned}$$

The center of the simplex is a feasible starting point. Thus we get a problem in the canonical form.

Application of our main algorithm to this problem either gives a solution $\mathbf{x}'$ such that $\mathbf{c}'^T \mathbf{x}' = 0$ or we conclude that minimum value of $\mathbf{c}'^T \mathbf{x}'$ is strictly positive. In the first case, the inverse image of $\mathbf{x}'$ gives an optimal solution to the original problem. In the latter case we conclude that the original problem does not have a finite optimum, i.e. it is either infeasible or unbounded.

6. Incremental computation of inverse

6.1. A modification to the main algorithm

The computational effort in each step of the main algorithm is dominated by the computation of $\mathbf{c}_p$:

$$\mathbf{c}_p = [I - B^T(BB^T)^{-1}B]D\mathbf{c}.$$

We have to find the inverse of (BB^T) or solve a linear system of equations of the form $(BB^T)\mathbf{u}=\mathbf{v}$.

Substituting $B=\begin{bmatrix} AD \\ \mathbf{e}^T \end{bmatrix}$

$$BB^T = \begin{bmatrix} AD^2A^T & AD\mathbf{e} \\ (AD\mathbf{e})^T & \mathbf{e}^T\mathbf{e} \end{bmatrix} = \begin{bmatrix} AD^2A^T & \mathbf{0} \\ \mathbf{0} & n \end{bmatrix}. \tag{1}$$

The only quantity that changes from step to step is the diagonal matrix D, since $D_{ii}^{(k)} = x_i^{(k)}$. In order to take advantage of the computations done in previous steps, we exploit the following facts about matrix inserve:

Let D, D' be $n \times n$ diagonal matrices.

1. If D and D' are "close" in some suitable norm, the inverse of AD'^2A^T can be used in place of the inverse of AD^2A^T.

2. If D and D' differ in only one entry, the inverse of AD'^2A^T can be computed in $O(n^2)$ arithmetic operations, given the inverse of AD^2A^T.

(Note: Instead of finding the inverse, one could also devise an algorithm based on other techniques such as LU decomposition. In practice, it may be better to update the LU decomposition directly, but this will not change the theoretical worst-case complexity of the algorithm.)

We define a diagonal matrix $D'^{(k)}$, a "working approximation" to $D^{(k)}$ in step k, such that

$$\frac{1}{2} \leqq \left[\frac{D_{ii}'^{(k)}}{D_{ii}^{(k)}}\right]^2 \leqq 2 \quad \text{for} \quad i = 1, 2, \ldots, n. \tag{2}$$

We use $(A(D'^{(k)})^2A^T)^{-1}$ in place of $(A(D^{(k)})^2A^T)^{-1}$. Initially, $D'^{(0)} = D^{(0)} = \frac{1}{n} I$.

We update D' by the following strategy:

1. $D'^{(k+1)} = \sigma^{(k)} D'^{(k)}$, where $\sigma^{(k)}$ is a scaling factor whose significance will be explained in Section 6.3. For the purpose of this section take $\sigma^{(k)} = 1$.

2. For $i=1$ to n do

if

$$\left[\frac{D_{ii}'^{(k+1)}}{D_{ii}^{(k+1)}}\right]^2 \notin \left[\frac{1}{2}, 2\right],$$

then let $D_{ii}'^{(k+1)} = D_{ii}^{(k+1)}$; and update $(A(D'^{(k+1)})^2A^T)^{-1}$ by rank-one modification.

Thus, this strategy maintains the property in equation (2). Now we describe the rank-one modification procedure. Suppose a symmetric square matrix M is

modified by a rank-one matrix expressed as the outer product of two vectors $\mathbf{u}$ and $\mathbf{v}$. Then its inverse can be modified by the equation

$$(M+\mathbf{u}\mathbf{v}^T)^{-1} = M^{-1} - \frac{(M^{-1}\mathbf{u})(M^{-1}\mathbf{v})^T}{1+\mathbf{u}^T M^{-1}\mathbf{v}}. \tag{3}$$

Given M^{-1}, computation of $(M+\mathbf{u}\mathbf{v}^T)^{-1}$ can be done in $O(n^2)$ steps. In order to apply equation (3), note that if D' and D'' differ only in the ith entry then

$$AD'^2A^T = AD''^2A^T + [D_{ii}'^2 - D_{ii}''^2]\mathbf{a}_i\mathbf{a}_i^T \tag{4}$$

where $\mathbf{a}_i$ is the ith column of A. This is clearly a rank-one modification. If D' and D'' differ in l entries we can perform l successive rank-one updates in $O(n^2 l)$ time to obtain the new inverse. All other operations required for computing $\mathbf{c}_p$, the orthogonal projection of $\mathbf{c}$, involve either multiplication of a vector by a matrix, or linear combination of two vectors, hence can be performed in $O(n^2)$ time.

6.2. Equivalence of the modified and original problems

In this Section we analyze the effect of the replacement of $D^{(k)}$ by $D'^{(k)}$. Consider the following generalization of the optimization problem over an inscribed sphere in the transformed space:

Problem P′:

$$\left.\begin{aligned} &\text{Minimize } \mathbf{c}'^T\mathbf{x} \\ &\text{subject to } B\mathbf{x} = \mathbf{0} \\ &\qquad\text{and } \mathbf{x}^T Q\mathbf{x} \leqq \alpha' r \end{aligned}\right\}. \tag{5}$$

(Taking $Q=I$ and $\alpha'=\alpha$ corresponds to the original problem.)

We are going to show that replacing $D^{(k)}$ by $D'^{(k)}$ corresponds to solving the problem P', for appropriate choice of Q. Let $h(\mathbf{x})=\mathbf{x}^T Q\mathbf{x}$. Its gradient is given by $\nabla h(\mathbf{x})=2Q\mathbf{x}$.

At the optimal solution $\mathbf{x}$ to the problem P',

$$\mathbf{c}' = B^T\boldsymbol{\lambda} + \mu \cdot 2Q\mathbf{x} \tag{6}$$

where the vector $\boldsymbol{\lambda}$ and the scalar μ are undetermined Lagrange multipliers

$$BQ^{-1}\mathbf{c}' = BQ^{-1}B^T\boldsymbol{\lambda} + 2\mu B\mathbf{x} = BQ^{-1}B^T\boldsymbol{\lambda}$$

$$\boldsymbol{\lambda} = (BQ^{-1}B^T)^{-1}BQ^{-1}\mathbf{c}'. \tag{7}$$

Substituting for $\boldsymbol{\lambda}$ in equation (6), $\mathbf{x}$, which we now denote by $\mathbf{c}_p$, is given (up to a scale factor) by

$$\mathbf{x} = \mathbf{c}_p = [I - Q^{-1}B^T(BQ^{-1}B^T)^{-1}B]Q^{-1}\mathbf{c}'. \tag{8}$$

This requires the inverse of $BQ^{-1}B^T$. Substituting $B=\begin{bmatrix} AD \\ \hline \mathbf{e}^T \end{bmatrix}$ we get

$$BQ^{-1}B^T = \begin{bmatrix} ADQ^{-1}DA^T & ADQ^{-1}\mathbf{e} \\ (ADQ^{-1}\mathbf{e})^T & \mathbf{e}^T Q^{-1}\mathbf{e} \end{bmatrix}. \tag{9}$$

It is enough to know the inverse of the submatrix $ADQ^{-1}DA^T$, because then $(BQ^{-1}B^T)^{-1}$ can be computed in $O(n^2)$ additional steps by the formula

$$(10)\quad \begin{bmatrix} M & \mathbf{a} \\ \mathbf{a}^T & c \end{bmatrix}^{-1} = \frac{1}{c-\mathbf{a}^T M^{-1}\mathbf{a}} \begin{bmatrix} (c-\mathbf{a}^T M^{-1}\mathbf{a})M^{-1}+(M^{-1}\mathbf{a})(M^{-1}\mathbf{a})^T & -M^{-1}\mathbf{a} \\ -(M^{-1}\mathbf{a})^T & 1 \end{bmatrix}$$

where M, $\mathbf{a}$ and c are $m\times m$, $m\times 1$, 1×1 matrices, respectively. But we know $(AD'^2A^T)^{-1}$ where $D'=D\cdot E$, and E is a diagonal "error matrix" such that $E_{ii}^2\in \in[1/2, 2]$. Setting $Q=E^{-2}$ we get

$$AD'^2A^T = ADE^2DA^T = ADQ^{-1}DA^T.$$

In the modified algorithm we keep $(AD'^2A^T)^{-1}$ around, update it whenever any entry in D' changes and use it in equation (8) to compute $\mathbf{c}_p$ and thereby solve the modified problem P'.

Now we relate the solution of problem P' to the main optimization problem. Since $Q_{ii}=E_{ii}^{-2}\in[1/2, 2]$

$$(11)\qquad \frac{1}{2}\mathbf{x}^T\mathbf{x} \leqq \mathbf{x}^T Q\mathbf{x} \leqq 2\mathbf{x}^T\mathbf{x}.$$

Hence

$$B\left(\mathbf{a}_0, \frac{\alpha'}{2}r\right) \leqq \{\mathbf{x}|\mathbf{x}^T Q\mathbf{x} \leqq \alpha' r\} \subseteqq B(\mathbf{a}_0, 2\alpha' r).$$

Taking $\alpha'=\alpha/2$, and taking intersection with Ω''

$$(12)\qquad B(\mathbf{a}_0, \alpha r/4)\cap\Omega'' \subseteqq \{\mathbf{x}|\mathbf{x}^T Q\mathbf{x} \leqq \alpha' r, B\mathbf{x} = 0\} \subseteqq B(\mathbf{a}_0, \alpha r)\cap\Omega''.$$

Because of the first inclusion,

$$(13)\qquad \begin{aligned} &\min\{f'(\mathbf{x})|\mathbf{x}^T Q\mathbf{x} \leqq \alpha' r, B\mathbf{x} = \mathbf{0}\} \\ &\leqq \min\{f'(\mathbf{x})|\mathbf{x}\in B(\mathbf{a}_0, \alpha r/4)\cap\Omega''\} \leqq f'(\mathbf{a}_0)-\ln(1+\alpha/4). \end{aligned}$$

The last inequality follows from Theorem 3. Thus for the modified algorithm, the claim of Theorem 3 is modified as follows

$$(14)\qquad f'(\mathbf{b}') \leqq f'(\mathbf{a}_0)-\ln(1+\alpha/4).$$

Because of the second inclusion, Lemma 4.2 continues to be valid and we can approximate minimization of $f'(\mathbf{x})$ by minimization of a linear function. The claim (as in Theorem 2) about one step of the algorithm $f(\mathbf{x}^{(k+1)})\leqq f(\mathbf{x}^{(k)})-\delta$ also remains valid where δ is redefined as

$$(15)\qquad \delta = \ln(1+\alpha/4)-\frac{\beta^2}{2(1-\beta)}.$$

This affects the number of steps by only a constant factor and the algorithm still works correctly.

6.3. Performance of the modified algorithm

In this Subsection we show that the total number of rank-one updating operations in m steps of the modified algorithm is $O(m\sqrt{n})$. Since each rank-one modification requires $O(n^2)$ arithmetic operations, the average work per step is $O(n^{2.5})$ as compared to $O(n^3)$ in the simpler form of the algorithm.

In each step $|\mathbf{b}'-\mathbf{a}_0|\leqq\alpha r$. Substituting $\mathbf{b}'=T(\mathbf{x}^{(k+1)})$, $\mathbf{a}_0=n^{-1}\mathbf{e}$ and $T(\mathbf{x})=\dfrac{D^{-1}\mathbf{x}}{\mathbf{e}^TD^{-1}\mathbf{x}}$, $r=\dfrac{1}{\sqrt{n(n-1)}}$

$$\sum_i\left[\frac{x_i^{(k+1)}/x_i^{(k)}}{\sum_j x_j^{(k+1)}/x_j^{(k)}}-\frac{1}{n}\right]^2\leqq\frac{\alpha^2}{n(n-1)}.$$

Let

$$\sigma^{(k)}=\frac{1}{n}\sum_j x_j^{(k+1)}/x_j^{(k)}. \tag{16}$$

Then

$$\sum_i\left[\frac{x_i^{(k+1)}}{x_i^{(k)}\sigma^{(k)}}-1\right]^2\leqq\beta^2.$$

Let

$$\Delta_i^{(k)}=\frac{x_i^{(k+1)}}{x_i^{(k)}\sigma^{(k)}}. \tag{17}$$

Therefore $\sum_i[\Delta_i^{(k)}-1]^2\leqq\beta^2$. Recall that $D^{(k)}=\mathrm{Diag}\,\{x_1^{(k)},x_2^{(k)},\dots,x_n^{(k)}\}$, $D^{(k+1)}=\mathrm{Diag}\,\{x_1^{(k+1)},x_2^{(k+1)},\dots,x_n^{(k+1)}\}$. $D'^{(k)}$ is updated in two stages.

First we scale $D'^{(k)}$ by a factor $\sigma^{(k)}$ defined in equation (16)

$$D'^{(k+1)}=\sigma^{(k)}D'^{(k)}. \tag{18}$$

Then for each entry i such that

$$\left[\frac{D_{ii}'^{(k+1)}}{D_{ii}^{(k+1)}}\right]^2\notin\left[\frac{1}{2},\ 2\right]$$

we reset $D_{ii}'^{(k+1)}=D_{ii}^{(k+1)}$.

Define discrepancy between $D^{(k)}$ and $D'^{(k)}$ as

$$\delta_i^{(k)}=\ln\left(\frac{D_{ii}'^{(k)}}{D_{ii}^{(k)}}\right). \tag{19}$$

Just after an updating operation for index i,

$$\delta_i^{(k)}=0.$$

Just before an updating operation for index i,

$$|\delta_i^{(k)}|\geqq\ln\sqrt{2}.$$

Between two successive updates, say at steps k_1 and k_2,

$$\delta_i^{(k_2)} - \delta_i^{(k_1)} = \sum_{k=k_1}^{k_2-1} [\delta_i^{(k+1)} - \delta_i^{(k)}].$$

Let $\Delta\delta_i^{(k)}$ be the change in the discrepancy at step k.

$$\Delta\delta_i^{(k)} = \delta_i^{(k+1)} - \delta_i^{(k)}. \tag{20}$$

Then

$$\ln \sqrt{2} \leqq |\delta_i^{(k_2)}| = |\delta_i^{(k_2)} - \delta_i^{(k_1)}| \leqq \sum_{k=k_1}^{k_2-1} |\Delta\delta_i^{(k)}|. \tag{21}$$

Now we analyze the change in discrepancy in step k, assuming that no updating operation was performed for index i.

$$\Delta\delta_i^{(k)} = \delta_i^{(k+1)} - \delta_i^{(k)} = \ln \frac{D_{ii}'^{(k+1)}}{D_{ii}^{(k+1)}} \frac{D_{ii}^{(k)}}{D_{ii}'^{(k)}} = \ln \frac{\sigma^{(k)} x_i^{(k)}}{x_i^{(k+1)}} = -\ln \Delta_i^{(k)}$$

$$|\Delta\delta_i^{(k)}| = |\ln \Delta_i^{(k)}|. \tag{22}$$

Let $n_i =$ the number of updating operations corresponding to index i in m steps of the algorithm and $N = \sum_{i=1}^{n} n_i$, be the total number of updating operations in m steps.

By equations (20) and (21),

$$\ln \sqrt{2}\, n_i \leqq \sum_{k=1}^{m} |\ln \Delta_i^{(k)}|$$

$$\ln \sqrt{2}\, N \leqq \sum_{i=1}^{n} \sum_{k=1}^{m} |\ln \Delta_i^{(k)}|. \tag{23}$$

We use equation (17) to bound the R.H.S. of equation (23)

$$\sum_i [\Delta_i^{(k)} - 1]^2 \leqq \beta^2 \Rightarrow |\Delta_i^{(k)} - 1| \leqq \beta$$

$$\Rightarrow |\ln \Delta_i^{(k)} - (\Delta_i^{(k)} - 1)| \leqq \frac{(\Delta_i^{(k)} - 1)^2}{2(1-\beta)} \quad \text{by Lemma 4.1}$$

$$\sum_i [\Delta_i^{(k)} - 1]^2 \leqq \beta^2 \Rightarrow \sum_i |\Delta_i^{(k)} - 1| \leqq \sqrt{n}\,\beta.$$

Hence

$$\begin{aligned}\sum_{i=1}^{n} |\ln \Delta_i^{(k)}| &= \sum_{i=1}^{n} |\ln \Delta_i^{(k)} - (\Delta_i^{(k)} - 1) + (\Delta_i^{(k)} - 1)| \\ &\leqq \sum_{i=1}^{n} |\ln \Delta_i^{(k)} - (\Delta_i^{(k)} - 1)| + \sum_{i=1}^{n} |\Delta_i^{(k)} - 1| \\ &\leqq \sum_i \frac{[\Delta_i^{(k)} - 1]^2}{2(1-\beta)} + \sqrt{n}\,\beta \\ &\leqq \frac{\beta^2}{2(1-\beta)} + \sqrt{n}\,\beta.\end{aligned}$$

(24) $$\sum_{i=1}^{n} \sum_{k=1}^{m} |\ln \Delta_i^{(k)}| \leqq m\left[\frac{\beta^2}{2(1-\beta)} + \sqrt{n}\,\beta\right] = O(m\sqrt{n}).$$

From equations (22) and (24)

(25) $$N = O(m\sqrt{n}).$$

7. Concluding remarks

7.1. Global analysis of optimization algorithms

While theoretical methods for analyzing local convergence of non-linear programming and other geometric algorithms are well-developed the state-of-the-art of global convergence analysis is rather unsatisfactory. The algorithmic and analytical techniques introduced in this paper may turn out to be valuable in designing geometric algorithms with provably good global convergence properties. Our method can be thought of as a steepest descent method with respect to a particular metric space over a simplex defined in terms of "cross-ratio", a projective invariant. The global nature of our result was made possible because any (strictly interior) point in the feasible region can be mapped to any other such point by a transformation that preserves affine spaces as well as the metric. This metric can be easily generalized to arbitrary convex sets with "well-behaved" boundary and to intersections of such convex sets. This metric effectively transforms the feasible region so that the boundary of the region is at infinite distance from the interior points. Furthermore, this transformation is independent of the objective function being optimized. Contrast this with the penalty function methods which require an ad hoc mixture of objective function and penalty function. It is not clear a priori in what proportion the two functions should be mixed and the right proportion depends on both the objective function and the feasible region.

7.2. Comments on the factor "L" in running time

Since representing the output of a linear programming problem requires $O(L)$ bits per variable in the worst case, the factor L must appear in the worst-case complexity of any algorithm for linear programming. In practice L is much smaller than n. Therefore, whenever it is possible to replace a factor of n by L, one should always do so if one is interested in an algorithm which is efficient in practice.

7.3. Performance in practice

Each step of the algorithm requires optimization of a linear function over an ellipsoid or equivalently, solution of a system of linear equations of the type $(ADA^T)\mathbf{x}=\mathbf{b}$, where A is a fixed matrix and D is a diagonal matrix with positive entries which changes by small amount from step to step. We devise a method based on successive rank-one modifications and prove a worst-case bound of $O(n^{2.5})$ arithmetic operations per step.

In practice, the matrices encountered are sparse, hence more efficient methods for solving the above problem are possible. Another feature of the algorithm which can lead to computational saving is that it is not necessary to find the exact solution to the optimization problem stated above. Here it is useful to distinguish between two types of approximate solutions. Let $\mathbf{x}_0$ be the exact solution and $\mathbf{x}$ an approximate solution and let $\mathbf{c}^T\mathbf{x}$ be the objective function. A strong approximation (or approximation in the solution space) requires that $\mathbf{x}$ be close to $\mathbf{x}_0$ in some suitable norm. A weak approximation (or approximation in objective function space) requires that $\mathbf{c}^T\mathbf{x}$ be close to $\mathbf{c}^T\mathbf{x}_0$. A weak approximation is sufficient for our method, and is easier to achieve numerically. The optimization problem in each step is a good candidate for applying iterative methods to obtain approximate solutions. The worst-case bound is based on finding an exact solution.

The number of steps of the algorithm depends on the "R/r" ratio, i.e. the ratio of radius of the sphere circumscribing the polytope to the radius of the inscribed sphere. We prove an $O(n)$ upper bound on this ratio. However it is likely to be smaller than n in typical problems of practical interest and also in the worst-case if the circumscribing sphere is chosen to include only that portion of the polytope having values of objective function better than the current value, and after computing the direction of one step (as described in section 2) its length is chosen to be $\alpha r'$ rather than αr where r' is the largest possible step length without going outside the feasible region. The possibility of improvement in R/r ratio gives rise to a number of research problems for further investigation, such as probabilistic analysis of the problem, experimental measurements on problems arising in practical applications, analysis of special cases of practical interest such as "box" type of constraints and finally, the most difficult of all, obtaining a better worst-case bound.

7.4. Numerical stability and round-off errors

In the ellipsoid algorithm, the round-off errors in numerical computation accumulate from step to step. The amount of precision required in arithmetic operations grows with the number of steps. In our algorithm, errors do not accumulate. On the contrary, the algorithm is self-correcting in the sense that the round-off error made in one step gets compensated for by future steps. If we allow 1% error in computation of the vector $\mathbf{x}^{(k)}$ in each step, we can compensate for the errors by running the algorithm 2% more steps.

7.5. Multiple columns and parallel computation

In the simplex method, the current solution is modified by introducing a non-zero coefficient for one of the columns in the constraint matrix. Our method allows the current solution to be modified by introducing several columns at once. In the algorithm described in this paper, all columns are "active", but a variation in which a suitably chosen subset of columns is active at one time is possible. It seems that on a highly parallel architecture a method that uses many columns at once may be preferable to the simplex method.

7.6. Other areas for further research

Since many polynomial-time solvable combinatorial problems are special cases of linear programming, this work raises the possibility of applying the new techniques to such special cases. Problems solved by the column generation method require a special variant of the algorithm.

Acknowledgement. The author wishes to thank Arjen K. Lenstra, Mark Manasse, Joan Plumstead, Alon Itai, László Babai, Mike Garey and Ron Graham for pointing out errors and making suggestions for improvement.

References

[1] H. S. M. Coxeter, *Introduction to Geometry*, Wiley (1961).
[2] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ (1963).
[3] M. Grötschel, L. Lovász and A. Schrijver, The Ellipsoid Method and its Consequences in Combinatorial Optimization, *Combinatorica* **1** (1981), 169—197.
[4] L. G. Khachiyan, A polynomial Algorithm in Linear Programming, *Doklady Akademii Nauk SSSR* **244**:S (1979), 1093—1096, translated in *Soviet Mathematics Doklady* **20**:1 (1979), 191—194.
[5] V. Klee and G. L. Minty, How good is the simplex algorithm? in *Inequalities III*, (ed. O. Shisha) Academic Press, New York, 1972, 159-179.
[6] O. Veblen and J. W. Young, *Projective Geometry, 1—2*, Blaisdell, New York, (1938).
[7] R. J. Walker, *Algebraic Curves*, Princeton University Press (1950).

N. Karmarkar
AT&T Bell Laboratories
Murray Hill, NJ 07974
U.S.A.

On the Motion of Small Particles Suspended in Liquids at Rest Required by the Molecular-Kinetic Theory of Heat*

Albert Einstein

*Originally published in *Annalen der Physik* 17 [1905]: 549-560

IN THIS PAPER it will be shown that, according to the molecular-kinetic theory of heat, bodies of a microscopically visible size suspended in liquids must, as a result of thermal molecular motions, perform motions of such magnitude that they can be easily observed with a microscope. It is possible that the motions to be discussed here are identical with so-called Brownian molecular motion; however, the data available to me on the latter are so imprecise that I could not form a judgment on the question. If the motion to be discussed here can actually be observed, together with the laws it is expected to obey, then classical thermodynamics can no longer be viewed as applying to regions that can be distinguished even with a microscope, and an exact determination of actual atomic sizes becomes possible. On the other hand, if the prediction of this motion were to be proved wrong, this fact would provide a far-reaching argument against the molecular-kinetic conception of heat.

1 On the Osmotic Pressure to be Ascribed to Suspended Particles

Let z gram-molecules of a non-electrolyte be dissolved in a part V^* of the total volume V of a liquid. If the volume V^* is separated from the pure solvent by a wall that is permeable to the solvent but not to the solute, then this wall is subjected to a so-called osmotic pressure, which for sufficiently large values of V^*/z satisfies the equation

$$pV^* = RTz. \tag{1}$$

But if instead of the solute, the partial volume V^* of the liquid contains small suspended bodies that also cannot pass through the solvent-permeable wall, then, according to the classical theory of thermodynamics, we should not expect–at least if we neglect the force of gravity, which does not interest us here–any pressure to be exerted on the wall; for according to the usual interpretation, the "free energy" of the system does not seem to depend on the position of the wall and of the suspended bodies, but only on the total mass and properties of the suspended substance, the liquid, and the wall, as well as on the pressure and temperature. To be sure, the energy and entropy of the interfaces (capillary forces) should also be considered when calculating the free energy; but we can disregard them here because changes in the position of the wall and suspended bodies will not cause changes in the size and state of the contact surfaces.

But a different interpretation arises from the standpoint of the molecular-kinetic theory of heat. According to this theory, a dissolved molecule differs from a suspended body only in size, and it is difficult to see why suspended bodies should not produce the same osmotic pressure as an equal number of dissolved molecules. We have to assume that the suspended bodies perform an irregular, albeit very slow, motion in the liquid due to the liquid's molecular motion; if prevented by the wall from leaving the volume V^*, they will exert pressure upon the wall just like molecules in solution. Thus, if n suspended bodies are present in the volume V^*, i.e., $n/V^* = \nu$ in a unit volume, and

if neighboring bodies are sufficiently far separated from each other, there will be a corresponding osmotic pressure p of magnitude

$$p = \frac{RT}{V^*}\frac{n}{N} = \frac{RT}{N} \cdot \nu, \tag{2}$$

where N denotes the number of actual molecules per gram-molecule. It will be shown in the next section that the molecular-kinetic theory of heat does indeed lead to this broader interpretation of osmotic pressure.

2 Osmotic Pressure from the Standpoint of the Molecular-Kinetic Theory of Heat [1]

If $p_1 p_2 \dots p_l$ are state variables of a physical system that completely determine the system's instantaneous state (e.g., the coordinates and velocity components of all atoms of the system), and if the complete system of equations for changes of these variables is given in the form

$$\frac{\partial p_\nu}{\partial t} = \varphi_\nu(p_1 \dots p_l) \;\; (\nu = 1, 2, \dots l), \tag{3}$$

with $\sum \frac{\partial \varphi_\nu}{\partial p_\nu} = 0$, then the entropy of the system is given by the expression

$$S = \frac{\bar{E}}{T} + 2\kappa \ln \int e^{-\frac{E}{2\kappa T}} dp_1 \dots dp_l. \tag{4}$$

Here T denotes the absolute temperature, $\bar{E}$ the energy of the system, and E the energy as a function of the p_ν. The integral extends over all possible values of p_ν consistent with the conditions of the problem. κ is connected with the constant N mentioned above by the relation $2\kappa N = R$. Hence we get for the free energy F

$$F = -\frac{R}{N} T \ln \int e^{-\frac{EN}{RT}} dp_1 \dots dp_l = -\frac{RT}{N} \ln B. \tag{5}$$

Let us now imagine a liquid enclosed in volume V; let a part V^* of the volume V contain n solute molecules or suspended bodies, which are retained in the volume V^* by a semipermeable wall; the integration limits of the integral R occurring in the expressions for S and F will be affected accordingly. Let the total volume of the solute molecules or suspended bodies be small compared with V^*. In accordance with the theory mentioned, let this system be completely described by the variables $p_1 \dots p_l$.

Even if the molecular picture were extended to include all details, the calculation of the integral B would he so difficult that an exact calculation of F is hardly conceivable. However, here we only need to know how F depends on the size of the volume V^* in which all the solute molecules or suspended bodies (hereafter called "particles" for brevity) are contained.

Let us call the rectangular coordinates of the center of gravity of the first particle x_1, y_1, z_1, those of the second x_2, y_2, z_2, etc., and those of the last particle x_n, y_n, z_n, and assign to the centers of gravity of the particles the infinitesimally small parallelepiped regions $dx_1 dy_1 dz_1, dx_2 dy_2 dz_2 \dots dx_n dy_n dz_n$, all of which lie in V^*. We want to evaluate the integral occurring in the expression for F, with the restriction that the centers of gravity of the particles shall lie in the regions just assigned to them. In any case, this integral can be put into the form

$$dB = dx_1 dy_1 \dots dz_n \cdot J, \tag{6}$$

[1] In this section it is assumed that the reader is familiar with the author's papers on the foundations of thermodynamics (cf. *Ann. d. Phys.* 9 [1902]: 417 and 11 [1903]: 170). Knowledge of these Papers and of this section of the present paper is not essential for an understanding of the results in the present paper.

where J is independent of $dx_1 dy_1$, etc., as well as of V^*, i.e., of the position of the semipermeable wall. But J is also independent of the particular choice of the *positions* of the center of gravity regions and of the value of V^*, as will be shown immediately. For if a second system of infinitesimally small regions were assigned to the centers of gravity of the particles and denoted by $dx'_1 dy'_1 dz'_1, dx'_2 dy'_2 dz'_2 \ldots dx'_n dy'_n dz'_n$, and if these regions differed from the originally assigned ones by their position alone, but not by their size, and if, likewise, all of them were contained in V^*, we would similarly have

$$dB' = dx'_1 dy'_1 \ldots dz'_n \cdot J', \tag{7}$$

where

$$dx_1 dy_1 \ldots dz_n = dx'_1 dy'_1 \ldots dz'_n. \tag{8}$$

Hence,

$$\frac{dB}{dB'} = \frac{J}{J'}. \tag{9}$$

But from the molecular theory of heat, as presented in the papers cited,[2] it is easily deduced that dB/B and dB'/B are respectively equal to the probabilities that at an arbitrarily chosen moment the centers of gravity of the particles will be found in the regions $(dx_1 \ldots dz_n)$ and $(dx'_l \ldots dz'_n)$ respectively. If the motions of the individual particles are independent of one another (to a sufficient approximation) and if the liquid is homogeneous and no forces act on the particles, then for regions of the same size the probabilities of the two systems of regions will be equal, so that

$$\frac{dB}{B} = \frac{dB'}{B}. \tag{10}$$

But from this equation and the previous one it follows that

$$J = J'. \tag{11}$$

This proves that J does not depend on either V^* or $x_1, y_1, \ldots z_n$. By integrating, we get

$$B = \int J dx_l ... dz_n = J V^{*n}, \tag{12}$$

and hence

$$F = -\frac{RT}{N}\{\ln J + n \ln V^*\} \tag{13}$$

and

$$p = -\frac{\partial F}{\partial V^*} = \frac{RT}{V^*}\frac{n}{N} = \frac{RT}{N}\nu. \tag{14}$$

This analysis shows that the existence of osmotic pressure can be deduced from the molecular-kinetic theory of heat, and that, at high dilution, according to this theory, equal numbers of solute molecules and suspended particles behave identically as regards osmotic pressure.

3 Theory of Diffusion of Small Suspended Spheres

Suppose suspended particles are randomly distributed in a liquid. We will investigate their state of dynamic equilibrium under the assumption that a force K, which depends on the position but not on the time, acts on the individual particles. For the sake of simplicity, we shall assume that the force acts everywhere in the direction of the X-axis.

[2] A. Einstein, *Ann. d. Phys.* 11 (1903): 170.

If the number of suspended particles per unit volume is ν, then in the case of thermodynamic equilibrium ν is a function of x such that the variation of the free energy vanishes for an arbitrary virtual displacement δx of the suspended substance. Thus

$$\delta F = \delta E - T\,\delta S = 0. \tag{15}$$

Let us assume that the liquid has a unit cross section perpendicular to the X-axis, and that it is bounded by the planes $x = 0$ and $x = l$. We then have

$$\delta E = -\int_0^l K\nu\,\delta x\,dx \tag{16}$$

and

$$\delta S = \int_0^l R\frac{\nu}{N}\frac{\partial \delta x}{\partial x}dx = -\frac{R}{N}\int_0^l \frac{\partial \nu}{\partial x}\delta x dx. \tag{17}$$

The required equilibrium condition is therefore

$$-K\nu + \frac{RT}{N}\frac{\partial \nu}{\partial x} = 0 \tag{18}$$

or

$$K\nu - \frac{\partial p}{\partial x} = 0. \tag{19}$$

The last equation asserts that the force K is equilibrated by the force of osmotic pressure.

We can use equation (18) to determine the diffusion coefficient of the suspended substance. We can look upon the dynamic equilibrium state considered here as a superposition of two processes proceeding in opposite directions, namely:

1. A motion of the suspended substance under the influence of the force K that acts on each suspended particle.

2. A process of diffusion, which is to be regarded as the result of the disordered motions of the particles produced by thermal molecular motion.

If the suspended particles have spherical form (where P is the radius of the sphere) and the coefficient of viscosity of the liquid is k, then the force K imparts to an individual particle the velocity[3]

$$\frac{K}{6\pi k P}, \tag{20}$$

and

$$\frac{\nu K}{6\pi k P}$$

particles will pass through a unit area per unit time.

Further, if D denotes the diffusion coefficient of the suspended substance and μ the mass of a particle, then

$$-D\frac{\partial(\mu\nu)}{\partial x} \text{ grams} \tag{21}$$

or

$$-D\frac{\partial \nu}{\partial x} \tag{22}$$

[3]Cf., e.g., G. Kirchhoff, *Vorlesungen uber Mechanik*, 26. Vorl., 4 (*Lectures on Mechanics*, Lecture 26, sec. 4).

particles will pass across a unit area per unit time as the result of diffusion. Since dynamic equilibrium prevails, we must have

$$\frac{\nu K}{6\pi k P} - D\frac{\partial \nu}{\partial x} = 0. \tag{23}$$

From the two conditions (18) and (23) found for dvnamic equilibrium, we can calculate the diffusion coefficient. We get

$$D = \frac{RT}{N} \cdot \frac{1}{6\pi k P}. \tag{24}$$

Thus, except for universal constants and the absolute temperature, the diffusion coefficient of the suspended substance depends only on the viscosity of the liquid and on the size of the suspended particles.

4 On the Disordered Motion of Particles Suspended in a Liquid and its Relation to Diffusion

We shall now turn to a closer examination of the disordered motions that arise from thermal molecular motion and give rise to the diffusion investigated in the last section.

Obviously, we must assume that each individual particle executes a motion that is independent of the motions of all the other particles; the motions of the same particle in different time intervals must also be considered as mutually independent processes, so long as we think of these time intervals as chosen not to be too small.

We now introduce a time interval τ, which is very small compared with observable time intervals but still large enough that the notions performed by a particle during two consecutive time intervals τ can be considered as mutually independent events.

Suppose, now, that a total of n suspended particles is present in a liquid. In a time interval τ, the X-coordinates of the individual particles will increase by Δ, where Δ has a different (positive or negative) value for each particle. A certain probability distribution law will hold for Δ: the number dn of particles experiencing a displacement that lies between Δ and $\Delta + d\Delta$ in the time interval τ will be expressed by an equation of the form

$$dn = n\varphi(\Delta)d\Delta, \tag{25}$$

where

$$\int_{-\infty}^{+\infty} \varphi(\Delta)d\Delta = 1, \tag{26}$$

and φ differs from zero only for very small values of Δ and satisfies the condition

$$\varphi(\Delta) = \varphi(-\Delta). \tag{27}$$

We will now investigate how the diffusion coefficient depends on φ, restricting ourselves again to the ease where the number ν of particles per unit volume only depends on x and t.

Let $\nu = f(x,t)$ be the number of particles per unit volume; we calculate the distribution of particles at time $t+\tau$ from their distribution at time t. From the definition of the function $\varphi(\Delta)$ we can easily obtain the number of particles found at time $t+\tau$ between two planes perpendicular to the X-axis with abscissas x and $x+dx$. One obtains

$$f(x,t+\tau)dx = dx \cdot \int_{\Delta=-\infty}^{\Delta=+\infty} f(x+\Delta)\varphi(\Delta)d\Delta. \tag{28}$$

But since τ is very small, we can put

$$f(x, t+\tau) = f(x,t) + \tau \frac{\partial f}{\partial t}. \tag{29}$$

Further, let us expand $f(x+\Delta, t)$ in powers of Δ:

$$f(x+\Delta, t) = f(x,t) + \Delta \frac{\partial f(x,t)}{\partial x} + \frac{\Delta^2}{2!} \frac{\partial^2 f(x,t)}{\partial x^2} \ldots \text{ad inf.} \tag{30}$$

We can bring this expansion under the integral sign since only very small values of Δ contribute anything to the latter. We obtain

$$f + \frac{\partial f}{\partial t} \cdot \tau = f \cdot \int_{-\infty}^{+\infty} \varphi(\Delta) d\Delta + \frac{\partial f}{\partial x} \int_{-\infty}^{+\infty} \Delta \varphi(\Delta) d\Delta + \frac{\partial^2 f}{\partial x^2} \int_{-\infty}^{+\infty} \frac{\Delta^2}{2} \varphi(\Delta) d\Delta \ldots \tag{31}$$

On the right-hand side, the second, fourth, etc., terms vanish since $\varphi(x) = \varphi(-x)$, while for the first, third, fifth, etc., terms, each successive term is very small compared with the one preceding it. From this equation, by taking into account that

$$\int_{-\infty}^{+\infty} \varphi(\Delta) d\Delta = 1, \tag{32}$$

and putting

$$\frac{1}{\tau} \int_{-\infty}^{+\infty} \frac{\Delta^2}{2} \varphi(\Delta) d\Delta = D, \tag{33}$$

and taking into account only the first and third terms of the right-hand side, we get

$$\frac{\partial f}{\partial t} = D \frac{\partial^2 f}{\partial x^2}. \tag{34}$$

This is the well-known differential equation for diffusion, and we recognize that D is the diffusion coefficient.

Another important point can be linked to this argument. We have assumed that the individual particles are all referred to the same coordinate system. However, this is not necessary since the motions of the individual particles are mutually independent. We will now refer the motion of each particle to a coordinate system whose origin coincides with the position of the center of gravity of the particle in question at time $t = 0$, with the difference that $f(x,t)dx$ now denotes the number of particles whose X-coordinate has increased between the times $t = 0$ and $t = t$ by a quantity that lies somewhere between x and $x + dx$. Thus, the function f varies according to equation (23) in this case as well. Further, it is obvious that for $x \neq 0$ and $t = 0$ we must have

$$f(x,t) = 0 \text{ and } \int_{-\infty}^{+\infty} f(x,t) dx = n. \tag{35}$$

The problem, which coincides with the problem of diffusion outwards from a point (neglecting the interaction between the diffusing particles), is now completely determined mathematically; its solution is

$$f(x,t) = \frac{n}{\sqrt{4\pi D}} \frac{e^{-\frac{x^2}{4Dt}}}{\sqrt{t}}. \tag{36}$$

The probability distribution of the resulting displacements during an arbitrary time t is thus the same as the distribution of random errors, which was to be expected. What is important, however, is how the constant in the exponent is related to the diffusion coefficient. With the help

of this equation we can now calculate the displacement λ_x in the direction of the X-axis that a particle experiences on the average, or, to be more precise, the root-mean-square displacement in the X-direction; it is

$$\lambda_x = \sqrt{\overline{x^2}} = \sqrt{2Dt}. \tag{37}$$

The mean displacement is thus proportional to the square root of the time. It can easily be shown that the root-mean-square of the *total displacements* of the particles has the value $\lambda_x\sqrt{3}$.

5 Formula for the Mean Displacement of Suspended Particles. A New Method of Determining the Actual Size of Atoms

In section 3 we found the following value for the diffusion coefficient D of a substance suspended in a liquid in the form of small spheres of radius P :

$$D = \frac{RT}{N}\frac{1}{6\pi kP}. \tag{38}$$

Further, we found in section 4 that the mean value of the displacements of the particles in the X-direction at time t equals

$$\lambda_x = \sqrt{2Dt}. \tag{39}$$

By eliminating D, we get:

$$\lambda_x = \sqrt{t} \cdot \sqrt{\frac{RT}{N}\frac{1}{3\pi kP}}. \tag{40}$$

This equation shows how λ_x depends on T, k, and P.

We will now calculate how large λ_x is for one second if N is taken to be $6 \cdot 10^{23}$ in accordance with the results of the kinetic theory of gases; water at 17^oC ($k = 1.35 \cdot 10^{-2}$) is chosen as the liquid, and the diameter of the particles is 0.001 mm. We get

$$\lambda_x = 8 \cdot 10^{-5} \text{ cm } = 0.8 \text{ micron}. \tag{41}$$

Therefore, the mean displacement in one minute would be about 6 microns.

Conversely the relation can be used to determine N. We obtain

$$N = \frac{t}{\lambda_x^2} \cdot \frac{RT}{3\pi kP}. \tag{42}$$

Let us hope that a researcher will soon succeed in solving the problem presented here, which is so important for the theory of heat.

Reprinted with corrections from *The Bell System Technical Journal,*
Vol. 27, pp. 379–423, 623–656, July, October, 1948.

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[1] and Hartley[2] on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

If the number of messages in the set is finite then this number or any monotonic function of this number can be regarded as a measure of the information produced when one message is chosen from the set, all choices being equally likely. As was pointed out by Hartley the most natural choice is the logarithmic function. Although this definition must be generalized considerably when we consider the influence of the statistics of the message and when we have a continuous range of messages, we will in all cases use an essentially logarithmic measure.

The logarithmic measure is more convenient for various reasons:

1. It is practically more useful. Parameters of engineering importance such as time, bandwidth, number of relays, etc., tend to vary linearly with the logarithm of the number of possibilities. For example, adding one relay to a group doubles the number of possible states of the relays. It adds 1 to the base 2 logarithm of this number. Doubling the time roughly squares the number of possible messages, or doubles the logarithm, etc.

2. It is nearer to our intuitive feeling as to the proper measure. This is closely related to (1) since we intuitively measures entities by linear comparison with common standards. One feels, for example, that two punched cards should have twice the capacity of one for information storage, and two identical channels twice the capacity of one for transmitting information.

3. It is mathematically more suitable. Many of the limiting operations are simple in terms of the logarithm but would require clumsy restatement in terms of the number of possibilities.

The choice of a logarithmic base corresponds to the choice of a unit for measuring information. If the base 2 is used the resulting units may be called binary digits, or more briefly *bits,* a word suggested by J. W. Tukey. A device with two stable positions, such as a relay or a flip-flop circuit, can store one bit of information. N such devices can store N bits, since the total number of possible states is 2^N and $\log_2 2^N = N$. If the base 10 is used the units may be called decimal digits. Since

$$\begin{aligned}\log_2 M &= \log_{10} M / \log_{10} 2 \\ &= 3.32 \log_{10} M,\end{aligned}$$

[1]Nyquist, H., "Certain Factors Affecting Telegraph Speed," *Bell System Technical Journal,* April 1924, p. 324; "Certain Topics in Telegraph Transmission Theory," *A.I.E.E. Trans.,* v. 47, April 1928, p. 617.

[2]Hartley, R. V. L., "Transmission of Information," *Bell System Technical Journal,* July 1928, p. 535.

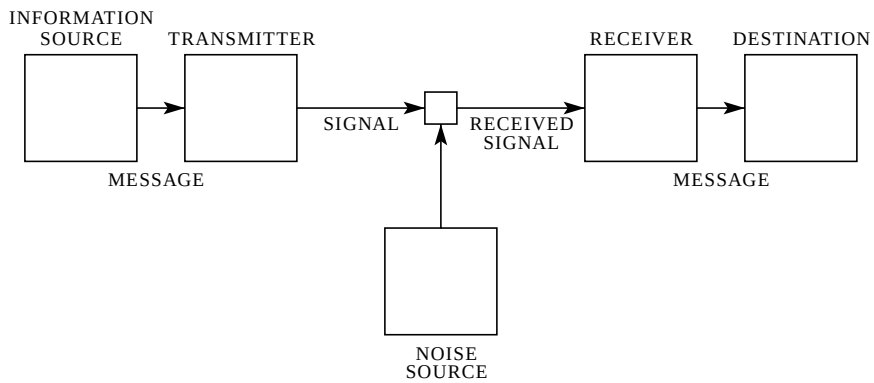


Fig. 1—Schematic diagram of a general communication system.

a decimal digit is about $3\frac{1}{3}$ bits. A digit wheel on a desk computing machine has ten stable positions and therefore has a storage capacity of one decimal digit. In analytical work where integration and differentiation are involved the base e is sometimes useful. The resulting units of information will be called natural units. Change from the base a to base b merely requires multiplication by $\log_b a$.

By a communication system we will mean a system of the type indicated schematically in Fig. 1. It consists of essentially five parts:

1. An *information source* which produces a message or sequence of messages to be communicated to the receiving terminal. The message may be of various types: (a) A sequence of letters as in a telegraph of teletype system; (b) A single function of time $f(t)$ as in radio or telephony; (c) A function of time and other variables as in black and white television — here the message may be thought of as a function $f(x,y,t)$ of two space coordinates and time, the light intensity at point (x,y) and time t on a pickup tube plate; (d) Two or more functions of time, say $f(t)$, $g(t)$, $h(t)$ — this is the case in "three-dimensional" sound transmission or if the system is intended to service several individual channels in multiplex; (e) Several functions of several variables — in color television the message consists of three functions $f(x,y,t)$, $g(x,y,t)$, $h(x,y,t)$ defined in a three-dimensional continuum — we may also think of these three functions as components of a vector field defined in the region — similarly, several black and white television sources would produce "messages" consisting of a number of functions of three variables; (f) Various combinations also occur, for example in television with an associated audio channel.

2. A *transmitter* which operates on the message in some way to produce a signal suitable for transmission over the channel. In telephony this operation consists merely of changing sound pressure into a proportional electrical current. In telegraphy we have an encoding operation which produces a sequence of dots, dashes and spaces on the channel corresponding to the message. In a multiplex PCM system the different speech functions must be sampled, compressed, quantized and encoded, and finally interleaved properly to construct the signal. Vocoder systems, television and frequency modulation are other examples of complex operations applied to the message to obtain the signal.

3. The *channel* is merely the medium used to transmit the signal from transmitter to receiver. It may be a pair of wires, a coaxial cable, a band of radio frequencies, a beam of light, etc.

4. The *receiver* ordinarily performs the inverse operation of that done by the transmitter, reconstructing the message from the signal.

5. The *destination* is the person (or thing) for whom the message is intended.

We wish to consider certain general problems involving communication systems. To do this it is first necessary to represent the various elements involved as mathematical entities, suitably idealized from their

physical counterparts. We may roughly classify communication systems into three main categories: discrete, continuous and mixed. By a discrete system we will mean one in which both the message and the signal are a sequence of discrete symbols. A typical case is telegraphy where the message is a sequence of letters and the signal a sequence of dots, dashes and spaces. A continuous system is one in which the message and signal are both treated as continuous functions, e.g., radio or television. A mixed system is one in which both discrete and continuous variables appear, e.g., PCM transmission of speech.

We first consider the discrete case. This case has applications not only in communication theory, but also in the theory of computing machines, the design of telephone exchanges and other fields. In addition the discrete case forms a foundation for the continuous and mixed cases which will be treated in the second half of the paper.

PART I: DISCRETE NOISELESS SYSTEMS

1. The Discrete Noiseless Channel

Teletype and telegraphy are two simple examples of a discrete channel for transmitting information. Generally, a discrete channel will mean a system whereby a sequence of choices from a finite set of elementary symbols $S_1, \ldots, S_n$ can be transmitted from one point to another. Each of the symbols S_i is assumed to have a certain duration in time t_i seconds (not necessarily the same for different S_i, for example the dots and dashes in telegraphy). It is not required that all possible sequences of the S_i be capable of transmission on the system; certain sequences only may be allowed. These will be possible signals for the channel. Thus in telegraphy suppose the symbols are: (1) A dot, consisting of line closure for a unit of time and then line open for a unit of time; (2) A dash, consisting of three time units of closure and one unit open; (3) A letter space consisting of, say, three units of line open; (4) A word space of six units of line open. We might place the restriction on allowable sequences that no spaces follow each other (for if two letter spaces are adjacent, it is identical with a word space). The question we now consider is how one can measure the capacity of such a channel to transmit information.

In the teletype case where all symbols are of the same duration, and any sequence of the 32 symbols is allowed the answer is easy. Each symbol represents five bits of information. If the system transmits n symbols per second it is natural to say that the channel has a capacity of $5n$ bits per second. This does not mean that the teletype channel will always be transmitting information at this rate — this is the maximum possible rate and whether or not the actual rate reaches this maximum depends on the source of information which feeds the channel, as will appear later.

In the more general case with different lengths of symbols and constraints on the allowed sequences, we make the following definition:

Definition: The capacity C of a discrete channel is given by

$$C = \operatorname*{Lim}_{T\to\infty} \frac{\log N(T)}{T}$$

where $N(T)$ is the number of allowed signals of duration T.

It is easily seen that in the teletype case this reduces to the previous result. It can be shown that the limit in question will exist as a finite number in most cases of interest. Suppose all sequences of the symbols $S_1, \ldots, S_n$ are allowed and these symbols have durations $t_1, \ldots, t_n$. What is the channel capacity? If $N(t)$ represents the number of sequences of duration t we have

$$N(t) = N(t-t_1) + N(t-t_2) + \cdots + N(t-t_n).$$

The total number is equal to the sum of the numbers of sequences ending in $S_1, S_2, \ldots, S_n$ and these are $N(t-t_1), N(t-t_2), \ldots, N(t-t_n)$, respectively. According to a well-known result in finite differences, $N(t)$ is then asymptotic for large t to X_0^t where X_0 is the largest real solution of the characteristic equation:

$$X^{-t_1} + X^{-t_2} + \cdots + X^{-t_n} = 1$$

and therefore

$$C = \log X_0.$$

In case there are restrictions on allowed sequences we may still often obtain a difference equation of this type and find C from the characteristic equation. In the telegraphy case mentioned above

$$N(t) = N(t-2) + N(t-4) + N(t-5) + N(t-7) + N(t-8) + N(t-10)$$

as we see by counting sequences of symbols according to the last or next to the last symbol occurring. Hence C is $-\log \mu_0$ where μ_0 is the positive root of $1 = \mu^2 + \mu^4 + \mu^5 + \mu^7 + \mu^8 + \mu^{10}$. Solving this we find $C = 0.539$.

A very general type of restriction which may be placed on allowed sequences is the following: We imagine a number of possible states $a_1, a_2, \ldots, a_m$. For each state only certain symbols from the set $S_1, \ldots, S_n$ can be transmitted (different subsets for the different states). When one of these has been transmitted the state changes to a new state depending both on the old state and the particular symbol transmitted. The telegraph case is a simple example of this. There are two states depending on whether or not a space was the last symbol transmitted. If so, then only a dot or a dash can be sent next and the state always changes. If not, any symbol can be transmitted and the state changes if a space is sent, otherwise it remains the same. The conditions can be indicated in a linear graph as shown in Fig. 2. The junction points correspond to the

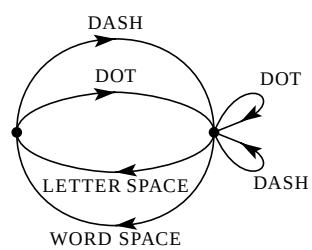


Fig. 2—Graphical representation of the constraints on telegraph symbols.

states and the lines indicate the symbols possible in a state and the resulting state. In Appendix 1 it is shown that if the conditions on allowed sequences can be described in this form C will exist and can be calculated in accordance with the following result:

Theorem 1: Let $b_{ij}^{(s)}$ be the duration of the sth symbol which is allowable in state i and leads to state j. Then the channel capacity C is equal to $\log W$ where W is the largest real root of the determinant equation:

$$\left| \sum_s W^{-b_{ij}^{(s)}} - \delta_{ij} \right| = 0$$

where $\delta_{ij} = 1$ if $i = j$ and is zero otherwise.

For example, in the telegraph case (Fig. 2) the determinant is:

$$\begin{vmatrix} -1 & (W^{-2} + W^{-4}) \\ (W^{-3} + W^{-6}) & (W^{-2} + W^{-4} - 1) \end{vmatrix} = 0.$$

On expansion this leads to the equation given above for this case.

2. The Discrete Source of Information

We have seen that under very general conditions the logarithm of the number of possible signals in a discrete channel increases linearly with time. The capacity to transmit information can be specified by giving this rate of increase, the number of bits per second required to specify the particular signal used.

We now consider the information source. How is an information source to be described mathematically, and how much information in bits per second is produced in a given source? The main point at issue is the effect of statistical knowledge about the source in reducing the required capacity of the channel, by the use

of proper encoding of the information. In telegraphy, for example, the messages to be transmitted consist of sequences of letters. These sequences, however, are not completely random. In general, they form sentences and have the statistical structure of, say, English. The letter E occurs more frequently than Q, the sequence TH more frequently than XP, etc. The existence of this structure allows one to make a saving in time (or channel capacity) by properly encoding the message sequences into signal sequences. This is already done to a limited extent in telegraphy by using the shortest channel symbol, a dot, for the most common English letter E; while the infrequent letters, Q, X, Z are represented by longer sequences of dots and dashes. This idea is carried still further in certain commercial codes where common words and phrases are represented by four- or five-letter code groups with a considerable saving in average time. The standardized greeting and anniversary telegrams now in use extend this to the point of encoding a sentence or two into a relatively short sequence of numbers.

We can think of a discrete source as generating the message, symbol by symbol. It will choose successive symbols according to certain probabilities depending, in general, on preceding choices as well as the particular symbols in question. A physical system, or a mathematical model of a system which produces such a sequence of symbols governed by a set of probabilities, is known as a stochastic process.[3] We may consider a discrete source, therefore, to be represented by a stochastic process. Conversely, any stochastic process which produces a discrete sequence of symbols chosen from a finite set may be considered a discrete source. This will include such cases as:

1. Natural written languages such as English, German, Chinese.

2. Continuous information sources that have been rendered discrete by some quantizing process. For example, the quantized speech from a PCM transmitter, or a quantized television signal.

3. Mathematical cases where we merely define abstractly a stochastic process which generates a sequence of symbols. The following are examples of this last type of source.

   (A) Suppose we have five letters A, B, C, D, E which are chosen each with probability .2, successive choices being independent. This would lead to a sequence of which the following is a typical example.

   B D C B C E C C C A D C B D D A A E C E E A
   A B B D A E E C A C E E B A E E C B C E A D.

   This was constructed with the use of a table of random numbers.[4]

   (B) Using the same five letters let the probabilities be .4, .1, .2, .2, .1, respectively, with successive choices independent. A typical message from this source is then:

   A A A C D C B D C E A A D A D A C E D A
   E A D C A B E D A D D C E C A A A A A D.

   (C) A more complicated structure is obtained if successive symbols are not chosen independently but their probabilities depend on preceding letters. In the simplest case of this type a choice depends only on the preceding letter and not on ones before that. The statistical structure can then be described by a set of transition probabilities $p_i(j)$, the probability that letter i is followed by letter j. The indices i and j range over all the possible symbols. A second equivalent way of specifying the structure is to give the "digram" probabilities $p(i, j)$, i.e., the relative frequency of the digram $i\ j$. The letter frequencies $p(i)$, (the probability of letter i), the transition probabilities

[3] See, for example, S. Chandrasekhar, "Stochastic Problems in Physics and Astronomy," *Reviews of Modern Physics*, v. 15, No. 1, January 1943, p. 1.

[4] Kendall and Smith, *Tables of Random Sampling Numbers*, Cambridge, 1939.

$p_i(j)$ and the digram probabilities $p(i,j)$ are related by the following formulas:

$$p(i) = \sum_j p(i,j) = \sum_j p(j,i) = \sum_j p(j)p_j(i)$$
$$p(i,j) = p(i)p_i(j)$$
$$\sum_j p_i(j) = \sum_i p(i) = \sum_{i,j} p(i,j) = 1.$$

As a specific example suppose there are three letters A, B, C with the probability tables:

$p_i(j)$		j		
		A	B	C
i	A	0	$\frac{4}{5}$	$\frac{1}{5}$
	B	$\frac{1}{2}$	$\frac{1}{2}$	0
	C	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{1}{10}$

i	$p(i)$
A	$\frac{9}{27}$
B	$\frac{16}{27}$
C	$\frac{2}{27}$

$p(i,j)$		j		
		A	B	C
i	A	0	$\frac{4}{15}$	$\frac{1}{15}$
	B	$\frac{8}{27}$	$\frac{8}{27}$	0
	C	$\frac{1}{27}$	$\frac{4}{135}$	$\frac{1}{135}$

A typical message from this source is the following:

A B B A B A B A B A B A B A B B B A B B B B B A B A B A B A B A B B B A C A C A B B A B B B B A B B A B A C B B B A B A.

The next increase in complexity would involve trigram frequencies but no more. The choice of a letter would depend on the preceding two letters but not on the message before that point. A set of trigram frequencies $p(i,j,k)$ or equivalently a set of transition probabilities $p_{ij}(k)$ would be required. Continuing in this way one obtains successively more complicated stochastic processes. In the general n-gram case a set of n-gram probabilities $p(i_1, i_2, \ldots, i_n)$ or of transition probabilities $p_{i_1,i_2,\ldots,i_{n-1}}(i_n)$ is required to specify the statistical structure.

(D) Stochastic processes can also be defined which produce a text consisting of a sequence of "words." Suppose there are five letters A, B, C, D, E and 16 "words" in the language with associated probabilities:

.10 A	.16 BEBE	.11 CABED	.04 DEB
.04 ADEB	.04 BED	.05 CEED	.15 DEED
.05 ADEE	.02 BEED	.08 DAB	.01 EAB
.01 BADD	.05 CA	.04 DAD	.05 EE

Suppose successive "words" are chosen independently and are separated by a space. A typical message might be:

DAB EE A BEBE DEED DEB ADEE ADEE EE DEB BEBE BEBE BEBE ADEE BED DEED DEED CEED ADEE A DEED DEED BEBE CABED BEBE BED DAB DEED ADEB.

If all the words are of finite length this process is equivalent to one of the preceding type, but the description may be simpler in terms of the word structure and probabilities. We may also generalize here and introduce transition probabilities between words, etc.

These artificial languages are useful in constructing simple problems and examples to illustrate various possibilities. We can also approximate to a natural language by means of a series of simple artificial languages. The zero-order approximation is obtained by choosing all letters with the same probability and independently. The first-order approximation is obtained by choosing successive letters independently but each letter having the same probability that it has in the natural language.[5] Thus, in the first-order approximation to English, E is chosen with probability .12 (its frequency in normal English) and W with probability .02, but there is no influence between adjacent letters and no tendency to form the preferred

[5]Letter, digram and trigram frequencies are given in *Secret and Urgent* by Fletcher Pratt, Blue Ribbon Books, 1939. Word frequencies are tabulated in *Relative Frequency of English Speech Sounds,* G. Dewey, Harvard University Press, 1923.

digrams such as TH, ED, etc. In the second-order approximation, digram structure is introduced. After a letter is chosen, the next one is chosen in accordance with the frequencies with which the various letters follow the first one. This requires a table of digram frequencies $p_i(j)$. In the third-order approximation, trigram structure is introduced. Each letter is chosen with probabilities which depend on the preceding two letters.

3. The Series of Approximations to English

To give a visual idea of how this series of processes approaches a language, typical sequences in the approximations to English have been constructed and are given below. In all cases we have assumed a 27-symbol "alphabet," the 26 letters and a space.

1. Zero-order approximation (symbols independent and equiprobable).

   XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZLHJQD.

2. First-order approximation (symbols independent but with frequencies of English text).

   OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

3. Second-order approximation (digram structure as in English).

   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

4. Third-order approximation (trigram structure as in English).

   IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

5. First-order word approximation. Rather than continue with tetragram, . . . , n-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies.

   REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

6. Second-order word approximation. The word transition probabilities are correct but no further structure is included.

   THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

The resemblance to ordinary English text increases quite noticeably at each of the above steps. Note that these samples have reasonably good structure out to about twice the range that is taken into account in their construction. Thus in (3) the statistical process insures reasonable text for two-letter sequences, but four-letter sequences from the sample can usually be fitted into good sentences. In (6) sequences of four or more words can easily be placed in sentences without unusual or strained constructions. The particular sequence of ten words "attack on an English writer that the character of this" is not at all unreasonable. It appears then that a sufficiently complex stochastic process will give a satisfactory representation of a discrete source.

The first two samples were constructed by the use of a book of random numbers in conjunction with (for example 2) a table of letter frequencies. This method might have been continued for (3), (4) and (5), since digram, trigram and word frequency tables are available, but a simpler equivalent method was used.

To construct (3) for example, one opens a book at random and selects a letter at random on the page. This letter is recorded. The book is then opened to another page and one reads until this letter is encountered. The succeeding letter is then recorded. Turning to another page this second letter is searched for and the succeeding letter recorded, etc. A similar process was used for (4), (5) and (6). It would be interesting if further approximations could be constructed, but the labor involved becomes enormous at the next stage.

4. Graphical Representation of a Markoff Process

Stochastic processes of the type described above are known mathematically as discrete Markoff processes and have been extensively studied in the literature.[6] The general case can be described as follows: There exist a finite number of possible "states" of a system; $S_1, S_2, \ldots, S_n$. In addition there is a set of transition probabilities; $p_i(j)$ the probability that if the system is in state S_i it will next go to state S_j. To make this Markoff process into an information source we need only assume that a letter is produced for each transition from one state to another. The states will correspond to the "residue of influence" from preceding letters.

The situation can be represented graphically as shown in Figs. 3, 4 and 5. The "states" are the junction

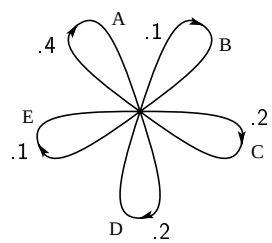


Fig. 3—A graph corresponding to the source in example B.

points in the graph and the probabilities and letters produced for a transition are given beside the corresponding line. Figure 3 is for the example B in Section 2, while Fig. 4 corresponds to the example C. In Fig. 3

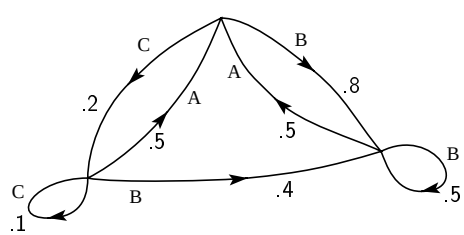


Fig. 4—A graph corresponding to the source in example C.

there is only one state since successive letters are independent. In Fig. 4 there are as many states as letters. If a trigram example were constructed there would be at most n^2 states corresponding to the possible pairs of letters preceding the one being chosen. Figure 5 is a graph for the case of word structure in example D. Here S corresponds to the "space" symbol.

5. Ergodic and Mixed Sources

As we have indicated above a discrete source for our purposes can be considered to be represented by a Markoff process. Among the possible discrete Markoff processes there is a group with special properties of significance in communication theory. This special class consists of the "ergodic" processes and we shall call the corresponding sources ergodic sources. Although a rigorous definition of an ergodic process is somewhat involved, the general idea is simple. In an ergodic process every sequence produced by the process

[6] For a detailed treatment see M. Fréchet, *Méthode des fonctions arbitraires. Théorie des événements en chaîne dans le cas d'un nombre fini d'états possibles*. Paris, Gauthier-Villars, 1938.

is the same in statistical properties. Thus the letter frequencies, digram frequencies, etc., obtained from particular sequences, will, as the lengths of the sequences increase, approach definite limits independent of the particular sequence. Actually this is not true of every sequence but the set for which it is false has probability zero. Roughly the ergodic property means statistical homogeneity.

All the examples of artificial languages given above are ergodic. This property is related to the structure of the corresponding graph. If the graph has the following two properties[7] the corresponding process will be ergodic:

1. The graph does not consist of two isolated parts A and B such that it is impossible to go from junction points in part A to junction points in part B along lines of the graph in the direction of arrows and also impossible to go from junctions in part B to junctions in part A.

2. A closed series of lines in the graph with all arrows on the lines pointing in the same orientation will be called a "circuit." The "length" of a circuit is the number of lines in it. Thus in Fig. 5 series BEBES is a circuit of length 5. The second property required is that the greatest common divisor of the lengths of all circuits in the graph be one.

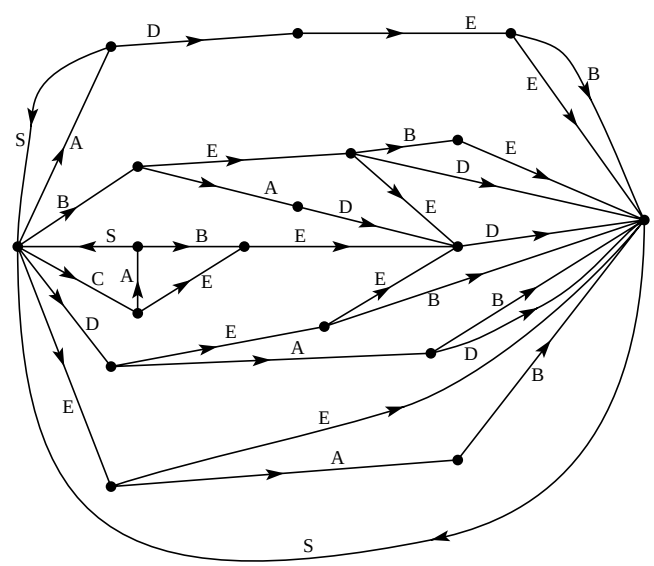


Fig. 5—A graph corresponding to the source in example D.

If the first condition is satisfied but the second one violated by having the greatest common divisor equal to $d > 1$, the sequences have a certain type of periodic structure. The various sequences fall into d different classes which are statistically the same apart from a shift of the origin (i.e., which letter in the sequence is called letter 1). By a shift of from 0 up to $d-1$ any sequence can be made statistically equivalent to any other. A simple example with $d = 2$ is the following: There are three possible letters a, b, c. Letter a is followed with either b or c with probabilities $\frac{1}{3}$ and $\frac{2}{3}$ respectively. Either b or c is always followed by letter a. Thus a typical sequence is

$$a\ b\ a\ c\ a\ c\ a\ c\ a\ b\ a\ c\ a\ b\ a\ b\ a\ c\ a\ c.$$

This type of situation is not of much importance for our work.

If the first condition is violated the graph may be separated into a set of subgraphs each of which satisfies the first condition. We will assume that the second condition is also satisfied for each subgraph. We have in this case what may be called a "mixed" source made up of a number of pure components. The components correspond to the various subgraphs. If $L_1, L_2, L_3, \ldots$ are the component sources we may write

$$L = p_1 L_1 + p_2 L_2 + p_3 L_3 + \cdots$$

[7]These are restatements in terms of the graph of conditions given in Fréchet.

where p_i is the probability of the component source L_i.

Physically the situation represented is this: There are several different sources $L_1, L_2, L_3, \ldots$ which are each of homogeneous statistical structure (i.e., they are ergodic). We do not know *a priori* which is to be used, but once the sequence starts in a given pure component L_i, it continues indefinitely according to the statistical structure of that component.

As an example one may take two of the processes defined above and assume $p_1 = .2$ and $p_2 = .8$. A sequence from the mixed source

$$L = .2L_1 + .8L_2$$

would be obtained by choosing first L_1 or L_2 with probabilities .2 and .8 and after this choice generating a sequence from whichever was chosen.

Except when the contrary is stated we shall assume a source to be ergodic. This assumption enables one to identify averages along a sequence with averages over the ensemble of possible sequences (the probability of a discrepancy being zero). For example the relative frequency of the letter A in a particular infinite sequence will be, with probability one, equal to its relative frequency in the ensemble of sequences.

If P_i is the probability of state i and $p_i(j)$ the transition probability to state j, then for the process to be stationary it is clear that the P_i must satisfy equilibrium conditions:

$$P_j = \sum_i P_i p_i(j).$$

In the ergodic case it can be shown that with any starting conditions the probabilities $P_j(N)$ of being in state j after N symbols, approach the equilibrium values as $N \to \infty$.

6. Choice, Uncertainty and Entropy

We have represented a discrete information source as a Markoff process. Can we define a quantity which will measure, in some sense, how much information is "produced" by such a process, or better, at what rate information is produced?

Suppose we have a set of possible events whose probabilities of occurrence are $p_1, p_2, \ldots, p_n$. These probabilities are known but that is all we know concerning which event will occur. Can we find a measure of how much "choice" is involved in the selection of the event or of how uncertain we are of the outcome?

If there is such a measure, say $H(p_1, p_2, \ldots, p_n)$, it is reasonable to require of it the following properties:

1. H should be continuous in the p_i.
2. If all the p_i are equal, $p_i = \frac{1}{n}$, then H should be a monotonic increasing function of n. With equally likely events there is more choice, or uncertainty, when there are more possible events.
3. If a choice be broken down into two successive choices, the original H should be the weighted sum of the individual values of H. The meaning of this is illustrated in Fig. 6. At the left we have three

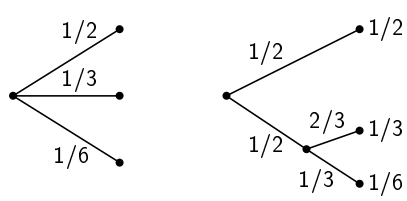


Fig. 6—Decomposition of a choice from three possibilities.

possibilities $p_1 = \frac{1}{2}$, $p_2 = \frac{1}{3}$, $p_3 = \frac{1}{6}$. On the right we first choose between two possibilities each with probability $\frac{1}{2}$, and if the second occurs make another choice with probabilities $\frac{2}{3}$, $\frac{1}{3}$. The final results have the same probabilities as before. We require, in this special case, that

$$H(\tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{6}) = H(\tfrac{1}{2}, \tfrac{1}{2}) + \tfrac{1}{2}H(\tfrac{2}{3}, \tfrac{1}{3}).$$

The coefficient $\frac{1}{2}$ is because this second choice only occurs half the time.

In Appendix 2, the following result is established:

Theorem 2: The only H satisfying the three above assumptions is of the form:

$$H = -K \sum_{i=1}^{n} p_i \log p_i$$

where K is a positive constant.

This theorem, and the assumptions required for its proof, are in no way necessary for the present theory. It is given chiefly to lend a certain plausibility to some of our later definitions. The real justification of these definitions, however, will reside in their implications.

Quantities of the form $H = -\sum p_i \log p_i$ (the constant K merely amounts to a choice of a unit of measure) play a central role in information theory as measures of information, choice and uncertainty. The form of H will be recognized as that of entropy as defined in certain formulations of statistical mechanics[8] where p_i is the probability of a system being in cell i of its phase space. H is then, for example, the H in Boltzmann's famous H theorem. We shall call $H = -\sum p_i \log p_i$ the entropy of the set of probabilities $p_1, \ldots, p_n$. If x is a chance variable we will write $H(x)$ for its entropy; thus x is not an argument of a function but a label for a number, to differentiate it from $H(y)$ say, the entropy of the chance variable y.

The entropy in the case of two possibilities with probabilities p and $q = 1 - p$, namely

$$H = -(p \log p + q \log q)$$

is plotted in Fig. 7 as a function of p.

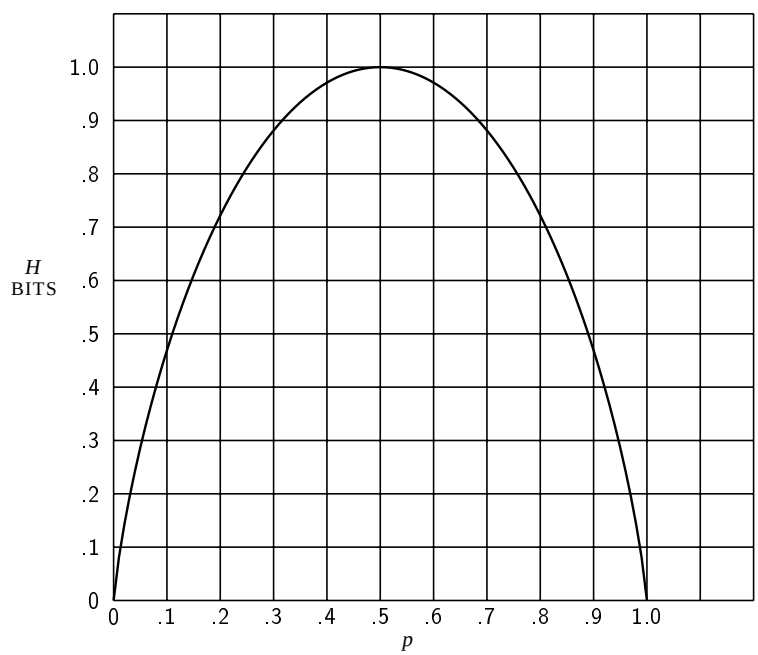


Fig. 7—Entropy in the case of two possibilities with probabilities p and $(1-p)$.

The quantity H has a number of interesting properties which further substantiate it as a reasonable measure of choice or information.

1. $H = 0$ if and only if all the p_i but one are zero, this one having the value unity. Thus only when we are certain of the outcome does H vanish. Otherwise H is positive.

2. For a given n, H is a maximum and equal to $\log n$ when all the p_i are equal (i.e., $\frac{1}{n}$). This is also intuitively the most uncertain situation.

[8] See, for example, R. C. Tolman, *Principles of Statistical Mechanics*, Oxford, Clarendon, 1938.

3. Suppose there are two events, x and y, in question with m possibilities for the first and n for the second. Let $p(i,j)$ be the probability of the joint occurrence of i for the first and j for the second. The entropy of the joint event is

$$H(x,y) = -\sum_{i,j} p(i,j) \log p(i,j)$$

while

$$H(x) = -\sum_{i,j} p(i,j) \log \sum_j p(i,j)$$
$$H(y) = -\sum_{i,j} p(i,j) \log \sum_i p(i,j).$$

It is easily shown that

$$H(x,y) \leq H(x) + H(y)$$

with equality only if the events are independent (i.e., $p(i,j) = p(i)p(j)$). The uncertainty of a joint event is less than or equal to the sum of the individual uncertainties.

4. Any change toward equalization of the probabilities $p_1, p_2, \ldots, p_n$ increases H. Thus if $p_1 < p_2$ and we increase p_1, decreasing p_2 an equal amount so that p_1 and p_2 are more nearly equal, then H increases. More generally, if we perform any "averaging" operation on the p_i of the form

$$p_i' = \sum_j a_{ij} p_j$$

where $\sum_i a_{ij} = \sum_j a_{ij} = 1$, and all $a_{ij} \geq 0$, then H increases (except in the special case where this transformation amounts to no more than a permutation of the p_j with H of course remaining the same).

5. Suppose there are two chance events x and y as in 3, not necessarily independent. For any particular value i that x can assume there is a conditional probability $p_i(j)$ that y has the value j. This is given by

$$p_i(j) = \frac{p(i,j)}{\sum_j p(i,j)}.$$

We define the *conditional entropy* of y, $H_x(y)$ as the average of the entropy of y for each value of x, weighted according to the probability of getting that particular x. That is

$$H_x(y) = -\sum_{i,j} p(i,j) \log p_i(j).$$

This quantity measures how uncertain we are of y on the average when we know x. Substituting the value of $p_i(j)$ we obtain

$$\begin{aligned} H_x(y) &= -\sum_{i,j} p(i,j) \log p(i,j) + \sum_{i,j} p(i,j) \log \sum_j p(i,j) \\ &= H(x,y) - H(x) \end{aligned}$$

or

$$H(x,y) = H(x) + H_x(y).$$

The uncertainty (or entropy) of the joint event x,y is the uncertainty of x plus the uncertainty of y when x is known.

6. From 3 and 5 we have

$$H(x) + H(y) \geq H(x,y) = H(x) + H_x(y).$$

Hence

$$H(y) \geq H_x(y).$$

The uncertainty of y is never increased by knowledge of x. It will be decreased unless x and y are independent events, in which case it is not changed.

7. The Entropy of an Information Source

Consider a discrete source of the finite state type considered above. For each possible state i there will be a set of probabilities $p_i(j)$ of producing the various possible symbols j. Thus there is an entropy H_i for each state. The entropy of the source will be defined as the average of these H_i weighted in accordance with the probability of occurrence of the states in question:

$$\begin{aligned} H &= \sum_i P_i H_i \\ &= -\sum_{i,j} P_i p_i(j) \log p_i(j) \,. \end{aligned}$$

This is the entropy of the source per symbol of text. If the Markoff process is proceeding at a definite time rate there is also an entropy per second

$$H' = \sum_i f_i H_i$$

where f_i is the average frequency (occurrences per second) of state i. Clearly

$$H' = mH$$

where m is the average number of symbols produced per second. H or H' measures the amount of information generated by the source per symbol or per second. If the logarithmic base is 2, they will represent bits per symbol or per second.

If successive symbols are independent then H is simply $-\sum p_i \log p_i$ where p_i is the probability of symbol i. Suppose in this case we consider a long message of N symbols. It will contain with high probability about $p_1 N$ occurrences of the first symbol, $p_2 N$ occurrences of the second, etc. Hence the probability of this particular message will be roughly

$$p = p_1^{p_1 N} p_2^{p_2 N} \cdots p_n^{p_n N}$$

or

$$\begin{aligned} \log p &\doteq N \sum_i p_i \log p_i \\ \log p &\doteq -NH \\ H &\doteq \frac{\log 1/p}{N}. \end{aligned}$$

H is thus approximately the logarithm of the reciprocal probability of a typical long sequence divided by the number of symbols in the sequence. The same result holds for any source. Stated more precisely we have (see Appendix 3):

Theorem 3: Given any $\epsilon > 0$ and $\delta > 0$, we can find an N_0 such that the sequences of any length $N \geq N_0$ fall into two classes:

1. *A set whose total probability is less than ϵ.*
2. *The remainder, all of whose members have probabilities satisfying the inequality*

$$\left| \frac{\log p^{-1}}{N} - H \right| < \delta.$$

In other words we are almost certain to have $\dfrac{\log p^{-1}}{N}$ very close to H when N is large.

A closely related result deals with the number of sequences of various probabilities. Consider again the sequences of length N and let them be arranged in order of decreasing probability. We define $n(q)$ to be the number we must take from this set starting with the most probable one in order to accumulate a total probability q for those taken.

Theorem 4:

$$\operatorname*{Lim}_{N\to\infty} \frac{\log n(q)}{N} = H$$

when q does not equal 0 *or* 1.

We may interpret $\log n(q)$ as the number of bits required to specify the sequence when we consider only the most probable sequences with a total probability q. Then $\frac{\log n(q)}{N}$ is the number of bits per symbol for the specification. The theorem says that for large N this will be independent of q and equal to H. The rate of growth of the logarithm of the number of reasonably probable sequences is given by H, regardless of our interpretation of "reasonably probable." Due to these results, which are proved in Appendix 3, it is possible for most purposes to treat the long sequences as though there were just 2^{HN} of them, each with a probability 2^{-HN}.

The next two theorems show that H and H' can be determined by limiting operations directly from the statistics of the message sequences, without reference to the states and transition probabilities between states.

Theorem 5: Let $p(B_i)$ be the probability of a sequence B_i of symbols from the source. Let

$$G_N = -\frac{1}{N}\sum_i p(B_i)\log p(B_i)$$

where the sum is over all sequences B_i containing N symbols. Then G_N is a monotonic decreasing function of N and

$$\operatorname*{Lim}_{N\to\infty} G_N = H.$$

Theorem 6: Let $p(B_i,S_j)$ be the probability of sequence B_i followed by symbol S_j and $p_{B_i}(S_j) = p(B_i,S_j)/p(B_i)$ be the conditional probability of S_j after B_i. Let

$$F_N = -\sum_{i,j} p(B_i,S_j)\log p_{B_i}(S_j)$$

where the sum is over all blocks B_i of $N-1$ symbols and over all symbols S_j. Then F_N is a monotonic decreasing function of N,

$$\begin{aligned} F_N &= NG_N - (N-1)G_{N-1}, \\ G_N &= \frac{1}{N}\sum_{n=1}^{N} F_n, \\ F_N &\le G_N, \end{aligned}$$

and $\operatorname{Lim}_{N\to\infty} F_N = H$.

These results are derived in Appendix 3. They show that a series of approximations to H can be obtained by considering only the statistical structure of the sequences extending over $1,2,\ldots,N$ symbols. F_N is the better approximation. In fact F_N is the entropy of the Nth order approximation to the source of the type discussed above. If there are no statistical influences extending over more than N symbols, that is if the conditional probability of the next symbol knowing the preceding $(N-1)$ is not changed by a knowledge of any before that, then $F_N = H$. F_N of course is the conditional entropy of the next symbol when the $(N-1)$ preceding ones are known, while G_N is the entropy per symbol of blocks of N symbols.

The ratio of the entropy of a source to the maximum value it could have while still restricted to the same symbols will be called its *relative entropy*. This is the maximum compression possible when we encode into the same alphabet. One minus the relative entropy is the *redundancy*. The redundancy of ordinary English, not considering statistical structure over greater distances than about eight letters, is roughly 50%. This means that when we write English half of what we write is determined by the structure of the language and half is chosen freely. The figure 50% was found by several independent methods which all gave results in

this neighborhood. One is by calculation of the entropy of the approximations to English. A second method is to delete a certain fraction of the letters from a sample of English text and then let someone attempt to restore them. If they can be restored when 50% are deleted the redundancy must be greater than 50%. A third method depends on certain known results in cryptography.

Two extremes of redundancy in English prose are represented by Basic English and by James Joyce's book "Finnegans Wake". The Basic English vocabulary is limited to 850 words and the redundancy is very high. This is reflected in the expansion that occurs when a passage is translated into Basic English. Joyce on the other hand enlarges the vocabulary and is alleged to achieve a compression of semantic content.

The redundancy of a language is related to the existence of crossword puzzles. If the redundancy is zero any sequence of letters is a reasonable text in the language and any two-dimensional array of letters forms a crossword puzzle. If the redundancy is too high the language imposes too many constraints for large crossword puzzles to be possible. A more detailed analysis shows that if we assume the constraints imposed by the language are of a rather chaotic and random nature, large crossword puzzles are just possible when the redundancy is 50%. If the redundancy is 33%, three-dimensional crossword puzzles should be possible, etc.

8. Representation of the Encoding and Decoding Operations

We have yet to represent mathematically the operations performed by the transmitter and receiver in encoding and decoding the information. Either of these will be called a discrete transducer. The input to the transducer is a sequence of input symbols and its output a sequence of output symbols. The transducer may have an internal memory so that its output depends not only on the present input symbol but also on the past history. We assume that the internal memory is finite, i.e., there exist a finite number m of possible states of the transducer and that its output is a function of the present state and the present input symbol. The next state will be a second function of these two quantities. Thus a transducer can be described by two functions:

$$y_n = f(x_n, \alpha_n)$$
$$\alpha_{n+1} = g(x_n, \alpha_n)$$

where

x_n is the nth input symbol,

α_n is the state of the transducer when the nth input symbol is introduced,

y_n is the output symbol (or sequence of output symbols) produced when x_n is introduced if the state is α_n.

If the output symbols of one transducer can be identified with the input symbols of a second, they can be connected in tandem and the result is also a transducer. If there exists a second transducer which operates on the output of the first and recovers the original input, the first transducer will be called non-singular and the second will be called its inverse.

Theorem 7: The output of a finite state transducer driven by a finite state statistical source is a finite state statistical source, with entropy (per unit time) less than or equal to that of the input. If the transducer is non-singular they are equal.

Let α represent the state of the source, which produces a sequence of symbols x_i; and let β be the state of the transducer, which produces, in its output, blocks of symbols y_j. The combined system can be represented by the "product state space" of pairs (α, β). Two points in the space (α_1, β_1) and (α_2, β_2), are connected by a line if α_1 can produce an x which changes β_1 to β_2, and this line is given the probability of that x in this case. The line is labeled with the block of y_j symbols produced by the transducer. The entropy of the output can be calculated as the weighted sum over the states. If we sum first on β each resulting term is less than or equal to the corresponding term for α, hence the entropy is not increased. If the transducer is non-singular let its output be connected to the inverse transducer. If H'_1, H'_2 and H'_3 are the output entropies of the source, the first and second transducers respectively, then $H'_1 \geq H'_2 \geq H'_3 = H'_1$ and therefore $H'_1 = H'_2$.

Suppose we have a system of constraints on possible sequences of the type which can be represented by a linear graph as in Fig. 2. If probabilities $p_{ij}^{(s)}$ were assigned to the various lines connecting state i to state j this would become a source. There is one particular assignment which maximizes the resulting entropy (see Appendix 4).

Theorem 8: Let the system of constraints considered as a channel have a capacity $C = \log W$. If we assign

$$p_{ij}^{(s)} = \frac{B_j}{B_i} W^{-\ell_{ij}^{(s)}}$$

where $\ell_{ij}^{(s)}$ is the duration of the sth symbol leading from state i to state j and the B_i satisfy

$$B_i = \sum_{s,j} B_j W^{-\ell_{ij}^{(s)}}$$

then H is maximized and equal to C.

By proper assignment of the transition probabilities the entropy of symbols on a channel can be maximized at the channel capacity.

9. The Fundamental Theorem for a Noiseless Channel

We will now justify our interpretation of H as the rate of generating information by proving that H determines the channel capacity required with most efficient coding.

Theorem 9: Let a source have entropy H (bits per symbol) and a channel have a capacity C (bits per second). Then it is possible to encode the output of the source in such a way as to transmit at the average rate $\frac{C}{H} - \epsilon$ symbols per second over the channel where ϵ is arbitrarily small. It is not possible to transmit at an average rate greater than $\frac{C}{H}$.

The converse part of the theorem, that $\frac{C}{H}$ cannot be exceeded, may be proved by noting that the entropy of the channel input per second is equal to that of the source, since the transmitter must be non-singular, and also this entropy cannot exceed the channel capacity. Hence $H' \leq C$ and the number of symbols per second $= H'/H \leq C/H$.

The first part of the theorem will be proved in two different ways. The first method is to consider the set of all sequences of N symbols produced by the source. For N large we can divide these into two groups, one containing less than $2^{(H+\eta)N}$ members and the second containing less than 2^{RN} members (where R is the logarithm of the number of different symbols) and having a total probability less than μ. As N increases η and μ approach zero. The number of signals of duration T in the channel is greater than $2^{(C-\theta)T}$ with θ small when T is large. if we choose

$$T = \left(\frac{H}{C} + \lambda\right) N$$

then there will be a sufficient number of sequences of channel symbols for the high probability group when N and T are sufficiently large (however small λ) and also some additional ones. The high probability group is coded in an arbitrary one-to-one way into this set. The remaining sequences are represented by larger sequences, starting and ending with one of the sequences not used for the high probability group. This special sequence acts as a start and stop signal for a different code. In between a sufficient time is allowed to give enough different sequences for all the low probability messages. This will require

$$T_1 = \left(\frac{R}{C} + \varphi\right) N$$

where φ is small. The mean rate of transmission in message symbols per second will then be greater than

$$\left[(1-\delta)\frac{T}{N} + \delta\frac{T_1}{N}\right]^{-1} = \left[(1-\delta)\left(\frac{H}{C} + \lambda\right) + \delta\left(\frac{R}{C} + \varphi\right)\right]^{-1}.$$

As N increases δ, λ and φ approach zero and the rate approaches $\dfrac{C}{H}$.

Another method of performing this coding and thereby proving the theorem can be described as follows: Arrange the messages of length N in order of decreasing probability and suppose their probabilities are $p_1 \geq p_2 \geq p_3 \cdots \geq p_n$. Let $P_s = \sum_1^{s-1} p_i$; that is P_s is the cumulative probability up to, but not including, p_s. We first encode into a binary system. The binary code for message s is obtained by expanding P_s as a binary number. The expansion is carried out to m_s places, where m_s is the integer satisfying:

$$\log_2 \frac{1}{p_s} \leq m_s < 1 + \log_2 \frac{1}{p_s}.$$

Thus the messages of high probability are represented by short codes and those of low probability by long codes. From these inequalities we have

$$\frac{1}{2^{m_s}} \leq p_s < \frac{1}{2^{m_s - 1}}.$$

The code for P_s will differ from all succeeding ones in one or more of its m_s places, since all the remaining P_i are at least $\frac{1}{2^{m_s}}$ larger and their binary expansions therefore differ in the first m_s places. Consequently all the codes are different and it is possible to recover the message from its code. If the channel sequences are not already sequences of binary digits, they can be ascribed binary numbers in an arbitrary fashion and the binary code thus translated into signals suitable for the channel.

The average number H' of binary digits used per symbol of original message is easily estimated. We have

$$H' = \frac{1}{N} \sum m_s p_s.$$

But,

$$\frac{1}{N} \sum \left(\log_2 \frac{1}{p_s} \right) p_s \leq \frac{1}{N} \sum m_s p_s < \frac{1}{N} \sum \left(1 + \log_2 \frac{1}{p_s} \right) p_s$$

and therefore,

$$G_N \leq H' < G_N + \frac{1}{N}$$

As N increases G_N approaches H, the entropy of the source and H' approaches H.

We see from this that the inefficiency in coding, when only a finite delay of N symbols is used, need not be greater than $\frac{1}{N}$ plus the difference between the true entropy H and the entropy G_N calculated for sequences of length N. The per cent excess time needed over the ideal is therefore less than

$$\frac{G_N}{H} + \frac{1}{HN} - 1.$$

This method of encoding is substantially the same as one found independently by R. M. Fano.[9] His method is to arrange the messages of length N in order of decreasing probability. Divide this series into two groups of as nearly equal probability as possible. If the message is in the first group its first binary digit will be 0, otherwise 1. The groups are similarly divided into subsets of nearly equal probability and the particular subset determines the second binary digit. This process is continued until each subset contains only one message. It is easily seen that apart from minor differences (generally in the last digit) this amounts to the same thing as the arithmetic process described above.

10. Discussion and Examples

In order to obtain the maximum power transfer from a generator to a load, a transformer must in general be introduced so that the generator as seen from the load has the load resistance. The situation here is roughly analogous. The transducer which does the encoding should match the source to the channel in a statistical sense. The source as seen from the channel through the transducer should have the same statistical structure

[9] Technical Report No. 65, The Research Laboratory of Electronics, M.I.T., March 17, 1949.

as the source which maximizes the entropy in the channel. The content of Theorem 9 is that, although an exact match is not in general possible, we can approximate it as closely as desired. The ratio of the actual rate of transmission to the capacity C may be called the efficiency of the coding system. This is of course equal to the ratio of the actual entropy of the channel symbols to the maximum possible entropy.

In general, ideal or nearly ideal encoding requires a long delay in the transmitter and receiver. In the noiseless case which we have been considering, the main function of this delay is to allow reasonably good matching of probabilities to corresponding lengths of sequences. With a good code the logarithm of the reciprocal probability of a long message must be proportional to the duration of the corresponding signal, in fact

$$\left|\frac{\log p^{-1}}{T} - C\right|$$

must be small for all but a small fraction of the long messages.

If a source can produce only one particular message its entropy is zero, and no channel is required. For example, a computing machine set up to calculate the successive digits of π produces a definite sequence with no chance element. No channel is required to "transmit" this to another point. One could construct a second machine to compute the same sequence at the point. However, this may be impractical. In such a case we can choose to ignore some or all of the statistical knowledge we have of the source. We might consider the digits of π to be a random sequence in that we construct a system capable of sending any sequence of digits. In a similar way we may choose to use some of our statistical knowledge of English in constructing a code, but not all of it. In such a case we consider the source with the maximum entropy subject to the statistical conditions we wish to retain. The entropy of this source determines the channel capacity which is necessary and sufficient. In the π example the only information retained is that all the digits are chosen from the set $0, 1, \ldots, 9$. In the case of English one might wish to use the statistical saving possible due to letter frequencies, but nothing else. The maximum entropy source is then the first approximation to English and its entropy determines the required channel capacity.

As a simple example of some of these results consider a source which produces a sequence of letters chosen from among A, B, C, D with probabilities $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{8}$, successive symbols being chosen independently. We have

$$\begin{aligned} H &= -\left(\tfrac{1}{2}\log\tfrac{1}{2} + \tfrac{1}{4}\log\tfrac{1}{4} + \tfrac{2}{8}\log\tfrac{1}{8}\right) \\ &= \tfrac{7}{4} \text{ bits per symbol.} \end{aligned}$$

Thus we can approximate a coding system to encode messages from this source into binary digits with an average of $\frac{7}{4}$ binary digit per symbol. In this case we can actually achieve the limiting value by the following code (obtained by the method of the second proof of Theorem 9):

A	0
B	10
C	110
D	111

The average number of binary digits used in encoding a sequence of N symbols will be

$$N\left(\tfrac{1}{2}\times 1 + \tfrac{1}{4}\times 2 + \frac{2}{8}\times 3\right) = \tfrac{7}{4}N.$$

It is easily seen that the binary digits 0, 1 have probabilities $\frac{1}{2}$, $\frac{1}{2}$ so the H for the coded sequences is one bit per symbol. Since, on the average, we have $\frac{7}{4}$ binary symbols per original letter, the entropies on a time basis are the same. The maximum possible entropy for the original set is $\log 4 = 2$, occurring when A, B, C, D have probabilities $\frac{1}{4}$, $\frac{1}{4}$, $\frac{1}{4}$, $\frac{1}{4}$. Hence the relative entropy is $\frac{7}{8}$. We can translate the binary sequences into the original set of symbols on a two-to-one basis by the following table:

00	A'
01	B'
10	C'
11	D'

This double process then encodes the original message into the same symbols but with an average compression ratio $\frac{7}{8}$.

As a second example consider a source which produces a sequence of A's and B's with probability p for A and q for B. If $p \ll q$ we have

$$\begin{aligned} H &= -\log p^p (1-p)^{1-p} \\ &= -p \log p (1-p)^{(1-p)/p} \\ &\doteq p \log \frac{e}{p}. \end{aligned}$$

In such a case one can construct a fairly good coding of the message on a 0, 1 channel by sending a special sequence, say 0000, for the infrequent symbol A and then a sequence indicating the *number* of B's following it. This could be indicated by the binary representation with all numbers containing the special sequence deleted. All numbers up to 16 are represented as usual; 16 is represented by the next binary number after 16 which does not contain four zeros, namely $17 = 10001$, etc.

It can be shown that as $p \to 0$ the coding approaches ideal provided the length of the special sequence is properly adjusted.

PART II: THE DISCRETE CHANNEL WITH NOISE

11. Representation of a Noisy Discrete Channel

We now consider the case where the signal is perturbed by noise during transmission or at one or the other of the terminals. This means that the received signal is not necessarily the same as that sent out by the transmitter. Two cases may be distinguished. If a particular transmitted signal always produces the same received signal, i.e., the received signal is a definite function of the transmitted signal, then the effect may be called distortion. If this function has an inverse — no two transmitted signals producing the same received signal — distortion may be corrected, at least in principle, by merely performing the inverse functional operation on the received signal.

The case of interest here is that in which the signal does not always undergo the same change in transmission. In this case we may assume the received signal E to be a function of the transmitted signal S and a second variable, the noise N.

$$E = f(S,N)$$

The noise is considered to be a chance variable just as the message was above. In general it may be represented by a suitable stochastic process. The most general type of noisy discrete channel we shall consider is a generalization of the finite state noise-free channel described previously. We assume a finite number of states and a set of probabilities

$$p_{\alpha,i}(\beta, j).$$

This is the probability, if the channel is in state α and symbol i is transmitted, that symbol j will be received and the channel left in state β. Thus α and β range over the possible states, i over the possible transmitted signals and j over the possible received signals. In the case where successive symbols are independently perturbed by the noise there is only one state, and the channel is described by the set of transition probabilities $p_i(j)$, the probability of transmitted symbol i being received as j.

If a noisy channel is fed by a source there are two statistical processes at work: the source and the noise. Thus there are a number of entropies that can be calculated. First there is the entropy $H(x)$ of the source or of the input to the channel (these will be equal if the transmitter is non-singular). The entropy of the output of the channel, i.e., the received signal, will be denoted by $H(y)$. In the noiseless case $H(y) = H(x)$. The joint entropy of input and output will be $H(xy)$. Finally there are two conditional entropies $H_x(y)$ and $H_y(x)$, the entropy of the output when the input is known and conversely. Among these quantities we have the relations

$$H(x,y) = H(x) + H_x(y) = H(y) + H_y(x).$$

All of these entropies can be measured on a per-second or a per-symbol basis.

12. Equivocation and Channel Capacity

If the channel is noisy it is not in general possible to reconstruct the original message or the transmitted signal with *certainty* by any operation on the received signal E. There are, however, ways of transmitting the information which are optimal in combating noise. This is the problem which we now consider.

Suppose there are two possible symbols 0 and 1, and we are transmitting at a rate of 1000 symbols per second with probabilities $p_0 = p_1 = \frac{1}{2}$. Thus our source is producing information at the rate of 1000 bits per second. During transmission the noise introduces errors so that, on the average, 1 in 100 is received incorrectly (a 0 as 1, or 1 as 0). What is the rate of transmission of information? Certainly less than 1000 bits per second since about 1% of the received symbols are incorrect. Our first impulse might be to say the rate is 990 bits per second, merely subtracting the expected number of errors. This is not satisfactory since it fails to take into account the recipient's lack of knowledge of where the errors occur. We may carry it to an extreme case and suppose the noise so great that the received symbols are entirely independent of the transmitted symbols. The probability of receiving 1 is $\frac{1}{2}$ whatever was transmitted and similarly for 0. Then about half of the received symbols are correct due to chance alone, and we would be giving the system credit for transmitting 500 bits per second while actually no information is being transmitted at all. Equally "good" transmission would be obtained by dispensing with the channel entirely and flipping a coin at the receiving point.

Evidently the proper correction to apply to the amount of information transmitted is the amount of this information which is missing in the received signal, or alternatively the uncertainty when we have received a signal of what was actually sent. From our previous discussion of entropy as a measure of uncertainty it seems reasonable to use the conditional entropy of the message, knowing the received signal, as a measure of this missing information. This is indeed the proper definition, as we shall see later. Following this idea the rate of actual transmission, R, would be obtained by subtracting from the rate of production (i.e., the entropy of the source) the average rate of conditional entropy.

$$R = H(x) - H_y(x)$$

The conditional entropy $H_y(x)$ will, for convenience, be called the equivocation. It measures the average ambiguity of the received signal.

In the example considered above, if a 0 is received the *a posteriori* probability that a 0 was transmitted is .99, and that a 1 was transmitted is .01. These figures are reversed if a 1 is received. Hence

$$\begin{aligned} H_y(x) &= -[.99 \log .99 + 0.01 \log 0.01] \\ &= .081 \text{ bits/symbol} \end{aligned}$$

or 81 bits per second. We may say that the system is transmitting at a rate $1000 - 81 = 919$ bits per second. In the extreme case where a 0 is equally likely to be received as a 0 or 1 and similarly for 1, the *a posteriori* probabilities are $\frac{1}{2}$, $\frac{1}{2}$ and

$$\begin{aligned} H_y(x) &= -\left[\tfrac{1}{2}\log\tfrac{1}{2} + \tfrac{1}{2}\log\tfrac{1}{2}\right] \\ &= 1 \text{ bit per symbol} \end{aligned}$$

or 1000 bits per second. The rate of transmission is then 0 as it should be.

The following theorem gives a direct intuitive interpretation of the equivocation and also serves to justify it as the unique appropriate measure. We consider a communication system and an observer (or auxiliary device) who can see both what is sent and what is recovered (with errors due to noise). This observer notes the errors in the recovered message and transmits data to the receiving point over a "correction channel" to enable the receiver to correct the errors. The situation is indicated schematically in Fig. 8.

Theorem 10: If the correction channel has a capacity equal to $H_y(x)$ it is possible to so encode the correction data as to send it over this channel and correct all but an arbitrarily small fraction ϵ of the errors. This is not possible if the channel capacity is less than $H_y(x)$.

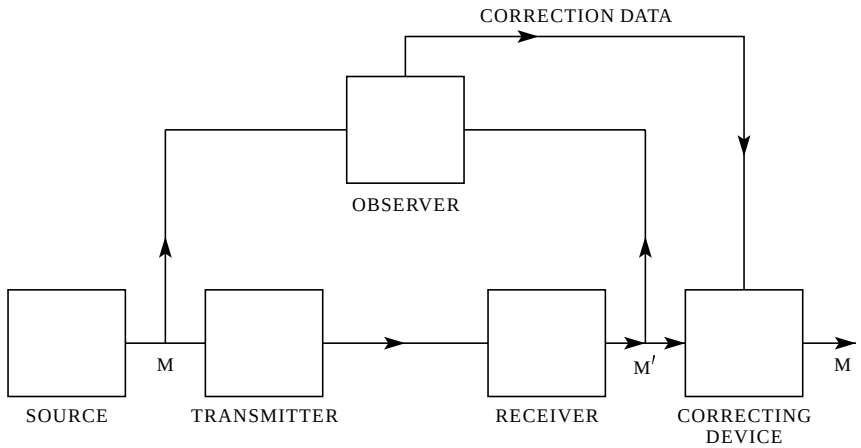


Fig. 8—Schematic diagram of a correction system.

Roughly then, $H_y(x)$ is the amount of additional information that must be supplied per second at the receiving point to correct the received message.

To prove the first part, consider long sequences of received message M' and corresponding original message M. There will be logarithmically $TH_y(x)$ of the M's which could reasonably have produced each M'. Thus we have $TH_y(x)$ binary digits to send each T seconds. This can be done with ϵ frequency of errors on a channel of capacity $H_y(x)$.

The second part can be proved by noting, first, that for any discrete chance variables x, y, z

$$H_y(x,z) \geq H_y(x).$$

The left-hand side can be expanded to give

$$\begin{aligned} H_y(z) + H_{yz}(x) &\geq H_y(x) \\ H_{yz}(x) \geq H_y(x) - H_y(z) &\geq H_y(x) - H(z). \end{aligned}$$

If we identify x as the output of the source, y as the received signal and z as the signal sent over the correction channel, then the right-hand side is the equivocation less the rate of transmission over the correction channel. If the capacity of this channel is less than the equivocation the right-hand side will be greater than zero and $H_{yz}(x) > 0$. But this is the uncertainty of what was sent, knowing both the received signal and the correction signal. If this is greater than zero the frequency of errors cannot be arbitrarily small.

Example:

> Suppose the errors occur at random in a sequence of binary digits: probability p that a digit is wrong and $q = 1 - p$ that it is right. These errors can be corrected if their position is known. Thus the correction channel need only send information as to these positions. This amounts to transmitting from a source which produces binary digits with probability p for 1 (incorrect) and q for 0 (correct). This requires a channel of capacity
>
> $$-[p \log p + q \log q]$$
>
> which is the equivocation of the original system.

The rate of transmission R can be written in two other forms due to the identities noted above. We have

$$\begin{aligned} R &= H(x) - H_y(x) \\ &= H(y) - H_x(y) \\ &= H(x) + H(y) - H(x,y). \end{aligned}$$

The first defining expression has already been interpreted as the amount of information sent less the uncertainty of what was sent. The second measures the amount received less the part of this which is due to noise. The third is the sum of the two amounts less the joint entropy and therefore in a sense is the number of bits per second common to the two. Thus all three expressions have a certain intuitive significance.

The capacity C of a noisy channel should be the maximum possible rate of transmission, i.e., the rate when the source is properly matched to the channel. We therefore define the channel capacity by

$$C = \text{Max}\big(H(x) - H_y(x)\big)$$

where the maximum is with respect to all possible information sources used as input to the channel. If the channel is noiseless, $H_y(x) = 0$. The definition is then equivalent to that already given for a noiseless channel since the maximum entropy for the channel is its capacity.

13. The Fundamental Theorem for a Discrete Channel with Noise

It may seem surprising that we should define a definite capacity C for a noisy channel since we can never send certain information in such a case. It is clear, however, that by sending the information in a redundant form the probability of errors can be reduced. For example, by repeating the message many times and by a statistical study of the different received versions of the message the probability of errors could be made very small. One would expect, however, that to make this probability of errors approach zero, the redundancy of the encoding must increase indefinitely, and the rate of transmission therefore approach zero. This is by no means true. If it were, there would not be a very well defined capacity, but only a capacity for a given frequency of errors, or a given equivocation; the capacity going down as the error requirements are made more stringent. Actually the capacity C defined above has a very definite significance. It is possible to send information at the rate C through the channel *with as small a frequency of errors or equivocation as desired* by proper encoding. This statement is not true for any rate greater than C. If an attempt is made to transmit at a higher rate than C, say $C + R_1$, then there will necessarily be an equivocation equal to or greater than the excess R_1. Nature takes payment by requiring just that much uncertainty, so that we are not actually getting any more than C through correctly.

The situation is indicated in Fig. 9. The rate of information into the channel is plotted horizontally and the equivocation vertically. Any point above the heavy line in the shaded region can be attained and those below cannot. The points on the line cannot in general be attained, but there will usually be two points on the line that can.

These results are the main justification for the definition of C and will now be proved.

Theorem 11: Let a discrete channel have the capacity C and a discrete source the entropy per second H. If $H \leq C$ there exists a coding system such that the output of the source can be transmitted over the channel with an arbitrarily small frequency of errors (or an arbitrarily small equivocation). If $H > C$ it is possible to encode the source so that the equivocation is less than $H - C + \epsilon$ where ϵ is arbitrarily small. There is no method of encoding which gives an equivocation less than $H - C$.

The method of proving the first part of this theorem is not by exhibiting a coding method having the desired properties, but by showing that such a code must exist in a certain group of codes. In fact we will

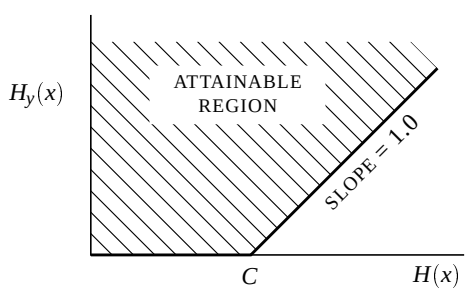


Fig. 9—The equivocation possible for a given input entropy to a channel.

average the frequency of errors over this group and show that this average can be made less than ϵ. If the average of a set of numbers is less than ϵ there must exist at least one in the set which is less than ϵ. This will establish the desired result.

The capacity C of a noisy channel has been defined as

$$C = \text{Max}\big(H(x) - H_y(x)\big)$$

where x is the input and y the output. The maximization is over all sources which might be used as input to the channel.

Let S_0 be a source which achieves the maximum capacity C. If this maximum is not actually achieved by any source let S_0 be a source which approximates to giving the maximum rate. Suppose S_0 is used as input to the channel. We consider the possible transmitted and received sequences of a long duration T. The following will be true:

1. The transmitted sequences fall into two classes, a high probability group with about $2^{TH(x)}$ members and the remaining sequences of small total probability.

2. Similarly the received sequences have a high probability set of about $2^{TH(y)}$ members and a low probability set of remaining sequences.

3. Each high probability output could be produced by about $2^{TH_y(x)}$ inputs. The probability of all other cases has a small total probability.

All the ϵ's and δ's implied by the words "small" and "about" in these statements approach zero as we allow T to increase and S_0 to approach the maximizing source.

The situation is summarized in Fig. 10 where the input sequences are points on the left and output sequences points on the right. The fan of cross lines represents the range of possible causes for a typical output.

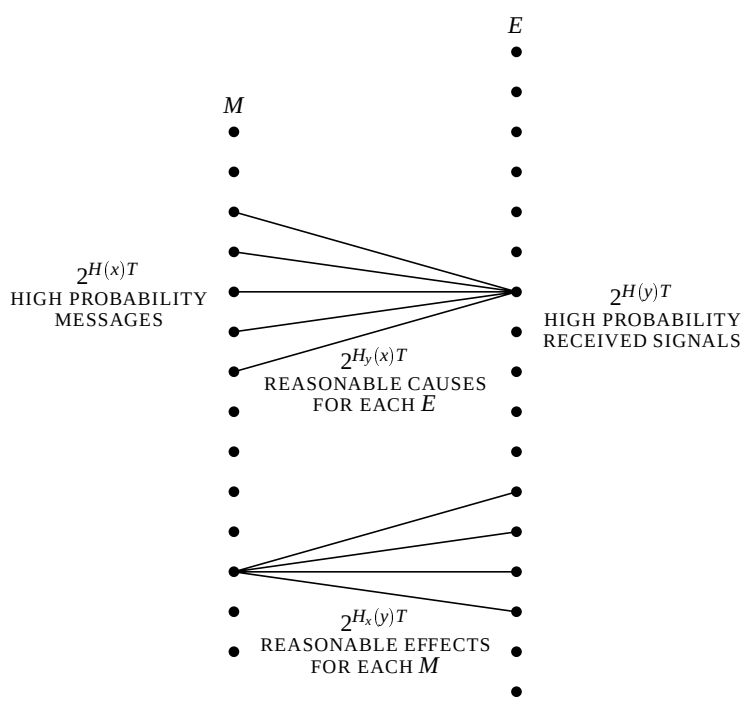


Fig. 10—Schematic representation of the relations between inputs and outputs in a channel.

Now suppose we have another source producing information at rate R with $R < C$. In the period T this source will have 2^{TR} high probability messages. We wish to associate these with a selection of the possible channel inputs in such a way as to get a small frequency of errors. We will set up this association in all

possible ways (using, however, only the high probability group of inputs as determined by the source S_0) and average the frequency of errors for this large class of possible coding systems. This is the same as calculating the frequency of errors for a random association of the messages and channel inputs of duration T. Suppose a particular output y_1 is observed. What is the probability of more than one message in the set of possible causes of y_1? There are 2^{TR} messages distributed at random in $2^{TH(x)}$ points. The probability of a particular point being a message is thus

$$2^{T(R-H(x))}.$$

The probability that none of the points in the fan is a message (apart from the actual originating message) is

$$P = \left[1 - 2^{T(R-H(x))}\right]^{2^{TH_y(x)}}.$$

Now $R < H(x) - H_y(x)$ so $R - H(x) = -H_y(x) - \eta$ with η positive. Consequently

$$P = \left[1 - 2^{-TH_y(x) - T\eta}\right]^{2^{TH_y(x)}}$$

approaches (as $T \to \infty$)

$$1 - 2^{-T\eta}.$$

Hence the probability of an error approaches zero and the first part of the theorem is proved.

The second part of the theorem is easily shown by noting that we could merely send C bits per second from the source, completely neglecting the remainder of the information generated. At the receiver the neglected part gives an equivocation $H(x) - C$ and the part transmitted need only add ϵ. This limit can also be attained in many other ways, as will be shown when we consider the continuous case.

The last statement of the theorem is a simple consequence of our definition of C. Suppose we can encode a source with $H(x) = C + a$ in such a way as to obtain an equivocation $H_y(x) = a - \epsilon$ with ϵ positive. Then $R = H(x) = C + a$ and

$$H(x) - H_y(x) = C + \epsilon$$

with ϵ positive. This contradicts the definition of C as the maximum of $H(x) - H_y(x)$.

Actually more has been proved than was stated in the theorem. If the average of a set of numbers is within ϵ of of their maximum, a fraction of at most $\sqrt{\epsilon}$ can be more than $\sqrt{\epsilon}$ below the maximum. Since ϵ is arbitrarily small we can say that almost all the systems are arbitrarily close to the ideal.

14. Discussion

The demonstration of Theorem 11, while not a pure existence proof, has some of the deficiencies of such proofs. An attempt to obtain a good approximation to ideal coding by following the method of the proof is generally impractical. In fact, apart from some rather trivial cases and certain limiting situations, no explicit description of a series of approximation to the ideal has been found. Probably this is no accident but is related to the difficulty of giving an explicit construction for a good approximation to a random sequence.

An approximation to the ideal would have the property that if the signal is altered in a reasonable way by the noise, the original can still be recovered. In other words the alteration will not in general bring it closer to another reasonable signal than the original. This is accomplished at the cost of a certain amount of redundancy in the coding. The redundancy must be introduced in the proper way to combat the particular noise structure involved. However, any redundancy in the source will usually help if it is utilized at the receiving point. In particular, if the source already has a certain redundancy and no attempt is made to eliminate it in matching to the channel, this redundancy will help combat noise. For example, in a noiseless telegraph channel one could save about 50% in time by proper encoding of the messages. This is not done and most of the redundancy of English remains in the channel symbols. This has the advantage, however, of allowing considerable noise in the channel. A sizable fraction of the letters can be received incorrectly and still reconstructed by the context. In fact this is probably not a bad approximation to the ideal in many cases, since the statistical structure of English is rather involved and the reasonable English sequences are not too far (in the sense required for the theorem) from a random selection.

As in the noiseless case a delay is generally required to approach the ideal encoding. It now has the additional function of allowing a large sample of noise to affect the signal before any judgment is made at the receiving point as to the original message. Increasing the sample size always sharpens the possible statistical assertions.

The content of Theorem 11 and its proof can be formulated in a somewhat different way which exhibits the connection with the noiseless case more clearly. Consider the possible signals of duration T and suppose a subset of them is selected to be used. Let those in the subset all be used with equal probability, and suppose the receiver is constructed to select, as the original signal, the most probable cause from the subset, when a perturbed signal is received. We define $N(T,q)$ to be the maximum number of signals we can choose for the subset such that the probability of an incorrect interpretation is less than or equal to q.

Theorem 12: $\operatorname*{Lim}_{T\to\infty} \dfrac{\log N(T,q)}{T} = C$, *where C is the channel capacity, provided that q does not equal 0 or 1.*

In other words, no matter how we set out limits of reliability, we can distinguish reliably in time T enough messages to correspond to about CT bits, when T is sufficiently large. Theorem 12 can be compared with the definition of the capacity of a noiseless channel given in Section 1.

15. Example of a Discrete Channel and its Capacity

A simple example of a discrete channel is indicated in Fig. 11. There are three possible symbols. The first is never affected by noise. The second and third each have probability p of coming through undisturbed, and q of being changed into the other of the pair. We have (letting $\alpha = -[p\log p + q\log q]$ and P and Q be the

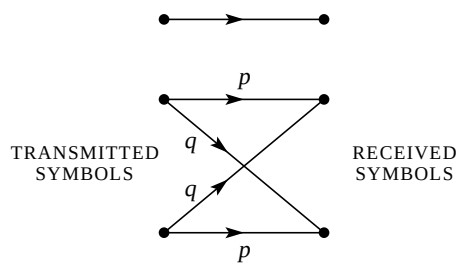


Fig. 11—Example of a discrete channel.

probabilities of using the first and second symbols)

$$\begin{aligned} H(x) &= -P\log P - 2Q\log Q \\ H_y(x) &= 2Q\alpha. \end{aligned}$$

We wish to choose P and Q in such a way as to maximize $H(x) - H_y(x)$, subject to the constraint $P + 2Q = 1$. Hence we consider

$$U = -P\log P - 2Q\log Q - 2Q\alpha + \lambda(P + 2Q)$$

$$\begin{aligned} \frac{\partial U}{\partial P} &= -1 - \log P + \lambda = 0 \\ \frac{\partial U}{\partial Q} &= -2 - 2\log Q - 2\alpha + 2\lambda = 0. \end{aligned}$$

Eliminating λ

$$\begin{aligned} \log P &= \log Q + \alpha \\ P &= Qe^{\alpha} = Q\beta \end{aligned}$$

$$P = \frac{\beta}{\beta+2} \qquad\qquad Q = \frac{1}{\beta+2}.$$

The channel capacity is then

$$C = \log \frac{\beta+2}{\beta}.$$

Note how this checks the obvious values in the cases $p = 1$ and $p = \frac{1}{2}$. In the first, $\beta = 1$ and $C = \log 3$, which is correct since the channel is then noiseless with three possible symbols. If $p = \frac{1}{2}$, $\beta = 2$ and $C = \log 2$. Here the second and third symbols cannot be distinguished at all and act together like one symbol. The first symbol is used with probability $P = \frac{1}{2}$ and the second and third together with probability $\frac{1}{2}$. This may be distributed between them in any desired way and still achieve the maximum capacity.

For intermediate values of p the channel capacity will lie between $\log 2$ and $\log 3$. The distinction between the second and third symbols conveys some information but not as much as in the noiseless case. The first symbol is used somewhat more frequently than the other two because of its freedom from noise.

16. The Channel Capacity in Certain Special Cases

If the noise affects successive channel symbols independently it can be described by a set of transition probabilities p_{ij}. This is the probability, if symbol i is sent, that j will be received. The maximum channel rate is then given by the maximum of

$$-\sum_{i,j} P_i p_{ij} \log \sum_i P_i p_{ij} + \sum_{i,j} P_i p_{ij} \log p_{ij}$$

where we vary the P_i subject to $\sum P_i = 1$. This leads by the method of Lagrange to the equations,

$$\sum_j p_{sj} \log \frac{p_{sj}}{\sum_i P_i p_{ij}} = \mu \qquad s = 1, 2, \ldots.$$

Multiplying by P_s and summing on s shows that $\mu = C$. Let the inverse of p_{sj} (if it exists) be h_{st} so that $\sum_s h_{st} p_{sj} = \delta_{tj}$. Then:

$$\sum_{s,j} h_{st} p_{sj} \log p_{sj} - \log \sum_i P_i p_{it} = C \sum_s h_{st}.$$

Hence:

$$\sum_i P_i p_{it} = \exp\Big[-C \sum_s h_{st} + \sum_{s,j} h_{st} p_{sj} \log p_{sj}\Big]$$

or,

$$P_i = \sum_t h_{it} \exp\Big[-C \sum_s h_{st} + \sum_{s,j} h_{st} p_{sj} \log p_{sj}\Big].$$

This is the system of equations for determining the maximizing values of P_i, with C to be determined so that $\sum P_i = 1$. When this is done C will be the channel capacity, and the P_i the proper probabilities for the channel symbols to achieve this capacity.

If each input symbol has the same set of probabilities on the lines emerging from it, and the same is true of each output symbol, the capacity can be easily calculated. Examples are shown in Fig. 12. In such a case $H_x(y)$ is independent of the distribution of probabilities on the input symbols, and is given by $-\sum p_i \log p_i$ where the p_i are the values of the transition probabilities from any input symbol. The channel capacity is

$$\operatorname{Max}\big[H(y) - H_x(y)\big] = \operatorname{Max} H(y) + \sum p_i \log p_i.$$

The maximum of $H(y)$ is clearly $\log m$ where m is the number of output symbols, since it is possible to make them all equally probable by making the input symbols equally probable. The channel capacity is therefore

$$C = \log m + \sum p_i \log p_i.$$

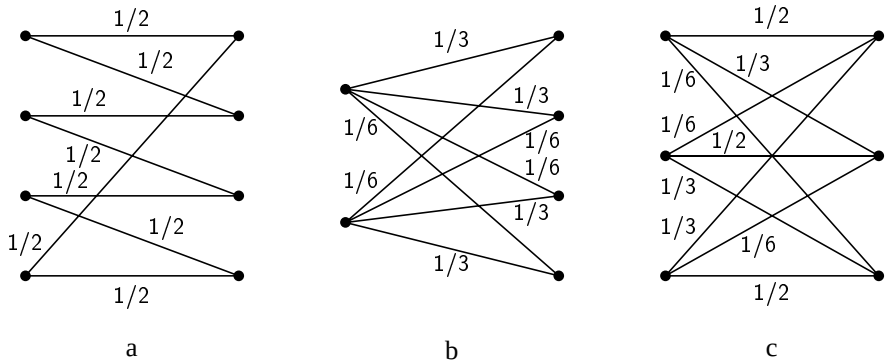


Fig. 12—Examples of discrete channels with the same transition probabilities for each input and for each output.

In Fig. 12a it would be

$$C = \log 4 - \log 2 = \log 2.$$

This could be achieved by using only the 1st and 3d symbols. In Fig. 12b

$$\begin{aligned} C &= \log 4 - \tfrac{2}{3}\log 3 - \tfrac{1}{3}\log 6 \\ &= \log 4 - \log 3 - \tfrac{1}{3}\log 2 \\ &= \log \tfrac{1}{3}2^{\frac{5}{3}}. \end{aligned}$$

In Fig. 12c we have

$$\begin{aligned} C &= \log 3 - \tfrac{1}{2}\log 2 - \tfrac{1}{3}\log 3 - \tfrac{1}{6}\log 6 \\ &= \log \frac{3}{2^{\frac{1}{2}}3^{\frac{1}{3}}6^{\frac{1}{6}}}. \end{aligned}$$

Suppose the symbols fall into several groups such that the noise never causes a symbol in one group to be mistaken for a symbol in another group. Let the capacity for the nth group be C_n (in bits per second) when we use only the symbols in this group. Then it is easily shown that, for best use of the entire set, the total probability P_n of all symbols in the nth group should be

$$P_n = \frac{2^{C_n}}{\sum 2^{C_n}}.$$

Within a group the probability is distributed just as it would be if these were the only symbols being used. The channel capacity is

$$C = \log \sum 2^{C_n}.$$

17. An Example of Efficient Coding

The following example, although somewhat unrealistic, is a case in which exact matching to a noisy channel is possible. There are two channel symbols, 0 and 1, and the noise affects them in blocks of seven symbols. A block of seven is either transmitted without error, or exactly one symbol of the seven is incorrect. These eight possibilities are equally likely. We have

$$\begin{aligned} C &= \mathrm{Max}\left[H(y) - H_x(y)\right] \\ &= \tfrac{1}{7}\left[7 + \tfrac{8}{8}\log \tfrac{1}{8}\right] \\ &= \tfrac{4}{7} \text{ bits/symbol}. \end{aligned}$$

An efficient code, allowing complete correction of errors and transmitting at the rate C, is the following (found by a method due to R. Hamming):

Let a block of seven symbols be $X_1, X_2, \ldots, X_7$. Of these X_3, X_5, X_6 and X_7 are message symbols and chosen arbitrarily by the source. The other three are redundant and calculated as follows:

X_4	is chosen to make				$\alpha = X_4 + X_5 + X_6 + X_7$	even
X_2	"	"	"	"	$\beta = X_2 + X_3 + X_6 + X_7$	"
X_1	"	"	"	"	$\gamma = X_1 + X_3 + X_5 + X_7$	"

When a block of seven is received α, β and γ are calculated and if even called zero, if odd called one. The binary number $\alpha\,\beta\gamma$ then gives the subscript of the X_i that is incorrect (if 0 there was no error).

APPENDIX 1

THE GROWTH OF THE NUMBER OF BLOCKS OF SYMBOLS WITH A FINITE STATE CONDITION

Let $N_i(L)$ be the number of blocks of symbols of length L ending in state i. Then we have

$$N_j(L) = \sum_{i,s} N_i\big(L - b_{ij}^{(s)}\big)$$

where $b_{ij}^1, b_{ij}^2, \ldots, b_{ij}^m$ are the length of the symbols which may be chosen in state i and lead to state j. These are linear difference equations and the behavior as $L \to \infty$ must be of the type

$$N_j = A_j W^L.$$

Substituting in the difference equation

$$A_j W^L = \sum_{i,s} A_i W^{L - b_{ij}^{(s)}}$$

or

$$A_j = \sum_{i,s} A_i W^{-b_{ij}^{(s)}}$$

$$\sum_i \Big(\sum_s W^{-b_{ij}^{(s)}} - \delta_{ij} \Big) A_i = 0.$$

For this to be possible the determinant

$$D(W) = |a_{ij}| = \Big| \sum_s W^{-b_{ij}^{(s)}} - \delta_{ij} \Big|$$

must vanish and this determines W, which is, of course, the largest real root of $D = 0$.

The quantity C is then given by

$$C = \operatorname*{Lim}_{L \to \infty} \frac{\log \sum A_j W^L}{L} = \log W$$

and we also note that the same growth properties result if we require that all blocks start in the same (arbitrarily chosen) state.

APPENDIX 2

DERIVATION OF $H = -\sum p_i \log p_i$

Let $H\Big(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\Big) = A(n)$. From condition (3) we can decompose a choice from s^m equally likely possibilities into a series of m choices from s equally likely possibilities and obtain

$$A(s^m) = mA(s).$$

Similarly

$$A(t^n) = nA(t).$$

We can choose n arbitrarily large and find an m to satisfy

$$s^m \leq t^n < s^{(m+1)}.$$

Thus, taking logarithms and dividing by $n \log s$,

$$\frac{m}{n} \leq \frac{\log t}{\log s} \leq \frac{m}{n} + \frac{1}{n} \quad \text{or} \quad \left| \frac{m}{n} - \frac{\log t}{\log s} \right| < \epsilon$$

where ϵ is arbitrarily small. Now from the monotonic property of $A(n)$,

$$A(s^m) \leq A(t^n) \leq A(s^{m+1})$$
$$mA(s) \leq nA(t) \leq (m+1)A(s).$$

Hence, dividing by $nA(s)$,

$$\frac{m}{n} \leq \frac{A(t)}{A(s)} \leq \frac{m}{n} + \frac{1}{n} \quad \text{or} \quad \left| \frac{m}{n} - \frac{A(t)}{A(s)} \right| < \epsilon$$

$$\left| \frac{A(t)}{A(s)} - \frac{\log t}{\log s} \right| < 2\epsilon \qquad A(t) = K \log t$$

where K must be positive to satisfy (2).

Now suppose we have a choice from n possibilities with commeasurable probabilities $p_i = \frac{n_i}{\sum n_i}$ where the n_i are integers. We can break down a choice from $\sum n_i$ possibilities into a choice from n possibilities with probabilities $p_1, \ldots, p_n$ and then, if the ith was chosen, a choice from n_i with equal probabilities. Using condition (3) again, we equate the total choice from $\sum n_i$ as computed by two methods

$$K \log \sum n_i = H(p_1, \ldots, p_n) + K \sum p_i \log n_i.$$

Hence

$$H = K \left[\sum p_i \log \sum n_i - \sum p_i \log n_i \right]$$
$$= -K \sum p_i \log \frac{n_i}{\sum n_i} = -K \sum p_i \log p_i.$$

If the p_i are incommeasurable, they may be approximated by rationals and the same expression must hold by our continuity assumption. Thus the expression holds in general. The choice of coefficient K is a matter of convenience and amounts to the choice of a unit of measure.

APPENDIX 3

THEOREMS ON ERGODIC SOURCES

If it is possible to go from any state with $P > 0$ to any other along a path of probability $p > 0$, the system is ergodic and the strong law of large numbers can be applied. Thus the number of times a given path p_{ij} in the network is traversed in a long sequence of length N is about proportional to the probability of being at i, say P_i, and then choosing this path, $P_i p_{ij} N$. If N is large enough the probability of percentage error $\pm\delta$ in this is less than ϵ so that for all but a set of small probability the actual numbers lie within the limits

$$(P_i p_{ij} \pm \delta) N.$$

Hence nearly all sequences have a probability p given by

$$p = \prod p_{ij}^{(P_i p_{ij} \pm \delta) N}$$

and $\dfrac{\log p}{N}$ is limited by

$$\frac{\log p}{N} = \sum (P_i p_{ij} \pm \delta) \log p_{ij}$$

or

$$\left| \frac{\log p}{N} - \sum P_i p_{ij} \log p_{ij} \right| < \eta.$$

This proves Theorem 3.

Theorem 4 follows immediately from this on calculating upper and lower bounds for $n(q)$ based on the possible range of values of p in Theorem 3.

In the mixed (not ergodic) case if

$$L = \sum p_i L_i$$

and the entropies of the components are $H_1 \geq H_2 \geq \cdots \geq H_n$ we have the

Theorem: $\operatorname*{Lim}_{N\to\infty} \frac{\log n(q)}{N} = \varphi(q)$ *is a decreasing step function,*

$$\varphi(q) = H_s \quad \textit{in the interval} \quad \sum_1^{s-1} \alpha_i < q < \sum_1^{s} \alpha_i.$$

To prove Theorems 5 and 6 first note that F_N is monotonic decreasing because increasing N adds a subscript to a conditional entropy. A simple substitution for $p_{B_i}(S_j)$ in the definition of F_N shows that

$$F_N = NG_N - (N-1)G_{N-1}$$

and summing this for all N gives $G_N = \dfrac{1}{N}\sum F_n$. Hence $G_N \geq F_N$ and G_N monotonic decreasing. Also they must approach the same limit. By using Theorem 3 we see that $\operatorname*{Lim}_{N\to\infty} G_N = H$.

APPENDIX 4

MAXIMIZING THE RATE FOR A SYSTEM OF CONSTRAINTS

Suppose we have a set of constraints on sequences of symbols that is of the finite state type and can be represented therefore by a linear graph. Let $\ell_{ij}^{(s)}$ be the lengths of the various symbols that can occur in passing from state i to state j. What distribution of probabilities P_i for the different states and $p_{ij}^{(s)}$ for choosing symbol s in state i and going to state j maximizes the rate of generating information under these constraints? The constraints define a discrete channel and the maximum rate must be less than or equal to the capacity C of this channel, since if all blocks of large length were equally likely, this rate would result, and if possible this would be best. We will show that this rate can be achieved by proper choice of the P_i and $p_{ij}^{(s)}$.

The rate in question is

$$\frac{-\sum P_i p_{ij}^{(s)} \log p_{ij}^{(s)}}{\sum P_i p_{ij}^{(s)} \ell_{ij}^{(s)}} = \frac{N}{M}.$$

Let $\ell_{ij} = \sum_s \ell_{ij}^{(s)}$. Evidently for a maximum $p_{ij}^{(s)} = k \exp \ell_{ij}^{(s)}$. The constraints on maximization are $\sum P_i = 1$, $\sum_j p_{ij} = 1$, $\sum P_i (p_{ij} - \delta_{ij}) = 0$. Hence we maximize

$$U = \frac{-\sum P_i p_{ij} \log p_{ij}}{\sum P_i p_{ij} \ell_{ij}} + \lambda \sum_i P_i + \sum \mu_i p_{ij} + \sum \eta_j P_i (p_{ij} - \delta_{ij})$$

$$\frac{\partial U}{\partial p_{ij}} = -\frac{M P_i (1 + \log p_{ij}) + N P_i \ell_{ij}}{M^2} + \lambda + \mu_i + \eta_i P_i = 0.$$

Solving for p_{ij}

$$p_{ij} = A_i B_j D^{-\ell_{ij}}.$$

Since

$$\sum_j p_{ij} = 1, \quad A_i^{-1} = \sum_j B_j D^{-\ell_{ij}}$$

$$p_{ij} = \frac{B_j D^{-\ell_{ij}}}{\sum_s B_s D^{-\ell_{is}}}.$$

The correct value of D is the capacity C and the B_j are solutions of

$$B_i = \sum B_j C^{-\ell_{ij}}$$

for then

$$p_{ij} = \frac{B_j}{B_i} C^{-\ell_{ij}}$$

$$\sum P_i \frac{B_j}{B_i} C^{-\ell_{ij}} = P_j$$

or

$$\sum \frac{P_i}{B_i} C^{-\ell_{ij}} = \frac{P_j}{B_j}.$$

So that if λ_i satisfy

$$\sum \gamma_i C^{-\ell_{ij}} = \gamma_j$$

$$P_i = B_i \gamma_i.$$

Both the sets of equations for B_i and γ_i can be satisfied since C is such that

$$|C^{-\ell_{ij}} - \delta_{ij}| = 0.$$

In this case the rate is

$$-\frac{\sum P_i p_{ij} \log \frac{B_j}{B_i} C^{-\ell_{ij}}}{\sum P_i p_{ij} \ell_{ij}} = C - \frac{\sum P_i p_{ij} \log \frac{B_j}{B_i}}{\sum P_i p_{ij} \ell_{ij}}$$

but

$$\sum P_i p_{ij} (\log B_j - \log B_i) = \sum_j P_j \log B_j - \sum P_i \log B_i = 0$$

Hence the rate is C and as this could never be exceeded this is the maximum, justifying the assumed solution.

PART III: MATHEMATICAL PRELIMINARIES

In this final installment of the paper we consider the case where the signals or the messages or both are continuously variable, in contrast with the discrete nature assumed heretofore. To a considerable extent the continuous case can be obtained through a limiting process from the discrete case by dividing the continuum of messages and signals into a large but finite number of small regions and calculating the various parameters involved on a discrete basis. As the size of the regions is decreased these parameters in general approach as limits the proper values for the continuous case. There are, however, a few new effects that appear and also a general change of emphasis in the direction of specialization of the general results to particular cases.

We will not attempt, in the continuous case, to obtain our results with the greatest generality, or with the extreme rigor of pure mathematics, since this would involve a great deal of abstract measure theory and would obscure the main thread of the analysis. A preliminary study, however, indicates that the theory can be formulated in a completely axiomatic and rigorous manner which includes both the continuous and discrete cases and many others. The occasional liberties taken with limiting processes in the present analysis can be justified in all cases of practical interest.

18. Sets and Ensembles of Functions

We shall have to deal in the continuous case with sets of functions and ensembles of functions. A set of functions, as the name implies, is merely a class or collection of functions, generally of one variable, time. It can be specified by giving an explicit representation of the various functions in the set, or implicitly by giving a property which functions in the set possess and others do not. Some examples are:

1. The set of functions:
$$f_\theta(t) = \sin(t+\theta).$$
Each particular value of θ determines a particular function in the set.

2. The set of all functions of time containing no frequencies over W cycles per second.

3. The set of all functions limited in band to W and in amplitude to A.

4. The set of all English speech signals as functions of time.

An *ensemble* of functions is a set of functions together with a probability measure whereby we may determine the probability of a function in the set having certain properties.[1] For example with the set,

$$f_\theta(t) = \sin(t+\theta),$$

we may give a probability distribution for θ, $P(\theta)$. The set then becomes an ensemble.

Some further examples of ensembles of functions are:

1. A finite set of functions $f_k(t)$ $(k = 1, 2, \ldots, n)$ with the probability of f_k being p_k.

2. A finite dimensional family of functions
$$f(\alpha_1, \alpha_2, \ldots, \alpha_n; t)$$
with a probability distribution on the parameters α_i:
$$p(\alpha_1, \ldots, \alpha_n).$$
For example we could consider the ensemble defined by
$$f(a_1, \ldots, a_n, \theta_1, \ldots, \theta_n; t) = \sum_{i=1}^{n} a_i \sin i(\omega t + \theta_i)$$
with the amplitudes a_i distributed normally and independently, and the phases θ_i distributed uniformly (from 0 to 2π) and independently.

[1] In mathematical terminology the functions belong to a measure space whose total measure is unity.

3. The ensemble

$$f(a_i,t) = \sum_{n=-\infty}^{+\infty} a_n \frac{\sin \pi(2Wt-n)}{\pi(2Wt-n)}$$

with the a_i normal and independent all with the same standard deviation $\sqrt{N}$. This is a representation of "white" noise, band limited to the band from 0 to W cycles per second and with average power N.[2]

4. Let points be distributed on the t axis according to a Poisson distribution. At each selected point the function $f(t)$ is placed and the different functions added, giving the ensemble

$$\sum_{k=-\infty}^{\infty} f(t+t_k)$$

where the t_k are the points of the Poisson distribution. This ensemble can be considered as a type of impulse or shot noise where all the impulses are identical.

5. The set of English speech functions with the probability measure given by the frequency of occurrence in ordinary use.

An ensemble of functions $f_\alpha(t)$ is *stationary* if the same ensemble results when all functions are shifted any fixed amount in time. The ensemble

$$f_\theta(t) = \sin(t+\theta)$$

is stationary if θ is distributed uniformly from 0 to 2π. If we shift each function by t_1 we obtain

$$\begin{aligned} f_\theta(t+t_1) &= \sin(t+t_1+\theta) \\ &= \sin(t+\varphi) \end{aligned}$$

with φ distributed uniformly from 0 to 2π. Each function has changed but the ensemble as a whole is invariant under the translation. The other examples given above are also stationary.

An ensemble is *ergodic* if it is stationary, and there is no subset of the functions in the set with a probability different from 0 and 1 which is stationary. The ensemble

$$\sin(t+\theta)$$

is ergodic. No subset of these functions of probability $\neq 0,1$ is transformed into itself under all time translations. On the other hand the ensemble

$$a\sin(t+\theta)$$

with a distributed normally and θ uniform is stationary but not ergodic. The subset of these functions with a between 0 and 1 for example is stationary.

Of the examples given, 3 and 4 are ergodic, and 5 may perhaps be considered so. If an ensemble is ergodic we may say roughly that each function in the set is typical of the ensemble. More precisely it is known that with an ergodic ensemble an average of any statistic over the ensemble is equal (with probability 1) to an average over the time translations of a particular function of the set.[3] Roughly speaking, each function can be expected, as time progresses, to go through, with the proper frequency, all the convolutions of any of the functions in the set.

[2]This representation can be used as a definition of band limited white noise. It has certain advantages in that it involves fewer limiting operations than do definitions that have been used in the past. The name "white noise," already firmly entrenched in the literature, is perhaps somewhat unfortunate. In optics white light means either any continuous spectrum as contrasted with a point spectrum, or a spectrum which is flat with *wavelength* (which is not the same as a spectrum flat with frequency).

[3]This is the famous ergodic theorem or rather one aspect of this theorem which was proved in somewhat different formulations by Birkoff, von Neumann, and Koopman, and subsequently generalized by Wiener, Hopf, Hurewicz and others. The literature on ergodic theory is quite extensive and the reader is referred to the papers of these writers for precise and general formulations; e.g., E. Hopf, "Ergodentheorie," *Ergebnisse der Mathematik und ihrer Grenzgebiete,* v. 5; "On Causality Statistics and Probability," *Journal of Mathematics and Physics,* v. XIII, No. 1, 1934; N. Wiener, "The Ergodic Theorem," *Duke Mathematical Journal,* v. 5, 1939.

Just as we may perform various operations on numbers or functions to obtain new numbers or functions, we can perform operations on ensembles to obtain new ensembles. Suppose, for example, we have an ensemble of functions $f_\alpha(t)$ and an operator T which gives for each function $f_\alpha(t)$ a resulting function $g_\alpha(t)$:

$$g_\alpha(t) = T f_\alpha(t).$$

Probability measure is defined for the set $g_\alpha(t)$ by means of that for the set $f_\alpha(t)$. The probability of a certain subset of the $g_\alpha(t)$ functions is equal to that of the subset of the $f_\alpha(t)$ functions which produce members of the given subset of g functions under the operation T. Physically this corresponds to passing the ensemble through some device, for example, a filter, a rectifier or a modulator. The output functions of the device form the ensemble $g_\alpha(t)$.

A device or operator T will be called invariant if shifting the input merely shifts the output, i.e., if

$$g_\alpha(t) = T f_\alpha(t)$$

implies

$$g_\alpha(t+t_1) = T f_\alpha(t+t_1)$$

for all $f_\alpha(t)$ and all t_1. It is easily shown (see Appendix 5 that if T is invariant and the input ensemble is stationary then the output ensemble is stationary. Likewise if the input is ergodic the output will also be ergodic.

A filter or a rectifier is invariant under all time translations. The operation of modulation is not since the carrier phase gives a certain time structure. However, modulation is invariant under all translations which are multiples of the period of the carrier.

Wiener has pointed out the intimate relation between the invariance of physical devices under time translations and Fourier theory.[4] He has shown, in fact, that if a device is linear as well as invariant Fourier analysis is then the appropriate mathematical tool for dealing with the problem.

An ensemble of functions is the appropriate mathematical representation of the messages produced by a continuous source (for example, speech), of the signals produced by a transmitter, and of the perturbing noise. Communication theory is properly concerned, as has been emphasized by Wiener, not with operations on particular functions, but with operations on ensembles of functions. A communication system is designed not for a particular speech function and still less for a sine wave, but for the ensemble of speech functions.

19. Band Limited Ensembles of Functions

If a function of time $f(t)$ is limited to the band from 0 to W cycles per second it is completely determined by giving its ordinates at a series of discrete points spaced $\frac{1}{2W}$ seconds apart in the manner indicated by the following result.[5]

Theorem 13: Let $f(t)$ contain no frequencies over W. Then

$$f(t) = \sum_{-\infty}^{\infty} X_n \frac{\sin \pi(2Wt-n)}{\pi(2Wt-n)}$$

where

$$X_n = f\left(\frac{n}{2W}\right).$$

[4]Communication theory is heavily indebted to Wiener for much of its basic philosophy and theory. His classic NDRC report, *The Interpolation, Extrapolation and Smoothing of Stationary Time Series* (Wiley, 1949), contains the first clear-cut formulation of communication theory as a statistical problem, the study of operations on time series. This work, although chiefly concerned with the linear prediction and filtering problem, is an important collateral reference in connection with the present paper. We may also refer here to Wiener's *Cybernetics* (Wiley, 1948), dealing with the general problems of communication and control.

[5]For a proof of this theorem and further discussion see the author's paper "Communication in the Presence of Noise" published in the *Proceedings of the Institute of Radio Engineers*, v. 37, No. 1, Jan., 1949, pp. 10–21.

In this expansion $f(t)$ is represented as a sum of orthogonal functions. The coefficients X_n of the various terms can be considered as coordinates in an infinite dimensional "function space." In this space each function corresponds to precisely one point and each point to one function.

A function can be considered to be substantially limited to a time T if all the ordinates X_n outside this interval of time are zero. In this case all but $2TW$ of the coordinates will be zero. Thus functions limited to a band W and duration T correspond to points in a space of $2TW$ dimensions.

A subset of the functions of band W and duration T corresponds to a region in this space. For example, the functions whose total energy is less than or equal to E correspond to points in a $2TW$ dimensional sphere with radius $r = \sqrt{2WE}$.

An *ensemble* of functions of limited duration and band will be represented by a probability distribution $p(x_1, \ldots, x_n)$ in the corresponding n dimensional space. If the ensemble is not limited in time we can consider the $2TW$ coordinates in a given interval T to represent substantially the part of the function in the interval T and the probability distribution $p(x_1, \ldots, x_n)$ to give the statistical structure of the ensemble for intervals of that duration.

20. Entropy of a Continuous Distribution

The entropy of a discrete set of probabilities $p_1, \ldots, p_n$ has been defined as:

$$H = -\sum p_i \log p_i.$$

In an analogous manner we define the entropy of a continuous distribution with the density distribution function $p(x)$ by:

$$H = -\int_{-\infty}^{\infty} p(x) \log p(x)\, dx.$$

With an n dimensional distribution $p(x_1, \ldots, x_n)$ we have

$$H = -\int \cdots \int p(x_1, \ldots, x_n) \log p(x_1, \ldots, x_n)\, dx_1 \cdots dx_n.$$

If we have two arguments x and y (which may themselves be multidimensional) the joint and conditional entropies of $p(x,y)$ are given by

$$H(x,y) = -\iint p(x,y) \log p(x,y)\, dx\, dy$$

and

$$H_x(y) = -\iint p(x,y) \log \frac{p(x,y)}{p(x)}\, dx\, dy$$
$$H_y(x) = -\iint p(x,y) \log \frac{p(x,y)}{p(y)}\, dx\, dy$$

where

$$p(x) = \int p(x,y)\, dy$$
$$p(y) = \int p(x,y)\, dx.$$

The entropies of continuous distributions have most (but not all) of the properties of the discrete case. In particular we have the following:

1. If x is limited to a certain volume v in its space, then $H(x)$ is a maximum and equal to $\log v$ when $p(x)$ is constant $(1/v)$ in the volume.

2. With any two variables x, y we have

$$H(x,y) \leq H(x) + H(y)$$

with equality if (and only if) x and y are independent, i.e., $p(x,y) = p(x)p(y)$ (apart possibly from a set of points of probability zero).

3. Consider a generalized averaging operation of the following type:

$$p'(y) = \int a(x,y)p(x)\,dx$$

with

$$\int a(x,y)\,dx = \int a(x,y)\,dy = 1, \qquad a(x,y) \geq 0.$$

Then the entropy of the averaged distribution $p'(y)$ is equal to or greater than that of the original distribution $p(x)$.

4. We have

$$H(x,y) = H(x) + H_x(y) = H(y) + H_y(x)$$

and

$$H_x(y) \leq H(y).$$

5. Let $p(x)$ be a one-dimensional distribution. The form of $p(x)$ giving a maximum entropy subject to the condition that the standard deviation of x be fixed at σ is Gaussian. To show this we must maximize

$$H(x) = -\int p(x)\log p(x)\,dx$$

with

$$\sigma^2 = \int p(x)x^2\,dx \quad \text{and} \quad 1 = \int p(x)\,dx$$

as constraints. This requires, by the calculus of variations, maximizing

$$\int \left[-p(x)\log p(x) + \lambda p(x)x^2 + \mu p(x)\right]dx.$$

The condition for this is

$$-1 - \log p(x) + \lambda x^2 + \mu = 0$$

and consequently (adjusting the constants to satisfy the constraints)

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma}e^{-(x^2/2\sigma^2)}.$$

Similarly in n dimensions, suppose the second order moments of $p(x_1,\ldots,x_n)$ are fixed at A_{ij}:

$$A_{ij} = \int\cdots\int x_i x_j p(x_1,\ldots,x_n)\,dx_1\cdots dx_n.$$

Then the maximum entropy occurs (by a similar calculation) when $p(x_1,\ldots,x_n)$ is the n dimensional Gaussian distribution with the second order moments A_{ij}.

6. The entropy of a one-dimensional Gaussian distribution whose standard deviation is σ is given by

$$H(x) = \log\sqrt{2\pi e}\sigma.$$

This is calculated as follows:

$$\begin{aligned}
p(x) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-(x^2/2\sigma^2)} \\
-\log p(x) &= \log\sqrt{2\pi}\sigma + \frac{x^2}{2\sigma^2} \\
H(x) &= -\int p(x)\log p(x)\,dx \\
&= \int p(x)\log\sqrt{2\pi}\sigma\,dx + \int p(x)\frac{x^2}{2\sigma^2}\,dx \\
&= \log\sqrt{2\pi}\sigma + \frac{\sigma^2}{2\sigma^2} \\
&= \log\sqrt{2\pi}\sigma + \log\sqrt{e} \\
&= \log\sqrt{2\pi e}\sigma.
\end{aligned}$$

Similarly the n dimensional Gaussian distribution with associated quadratic form a_{ij} is given by

$$p(x_1,\ldots,x_n) = \frac{|a_{ij}|^{\frac{1}{2}}}{(2\pi)^{n/2}} \exp\Big(-\tfrac{1}{2}\sum a_{ij}x_ix_j\Big)$$

and the entropy can be calculated as

$$H = \log(2\pi e)^{n/2}|a_{ij}|^{-\frac{1}{2}}$$

where $|a_{ij}|$ is the determinant whose elements are a_{ij}.

7. If x is limited to a half line ($p(x) = 0$ for $x \leq 0$) and the first moment of x is fixed at a:

$$a = \int_0^\infty p(x)x\,dx,$$

then the maximum entropy occurs when

$$p(x) = \frac{1}{a}e^{-(x/a)}$$

and is equal to $\log ea$.

8. There is one important difference between the continuous and discrete entropies. In the discrete case the entropy measures in an *absolute* way the randomness of the chance variable. In the continuous case the measurement is *relative to the coordinate system*. If we change coordinates the entropy will in general change. In fact if we change to coordinates $y_1\cdots y_n$ the new entropy is given by

$$H(y) = \int\cdots\int p(x_1,\ldots,x_n)J\left(\frac{x}{y}\right)\log p(x_1,\ldots,x_n)J\left(\frac{x}{y}\right)dy_1\cdots dy_n$$

where $J\left(\frac{x}{y}\right)$ is the Jacobian of the coordinate transformation. On expanding the logarithm and changing the variables to $x_1\cdots x_n$, we obtain:

$$H(y) = H(x) - \int\cdots\int p(x_1,\ldots,x_n)\log J\left(\frac{x}{y}\right)dx_1\ldots dx_n.$$

Thus the new entropy is the old entropy less the expected logarithm of the Jacobian. In the continuous case the entropy can be considered a measure of randomness *relative to an assumed standard*, namely the coordinate system chosen with each small volume element $dx_1 \cdots dx_n$ given equal weight. When we change the coordinate system the entropy in the new system measures the randomness when equal volume elements $dy_1 \cdots dy_n$ in the new system are given equal weight.

In spite of this dependence on the coordinate system the entropy concept is as important in the continuous case as the discrete case. This is due to the fact that the derived concepts of information rate and channel capacity depend on the *difference* of two entropies and this difference *does not* depend on the coordinate frame, each of the two terms being changed by the same amount.

The entropy of a continuous distribution can be negative. The scale of measurements sets an arbitrary zero corresponding to a uniform distribution over a unit volume. A distribution which is more confined than this has less entropy and will be negative. The rates and capacities will, however, always be non-negative.

9. A particular case of changing coordinates is the linear transformation

$$y_j = \sum_i a_{ij} x_i.$$

In this case the Jacobian is simply the determinant $|a_{ij}|^{-1}$ and

$$H(y) = H(x) + \log |a_{ij}|.$$

In the case of a rotation of coordinates (or any measure preserving transformation) $J = 1$ and $H(y) = H(x)$.

21. Entropy of an Ensemble of Functions

Consider an ergodic ensemble of functions limited to a certain band of width W cycles per second. Let

$$p(x_1, \ldots, x_n)$$

be the density distribution function for amplitudes $x_1, \ldots, x_n$ at n successive sample points. We define the entropy of the ensemble per degree of freedom by

$$H' = -\operatorname*{Lim}_{n\to\infty} \frac{1}{n} \int \cdots \int p(x_1, \ldots, x_n) \log p(x_1, \ldots, x_n)\, dx_1 \ldots dx_n.$$

We may also define an entropy H per second by dividing, not by n, but by the time T in seconds for n samples. Since $n = 2TW$, $H = 2WH'$.

With white thermal noise p is Gaussian and we have

$$\begin{aligned} H' &= \log \sqrt{2\pi e N}, \\ H &= W \log 2\pi e N. \end{aligned}$$

For a given average power N, white noise has the maximum possible entropy. This follows from the maximizing properties of the Gaussian distribution noted above.

The entropy for a continuous stochastic process has many properties analogous to that for discrete processes. In the discrete case the entropy was related to the logarithm of the *probability* of long sequences, and to the *number* of reasonably probable sequences of long length. In the continuous case it is related in a similar fashion to the logarithm of the *probability density* for a long series of samples, and the *volume* of reasonably high probability in the function space.

More precisely, if we assume $p(x_1, \ldots, x_n)$ continuous in all the x_i for all n, then for sufficiently large n

$$\left| \frac{\log p}{n} - H' \right| < \epsilon$$

for all choices of $(x_1, \ldots, x_n)$ apart from a set whose total probability is less than δ, with δ and ϵ arbitrarily small. This follows form the ergodic property if we divide the space into a large number of small cells.

The relation of H to volume can be stated as follows: Under the same assumptions consider the n dimensional space corresponding to $p(x_1, \ldots, x_n)$. Let $V_n(q)$ be the smallest volume in this space which includes in its interior a total probability q. Then

$$\operatorname*{Lim}_{n\to\infty} \frac{\log V_n(q)}{n} = H'$$

provided q does not equal 0 or 1.

These results show that for large n there is a rather well-defined volume (at least in the logarithmic sense) of high probability, and that within this volume the probability density is relatively uniform (again in the logarithmic sense).

In the white noise case the distribution function is given by

$$p(x_1, \ldots, x_n) = \frac{1}{(2\pi N)^{n/2}} \exp - \frac{1}{2N} \sum x_i^2.$$

Since this depends only on $\sum x_i^2$ the surfaces of equal probability density are spheres and the entire distribution has spherical symmetry. The region of high probability is a sphere of radius $\sqrt{nN}$. As $n \to \infty$ the probability of being outside a sphere of radius $\sqrt{n(N+\epsilon)}$ approaches zero and $\frac{1}{n}$ times the logarithm of the volume of the sphere approaches $\log \sqrt{2\pi e N}$.

In the continuous case it is convenient to work not with the entropy H of an ensemble but with a derived quantity which we will call the entropy power. This is defined as the power in a white noise limited to the same band as the original ensemble and having the same entropy. In other words if H' is the entropy of an ensemble its entropy power is

$$N_1 = \frac{1}{2\pi e} \exp 2H'.$$

In the geometrical picture this amounts to measuring the high probability volume by the squared radius of a sphere having the same volume. Since white noise has the maximum entropy for a given power, the entropy power of any noise is less than or equal to its actual power.

22. Entropy Loss in Linear Filters

Theorem 14: If an ensemble having an entropy H_1 per degree of freedom in band W is passed through a filter with characteristic $Y(f)$ the output ensemble has an entropy

$$H_2 = H_1 + \frac{1}{W} \int_W \log |Y(f)|^2 \, df.$$

The operation of the filter is essentially a linear transformation of coordinates. If we think of the different frequency components as the original coordinate system, the new frequency components are merely the old ones multiplied by factors. The coordinate transformation matrix is thus essentially diagonalized in terms of these coordinates. The Jacobian of the transformation is (for n sine and n cosine components)

$$J = \prod_{i=1}^{n} |Y(f_i)|^2$$

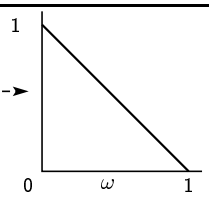
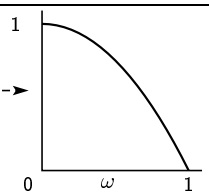
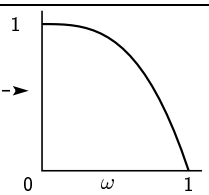
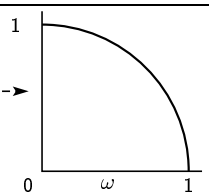
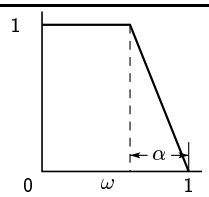
where the f_i are equally spaced through the band W. This becomes in the limit

$$\exp \frac{1}{W} \int_W \log |Y(f)|^2 \, df.$$

Since J is constant its average value is the same quantity and applying the theorem on the change of entropy with a change of coordinates, the result follows. We may also phrase it in terms of the entropy power. Thus if the entropy power of the first ensemble is N_1 that of the second is

$$N_1 \exp \frac{1}{W} \int_W \log |Y(f)|^2 \, df.$$

TABLE I

GAIN	ENTROPY POWER FACTOR	ENTROPY POWER GAIN IN DECIBELS	IMPULSE RESPONSE
$1-\omega$ 	$\frac{1}{e^2}$	-8.69	$\frac{\sin^2(t/2)}{t^2/2}$
$1-\omega^2$ 	$\left(\frac{2}{e}\right)^4$	-5.33	$2\left[\frac{\sin t}{t^3}-\frac{\cos t}{t^2}\right]$
$1-\omega^3$ 	0.411	-3.87	$6\left[\frac{\cos t-1}{t^4}-\frac{\cos t}{2t^2}+\frac{\sin t}{t^3}\right]$
$\sqrt{1-\omega^2}$ 	$\left(\frac{2}{e}\right)^2$	-2.67	$\frac{\pi}{2}\frac{J_1(t)}{t}$
	$\frac{1}{e^{2\alpha}}$	-8.69α	$\frac{1}{\alpha t^2}\left[\cos(1-\alpha)t-\cos t\right]$

The final entropy power is the initial entropy power multiplied by the geometric mean gain of the filter. If the gain is measured in *db*, then the output entropy power will be increased by the arithmetic mean *db* gain over W.

In Table I the entropy power loss has been calculated (and also expressed in *db*) for a number of ideal gain characteristics. The impulsive responses of these filters are also given for $W = 2\pi$, with phase assumed to be 0.

The entropy loss for many other cases can be obtained from these results. For example the entropy power factor $1/e^2$ for the first case also applies to any gain characteristic obtain from $1-\omega$ by a measure preserving transformation of the ω axis. In particular a linearly increasing gain $G(\omega) = \omega$, or a "saw tooth" characteristic between 0 and 1 have the same entropy loss. The reciprocal gain has the reciprocal factor. Thus $1/\omega$ has the factor e^2. Raising the gain to any power raises the factor to this power.

23. Entropy of a Sum of Two Ensembles

If we have two ensembles of functions $f_\alpha(t)$ and $g_\beta(t)$ we can form a new ensemble by "addition." Suppose the first ensemble has the probability density function $p(x_1,\ldots,x_n)$ and the second $q(x_1,\ldots,x_n)$. Then the

density function for the sum is given by the convolution:

$$r(x_1,\ldots,x_n) = \int\cdots\int p(y_1,\ldots,y_n)q(x_1-y_1,\ldots,x_n-y_n)\,dy_1\cdots dy_n.$$

Physically this corresponds to adding the noises or signals represented by the original ensembles of functions.

The following result is derived in Appendix 6.

Theorem 15: Let the average power of two ensembles be N_1 and N_2 and let their entropy powers be $\overline{N}_1$ and $\overline{N}_2$. Then the entropy power of the sum, $\overline{N}_3$, is bounded by

$$\overline{N}_1+\overline{N}_2 \le \overline{N}_3 \le N_1+N_2.$$

White Gaussian noise has the peculiar property that it can absorb any other noise or signal ensemble which may be added to it with a resultant entropy power approximately equal to the sum of the white noise power and the signal power (measured from the average signal value, which is normally zero), provided the signal power is small, in a certain sense, compared to noise.

Consider the function space associated with these ensembles having n dimensions. The white noise corresponds to the spherical Gaussian distribution in this space. The signal ensemble corresponds to another probability distribution, not necessarily Gaussian or spherical. Let the second moments of this distribution about its center of gravity be a_{ij}. That is, if $p(x_1,\ldots,x_n)$ is the density distribution function

$$a_{ij} = \int\cdots\int p(x_i-\alpha_i)(x_j-\alpha_j)\,dx_1\cdots dx_n$$

where the α_i are the coordinates of the center of gravity. Now a_{ij} is a positive definite quadratic form, and we can rotate our coordinate system to align it with the principal directions of this form. a_{ij} is then reduced to diagonal form b_{ii}. We require that each b_{ii} be small compared to N, the squared radius of the spherical distribution.

In this case the convolution of the noise and signal produce approximately a Gaussian distribution whose corresponding quadratic form is

$$N+b_{ii}.$$

The entropy power of this distribution is

$$\left[\prod(N+b_{ii})\right]^{1/n}$$

or approximately

$$= \left[(N)^n + \sum b_{ii}(N)^{n-1}\right]^{1/n}$$
$$\doteq N + \frac{1}{n}\sum b_{ii}.$$

The last term is the signal power, while the first is the noise power.

PART IV: THE CONTINUOUS CHANNEL

24. The Capacity of a Continuous Channel

In a continuous channel the input or transmitted signals will be continuous functions of time $f(t)$ belonging to a certain set, and the output or received signals will be perturbed versions of these. We will consider only the case where both transmitted and received signals are limited to a certain band W. They can then be specified, for a time T, by $2TW$ numbers, and their statistical structure by finite dimensional distribution functions. Thus the statistics of the transmitted signal will be determined by

$$P(x_1,\ldots,x_n) = P(x)$$

and those of the noise by the conditional probability distribution

$$P_{x_1,\ldots,x_n}(y_1,\ldots,y_n) = P_x(y).$$

The rate of transmission of information for a continuous channel is defined in a way analogous to that for a discrete channel, namely

$$R = H(x) - H_y(x)$$

where $H(x)$ is the entropy of the input and $H_y(x)$ the equivocation. The channel capacity C is defined as the maximum of R when we vary the input over all possible ensembles. This means that in a finite dimensional approximation we must vary $P(x) = P(x_1,\ldots,x_n)$ and maximize

$$-\int P(x)\log P(x)\,dx + \iint P(x,y)\log\frac{P(x,y)}{P(y)}\,dx\,dy.$$

This can be written

$$\iint P(x,y)\log\frac{P(x,y)}{P(x)P(y)}\,dx\,dy$$

using the fact that $\iint P(x,y)\log P(x)\,dx\,dy = \int P(x)\log P(x)\,dx$. The channel capacity is thus expressed as follows:

$$C = \operatorname*{Lim}_{T\to\infty}\operatorname*{Max}_{P(x)}\frac{1}{T}\iint P(x,y)\log\frac{P(x,y)}{P(x)P(y)}\,dx\,dy.$$

It is obvious in this form that R and C are independent of the coordinate system since the numerator and denominator in $\log\frac{P(x,y)}{P(x)P(y)}$ will be multiplied by the same factors when x and y are transformed in any one-to-one way. This integral expression for C is more general than $H(x) - H_y(x)$. Properly interpreted (see Appendix 7) it will always exist while $H(x) - H_y(x)$ may assume an indeterminate form $\infty - \infty$ in some cases. This occurs, for example, if x is limited to a surface of fewer dimensions than n in its n dimensional approximation.

If the logarithmic base used in computing $H(x)$ and $H_y(x)$ is two then C is the maximum number of binary digits that can be sent per second over the channel with arbitrarily small equivocation, just as in the discrete case. This can be seen physically by dividing the space of signals into a large number of small cells, sufficiently small so that the probability density $P_x(y)$ of signal x being perturbed to point y is substantially constant over a cell (either of x or y). If the cells are considered as distinct points the situation is essentially the same as a discrete channel and the proofs used there will apply. But it is clear physically that this quantizing of the volume into individual points cannot in any practical situation alter the final answer significantly, provided the regions are sufficiently small. Thus the capacity will be the limit of the capacities for the discrete subdivisions and this is just the continuous capacity defined above.

On the mathematical side it can be shown first (see Appendix 7) that if u is the message, x is the signal, y is the received signal (perturbed by noise) and v is the recovered message then

$$H(x) - H_y(x) \geq H(u) - H_v(u)$$

regardless of what operations are performed on u to obtain x or on y to obtain v. Thus no matter how we encode the binary digits to obtain the signal, or how we decode the received signal to recover the message, the discrete rate for the binary digits does not exceed the channel capacity we have defined. On the other hand, it is possible under very general conditions to find a coding system for transmitting binary digits at the rate C with as small an equivocation or frequency of errors as desired. This is true, for example, if, when we take a finite dimensional approximating space for the signal functions, $P(x,y)$ is continuous in both x and y except at a set of points of probability zero.

An important special case occurs when the noise is added to the signal and is independent of it (in the probability sense). Then $P_x(y)$ is a function only of the difference $n = (y - x)$,

$$P_x(y) = Q(y - x)$$

and we can assign a definite entropy to the noise (independent of the statistics of the signal), namely the entropy of the distribution $Q(n)$. This entropy will be denoted by $H(n)$.

Theorem 16: If the signal and noise are independent and the received signal is the sum of the transmitted signal and the noise then the rate of transmission is

$$R = H(y) - H(n),$$

i.e., the entropy of the received signal less the entropy of the noise. The channel capacity is

$$C = \operatorname*{Max}_{P(x)} H(y) - H(n).$$

We have, since $y = x + n$:

$$H(x,y) = H(x,n).$$

Expanding the left side and using the fact that x and n are independent

$$H(y) + H_y(x) = H(x) + H(n).$$

Hence

$$R = H(x) - H_y(x) = H(y) - H(n).$$

Since $H(n)$ is independent of $P(x)$, maximizing R requires maximizing $H(y)$, the entropy of the received signal. If there are certain constraints on the ensemble of transmitted signals, the entropy of the received signal must be maximized subject to these constraints.

25. Channel Capacity with an Average Power Limitation

A simple application of Theorem 16 is the case when the noise is a white thermal noise and the transmitted signals are limited to a certain average power P. Then the received signals have an average power $P+N$ where N is the average noise power. The maximum entropy for the received signals occurs when they also form a white noise ensemble since this is the greatest possible entropy for a power $P+N$ and can be obtained by a suitable choice of transmitted signals, namely if they form a white noise ensemble of power P. The entropy (per second) of the received ensemble is then

$$H(y) = W \log 2\pi e(P+N),$$

and the noise entropy is

$$H(n) = W \log 2\pi e N.$$

The channel capacity is

$$C = H(y) - H(n) = W \log \frac{P+N}{N}.$$

Summarizing we have the following:

Theorem 17: The capacity of a channel of band W perturbed by white thermal noise power N when the average transmitter power is limited to P is given by

$$C = W \log \frac{P+N}{N}.$$

This means that by sufficiently involved encoding systems we can transmit binary digits at the rate $W \log_2 \frac{P+N}{N}$ bits per second, with arbitrarily small frequency of errors. It is not possible to transmit at a higher rate by any encoding system without a definite positive frequency of errors.

To approximate this limiting rate of transmission the transmitted signals must approximate, in statistical properties, a white noise.[6] A system which approaches the ideal rate may be described as follows: Let

[6]This and other properties of the white noise case are discussed from the geometrical point of view in "Communication in the Presence of Noise," *loc. cit.*

$M = 2^s$ samples of white noise be constructed each of duration T. These are assigned binary numbers from 0 to $M-1$. At the transmitter the message sequences are broken up into groups of s and for each group the corresponding noise sample is transmitted as the signal. At the receiver the M samples are known and the actual received signal (perturbed by noise) is compared with each of them. The sample which has the least R.M.S. discrepancy from the received signal is chosen as the transmitted signal and the corresponding binary number reconstructed. This process amounts to choosing the most probable (*a posteriori*) signal. The number M of noise samples used will depend on the tolerable frequency ϵ of errors, but for almost all selections of samples we have

$$\operatorname*{Lim}_{\epsilon\to 0}\operatorname*{Lim}_{T\to\infty}\frac{\log M(\epsilon,T)}{T} = W\log\frac{P+N}{N},$$

so that no matter how small ϵ is chosen, we can, by taking T sufficiently large, transmit as near as we wish to $TW\log\dfrac{P+N}{N}$ binary digits in the time T.

Formulas similar to $C = W\log\dfrac{P+N}{N}$ for the white noise case have been developed independently by several other writers, although with somewhat different interpretations. We may mention the work of N. Wiener,[7] W. G. Tuller,[8] and H. Sullivan in this connection.

In the case of an arbitrary perturbing noise (not necessarily white thermal noise) it does not appear that the maximizing problem involved in determining the channel capacity C can be solved explicitly. However, upper and lower bounds can be set for C in terms of the average noise power N the noise entropy power N_1. These bounds are sufficiently close together in most practical cases to furnish a satisfactory solution to the problem.

Theorem 18: The capacity of a channel of band W perturbed by an arbitrary noise is bounded by the inequalities

$$W\log\frac{P+N_1}{N_1} \le C \le W\log\frac{P+N}{N_1}$$

where

$$\begin{aligned} P &= \textit{average transmitter power}\\ N &= \textit{average noise power}\\ N_1 &= \textit{entropy power of the noise.}\end{aligned}$$

Here again the average power of the perturbed signals will be $P+N$. The maximum entropy for this power would occur if the received signal were white noise and would be $W\log 2\pi e(P+N)$. It may not be possible to achieve this; i.e., there may not be any ensemble of transmitted signals which, added to the perturbing noise, produce a white thermal noise at the receiver, but at least this sets an upper bound to $H(y)$. We have, therefore

$$\begin{aligned} C &= \operatorname{Max} H(y) - H(n)\\ &\le W\log 2\pi e(P+N) - W\log 2\pi e N_1.\end{aligned}$$

This is the upper limit given in the theorem. The lower limit can be obtained by considering the rate if we make the transmitted signal a white noise, of power P. In this case the entropy power of the received signal must be at least as great as that of a white noise of power $P+N_1$ since we have shown in in a previous theorem that the entropy power of the sum of two ensembles is greater than or equal to the sum of the individual entropy powers. Hence

$$\operatorname{Max} H(y) \ge W\log 2\pi e(P+N_1)$$

[7] *Cybernetics, loc. cit.*

[8] "Theoretical Limitations on the Rate of Transmission of Information," *Proceedings of the Institute of Radio Engineers*, v. 37, No. 5, May, 1949, pp. 468–78.

and

$$\begin{aligned} C &\geq W \log 2\pi e(P+N_1) - W \log 2\pi e N_1 \\ &= W \log \frac{P+N_1}{N_1}. \end{aligned}$$

As P increases, the upper and lower bounds approach each other, so we have as an asymptotic rate

$$W \log \frac{P+N}{N_1}.$$

If the noise is itself white, $N = N_1$ and the result reduces to the formula proved previously:

$$C = W \log\left(1 + \frac{P}{N}\right).$$

If the noise is Gaussian but with a spectrum which is not necessarily flat, N_1 is the geometric mean of the noise power over the various frequencies in the band W. Thus

$$N_1 = \exp \frac{1}{W} \int_W \log N(f)\, df$$

where $N(f)$ is the noise power at frequency f.

Theorem 19: If we set the capacity for a given transmitter power P equal to

$$C = W \log \frac{P+N-\eta}{N_1}$$

then η is monotonic decreasing as P increases and approaches 0 as a limit.

Suppose that for a given power P_1 the channel capacity is

$$W \log \frac{P_1+N-\eta_1}{N_1}.$$

This means that the best signal distribution, say $p(x)$, when added to the noise distribution $q(x)$, gives a received distribution $r(y)$ whose entropy power is $(P_1+N-\eta_1)$. Let us increase the power to $P_1+\Delta P$ by adding a white noise of power ΔP to the signal. The entropy of the received signal is now at least

$$H(y) = W \log 2\pi e(P_1+N-\eta_1+\Delta P)$$

by application of the theorem on the minimum entropy power of a sum. Hence, since we can attain the H indicated, the entropy of the maximizing distribution must be at least as great and η must be monotonic decreasing. To show that $\eta \to 0$ as $P \to \infty$ consider a signal which is white noise with a large P. Whatever the perturbing noise, the received signal will be approximately a white noise, if P is sufficiently large, in the sense of having an entropy power approaching $P+N$.

26. The Channel Capacity with a Peak Power Limitation

In some applications the transmitter is limited not by the average power output but by the peak instantaneous power. The problem of calculating the channel capacity is then that of maximizing (by variation of the ensemble of transmitted symbols)

$$H(y) - H(n)$$

subject to the constraint that all the functions $f(t)$ in the ensemble be less than or equal to $\sqrt{S}$, say, for all t. A constraint of this type does not work out as well mathematically as the average power limitation. The most we have obtained for this case is a lower bound valid for all $\frac{S}{N}$, an "asymptotic" upper bound (valid for large $\frac{S}{N}$) and an asymptotic value of C for $\frac{S}{N}$ small.

Theorem 20: The channel capacity C for a band W perturbed by white thermal noise of power N is bounded by

$$C \geq W \log \frac{2}{\pi e^3} \frac{S}{N},$$

where S is the peak allowed transmitter power. For sufficiently large $\frac{S}{N}$

$$C \leq W \log \frac{\frac{2}{\pi e} S + N}{N} (1+\epsilon)$$

where ϵ is arbitrarily small. As $\frac{S}{N} \to 0$ (and provided the band W starts at 0)

$$C \Big/ W \log \left(1 + \frac{S}{N}\right) \to 1.$$

We wish to maximize the entropy of the received signal. If $\frac{S}{N}$ is large this will occur very nearly when we maximize the entropy of the transmitted ensemble.

The asymptotic upper bound is obtained by relaxing the conditions on the ensemble. Let us suppose that the power is limited to S not at every instant of time, but only at the sample points. The maximum entropy of the transmitted ensemble under these weakened conditions is certainly greater than or equal to that under the original conditions. This altered problem can be solved easily. The maximum entropy occurs if the different samples are independent and have a distribution function which is constant from $-\sqrt{S}$ to $+\sqrt{S}$. The entropy can be calculated as

$$W \log 4S.$$

The received signal will then have an entropy less than

$$W \log(4S + 2\pi e N)(1+\epsilon)$$

with $\epsilon \to 0$ as $\frac{S}{N} \to \infty$ and the channel capacity is obtained by subtracting the entropy of the white noise, $W \log 2\pi e N$:

$$W \log(4S + 2\pi e N)(1+\epsilon) - W \log(2\pi e N) = W \log \frac{\frac{2}{\pi e} S + N}{N} (1+\epsilon).$$

This is the desired upper bound to the channel capacity.

To obtain a lower bound consider the same ensemble of functions. Let these functions be passed through an ideal filter with a triangular transfer characteristic. The gain is to be unity at frequency 0 and decline linearly down to gain 0 at frequency W. We first show that the output functions of the filter have a peak power limitation S at all times (not just the sample points). First we note that a pulse $\frac{\sin 2\pi W t}{2\pi W t}$ going into the filter produces

$$\frac{1}{2} \frac{\sin^2 \pi W t}{(\pi W t)^2}$$

in the output. This function is never negative. The input function (in the general case) can be thought of as the sum of a series of shifted functions

$$a \frac{\sin 2\pi W t}{2\pi W t}$$

where a, the amplitude of the sample, is not greater than $\sqrt{S}$. Hence the output is the sum of shifted functions of the non-negative form above with the same coefficients. These functions being non-negative, the greatest positive value for any t is obtained when all the coefficients a have their maximum positive values, i.e., $\sqrt{S}$. In this case the input function was a constant of amplitude $\sqrt{S}$ and since the filter has unit gain for D.C., the output is the same. Hence the output ensemble has a peak power S.

The entropy of the output ensemble can be calculated from that of the input ensemble by using the theorem dealing with such a situation. The output entropy is equal to the input entropy plus the geometrical mean gain of the filter:

$$\int_0^W \log G^2\, df = \int_0^W \log\Big(\frac{W-f}{W}\Big)^2 df = -2W.$$

Hence the output entropy is

$$W \log 4S - 2W = W \log \frac{4S}{e^2}$$

and the channel capacity is greater than

$$W \log \frac{2}{\pi e^3} \frac{S}{N}.$$

We now wish to show that, for small $\frac{S}{N}$ (peak signal power over average white noise power), the channel capacity is approximately

$$C = W \log\left(1 + \frac{S}{N}\right).$$

More precisely $C \Big/ W \log\left(1+\frac{S}{N}\right) \to 1$ as $\frac{S}{N} \to 0$. Since the average signal power P is less than or equal to the peak S, it follows that for all $\frac{S}{N}$

$$C \le W \log\left(1+\frac{P}{N}\right) \le W \log\left(1+\frac{S}{N}\right).$$

Therefore, if we can find an ensemble of functions such that they correspond to a rate nearly $W \log\left(1+\frac{S}{N}\right)$ and are limited to band W and peak S the result will be proved. Consider the ensemble of functions of the following type. A series of t samples have the same value, either $+\sqrt{S}$ or $-\sqrt{S}$, then the next t samples have the same value, etc. The value for a series is chosen at random, probability $\frac{1}{2}$ for $+\sqrt{S}$ and $\frac{1}{2}$ for $-\sqrt{S}$. If this ensemble be passed through a filter with triangular gain characteristic (unit gain at D.C.), the output is peak limited to $\pm S$. Furthermore the average power is nearly S and can be made to approach this by taking t sufficiently large. The entropy of the sum of this and the thermal noise can be found by applying the theorem on the sum of a noise and a small signal. This theorem will apply if

$$\sqrt{t}\frac{S}{N}$$

is sufficiently small. This can be ensured by taking $\frac{S}{N}$ small enough (after t is chosen). The entropy power will be $S+N$ to as close an approximation as desired, and hence the rate of transmission as near as we wish to

$$W \log\left(\frac{S+N}{N}\right).$$

PART V: THE RATE FOR A CONTINUOUS SOURCE

27. Fidelity Evaluation Functions

In the case of a discrete source of information we were able to determine a definite rate of generating information, namely the entropy of the underlying stochastic process. With a continuous source the situation is considerably more involved. In the first place a continuously variable quantity can assume an infinite number of values and requires, therefore, an infinite number of binary digits for exact specification. This means that to transmit the output of a continuous source with *exact recovery* at the receiving point requires,

in general, a channel of infinite capacity (in bits per second). Since, ordinarily, channels have a certain amount of noise, and therefore a finite capacity, exact transmission is impossible.

This, however, evades the real issue. Practically, we are not interested in exact transmission when we have a continuous source, but only in transmission to within a certain tolerance. The question is, can we assign a definite rate to a continuous source when we require only a certain fidelity of recovery, measured in a suitable way. Of course, as the fidelity requirements are increased the rate will increase. It will be shown that we can, in very general cases, define such a rate, having the property that it is possible, by properly encoding the information, to transmit it over a channel whose capacity is equal to the rate in question, and satisfy the fidelity requirements. A channel of smaller capacity is insufficient.

It is first necessary to give a general mathematical formulation of the idea of fidelity of transmission. Consider the set of messages of a long duration, say T seconds. The source is described by giving the probability density, in the associated space, that the source will select the message in question $P(x)$. A given communication system is described (from the external point of view) by giving the conditional probability $P_x(y)$ that if message x is produced by the source the recovered message at the receiving point will be y. The system as a whole (including source and transmission system) is described by the probability function $P(x,y)$ of having message x and final output y. If this function is known, the complete characteristics of the system from the point of view of fidelity are known. Any evaluation of fidelity must correspond mathematically to an operation applied to $P(x,y)$. This operation must at least have the properties of a simple ordering of systems; i.e., it must be possible to say of two systems represented by $P_1(x,y)$ and $P_2(x,y)$ that, according to our fidelity criterion, either (1) the first has higher fidelity, (2) the second has higher fidelity, or (3) they have equal fidelity. This means that a criterion of fidelity can be represented by a numerically valued function:

$$v\big(P(x,y)\big)$$

whose argument ranges over possible probability functions $P(x,y)$.

We will now show that under very general and reasonable assumptions the function $v\big(P(x,y)\big)$ can be written in a seemingly much more specialized form, namely as an average of a function $\rho(x,y)$ over the set of possible values of x and y:

$$v\big(P(x,y)\big) = \iint P(x,y)\rho(x,y)\,dx\,dy.$$

To obtain this we need only assume (1) that the source and system are ergodic so that a very long sample will be, with probability nearly 1, typical of the ensemble, and (2) that the evaluation is "reasonable" in the sense that it is possible, by observing a typical input and output x_1 and y_1, to form a tentative evaluation on the basis of these samples; and if these samples are increased in duration the tentative evaluation will, with probability 1, approach the exact evaluation based on a full knowledge of $P(x,y)$. Let the tentative evaluation be $\rho(x,y)$. Then the function $\rho(x,y)$ approaches (as $T \to \infty$) a constant for almost all (x,y) which are in the high probability region corresponding to the system:

$$\rho(x,y) \to v\big(P(x,y)\big)$$

and we may also write

$$\rho(x,y) \to \iint P(x,y)\rho(x,y)\,dx\,dy$$

since

$$\iint P(x,y)\,dx\,dy = 1.$$

This establishes the desired result.

The function $\rho(x,y)$ has the general nature of a "distance" between x and y.[9] It measures how undesirable it is (according to our fidelity criterion) to receive y when x is transmitted. The general result given above can be restated as follows: Any reasonable evaluation can be represented as an average of a distance function over the set of messages and recovered messages x and y weighted according to the probability $P(x,y)$ of getting the pair in question, provided the duration T of the messages be taken sufficiently large.

The following are simple examples of evaluation functions:

[9] It is not a "metric" in the strict sense, however, since in general it does not satisfy either $\rho(x,y) = \rho(y,x)$ or $\rho(x,y) + \rho(y,z) \geq \rho(x,z)$.

1. R.M.S. criterion.
$$v = \overline{\big(x(t) - y(t)\big)^2}.$$
In this very commonly used measure of fidelity the distance function $\rho(x,y)$ is (apart from a constant factor) the square of the ordinary Euclidean distance between the points x and y in the associated function space.
$$\rho(x,y) = \frac{1}{T}\int_0^T \big[x(t) - y(t)\big]^2\,dt.$$

2. Frequency weighted R.M.S. criterion. More generally one can apply different weights to the different frequency components before using an R.M.S. measure of fidelity. This is equivalent to passing the difference $x(t) - y(t)$ through a shaping filter and then determining the average power in the output. Thus let
$$e(t) = x(t) - y(t)$$
and
$$f(t) = \int_{-\infty}^{\infty} e(\tau)k(t-\tau)\,d\tau$$
then
$$\rho(x,y) = \frac{1}{T}\int_0^T f(t)^2\,dt.$$

3. Absolute error criterion.
$$\rho(x,y) = \frac{1}{T}\int_0^T \big|x(t) - y(t)\big|\,dt.$$

4. The structure of the ear and brain determine implicitly an evaluation, or rather a number of evaluations, appropriate in the case of speech or music transmission. There is, for example, an "intelligibility" criterion in which $\rho(x,y)$ is equal to the relative frequency of incorrectly interpreted words when message $x(t)$ is received as $y(t)$. Although we cannot give an explicit representation of $\rho(x,y)$ in these cases it could, in principle, be determined by sufficient experimentation. Some of its properties follow from well-known experimental results in hearing, e.g., the ear is relatively insensitive to phase and the sensitivity to amplitude and frequency is roughly logarithmic.

5. The discrete case can be considered as a specialization in which we have tacitly assumed an evaluation based on the frequency of errors. The function $\rho(x,y)$ is then defined as the number of symbols in the sequence y differing from the corresponding symbols in x divided by the total number of symbols in x.

28. The Rate for a Source Relative to a Fidelity Evaluation

We are now in a position to define a rate of generating information for a continuous source. We are given $P(x)$ for the source and an evaluation v determined by a distance function $\rho(x,y)$ which will be assumed continuous in both x and y. With a particular system $P(x,y)$ the quality is measured by
$$v = \iint \rho(x,y)P(x,y)\,dx\,dy.$$
Furthermore the rate of flow of binary digits corresponding to $P(x,y)$ is
$$R = \iint P(x,y)\log\frac{P(x,y)}{P(x)P(y)}\,dx\,dy.$$
We define the rate R_1 of generating information for a given quality v_1 of reproduction to be the minimum of R when we keep v fixed at v_1 and vary $P_x(y)$. That is:
$$R_1 = \operatorname*{Min}_{P_x(y)} \iint P(x,y)\log\frac{P(x,y)}{P(x)P(y)}\,dx\,dy$$

subject to the constraint:

$$v_1 = \iint P(x,y)\rho(x,y)\,dx\,dy.$$

This means that we consider, in effect, all the communication systems that might be used and that transmit with the required fidelity. The rate of transmission in bits per second is calculated for each one and we choose that having the least rate. This latter rate is the rate we assign the source for the fidelity in question.

The justification of this definition lies in the following result:

Theorem 21: If a source has a rate R_1 for a valuation v_1 it is possible to encode the output of the source and transmit it over a channel of capacity C with fidelity as near v_1 as desired provided $R_1 \leq C$. This is not possible if $R_1 > C$.

The last statement in the theorem follows immediately from the definition of R_1 and previous results. If it were not true we could transmit more than C bits per second over a channel of capacity C. The first part of the theorem is proved by a method analogous to that used for Theorem 11. We may, in the first place, divide the (x,y) space into a large number of small cells and represent the situation as a discrete case. This will not change the evaluation function by more than an arbitrarily small amount (when the cells are very small) because of the continuity assumed for $\rho(x,y)$. Suppose that $P_1(x,y)$ is the particular system which minimizes the rate and gives R_1. We choose from the high probability y's a set at random containing

$$2^{(R_1+\epsilon)T}$$

members where $\epsilon \to 0$ as $T \to \infty$. With large T each chosen point will be connected by a high probability line (as in Fig. 10) to a set of x's. A calculation similar to that used in proving Theorem 11 shows that with large T almost all x's are covered by the fans from the chosen y points for almost all choices of the y's. The communication system to be used operates as follows: The selected points are assigned binary numbers. When a message x is originated it will (with probability approaching 1 as $T \to \infty$) lie within at least one of the fans. The corresponding binary number is transmitted (or one of them chosen arbitrarily if there are several) over the channel by suitable coding means to give a small probability of error. Since $R_1 \leq C$ this is possible. At the receiving point the corresponding y is reconstructed and used as the recovered message.

The evaluation v_1' for this system can be made arbitrarily close to v_1 by taking T sufficiently large. This is due to the fact that for each long sample of message $x(t)$ and recovered message $y(t)$ the evaluation approaches v_1 (with probability 1).

It is interesting to note that, in this system, the noise in the recovered message is actually produced by a kind of general quantizing at the transmitter and not produced by the noise in the channel. It is more or less analogous to the quantizing noise in PCM.

29. The Calculation of Rates

The definition of the rate is similar in many respects to the definition of channel capacity. In the former

$$R = \operatorname*{Min}_{P_x(y)} \iint P(x,y)\log\frac{P(x,y)}{P(x)P(y)}\,dx\,dy$$

with $P(x)$ and $v_1 = \iint P(x,y)\rho(x,y)\,dx\,dy$ fixed. In the latter

$$C = \operatorname*{Max}_{P(x)} \iint P(x,y)\log\frac{P(x,y)}{P(x)P(y)}\,dx\,dy$$

with $P_x(y)$ fixed and possibly one or more other constraints (e.g., an average power limitation) of the form $K = \iint P(x,y)\lambda(x,y)\,dx\,dy$.

A partial solution of the general maximizing problem for determining the rate of a source can be given. Using Lagrange's method we consider

$$\iint \left[P(x,y)\log\frac{P(x,y)}{P(x)P(y)} + \mu P(x,y)\rho(x,y) + \nu(x)P(x,y)\right]dx\,dy.$$

The variational equation (when we take the first variation on $P(x,y)$) leads to

$$P_y(x) = B(x)e^{-\lambda\rho(x,y)}$$

where λ is determined to give the required fidelity and $B(x)$ is chosen to satisfy

$$\int B(x)e^{-\lambda\rho(x,y)}\,dx = 1.$$

This shows that, with best encoding, the conditional probability of a certain cause for various received y, $P_y(x)$ will decline exponentially with the distance function $\rho(x,y)$ between the x and y in question.

In the special case where the distance function $\rho(x,y)$ depends only on the (vector) difference between x and y,

$$\rho(x,y) = \rho(x-y)$$

we have

$$\int B(x)e^{-\lambda\rho(x-y)}\,dx = 1.$$

Hence $B(x)$ is constant, say α, and

$$P_y(x) = \alpha e^{-\lambda\rho(x-y)}.$$

Unfortunately these formal solutions are difficult to evaluate in particular cases and seem to be of little value. In fact, the actual calculation of rates has been carried out in only a few very simple cases.

If the distance function $\rho(x,y)$ is the mean square discrepancy between x and y and the message ensemble is white noise, the rate can be determined. In that case we have

$$R = \mathrm{Min}\big[H(x) - H_y(x)\big] = H(x) - \mathrm{Max}\,H_y(x)$$

with $N = \overline{(x-y)^2}$. But the $\mathrm{Max}\,H_y(x)$ occurs when $y-x$ is a white noise, and is equal to $W_1 \log 2\pi eN$ where W_1 is the bandwidth of the message ensemble. Therefore

$$\begin{aligned} R &= W_1 \log 2\pi eQ - W_1 \log 2\pi eN \\ &= W_1 \log \frac{Q}{N} \end{aligned}$$

where Q is the average message power. This proves the following:

Theorem 22: The rate for a white noise source of power Q and band W_1 relative to an R.M.S. measure of fidelity is

$$R = W_1 \log \frac{Q}{N}$$

where N is the allowed mean square error between original and recovered messages.

More generally with any message source we can obtain inequalities bounding the rate relative to a mean square error criterion.

Theorem 23: The rate for any source of band W_1 is bounded by

$$W_1 \log \frac{Q_1}{N} \le R \le W_1 \log \frac{Q}{N}$$

where Q is the average power of the source, Q_1 its entropy power and N the allowed mean square error.

The lower bound follows from the fact that the $\mathrm{Max}\,H_y(x)$ for a given $\overline{(x-y)^2} = N$ occurs in the white noise case. The upper bound results if we place points (used in the proof of Theorem 21) not in the best way but at random in a sphere of radius $\sqrt{Q-N}$.

ACKNOWLEDGMENTS

The writer is indebted to his colleagues at the Laboratories, particularly to Dr. H. W. Bode, Dr. J. R. Pierce, Dr. B. McMillan, and Dr. B. M. Oliver for many helpful suggestions and criticisms during the course of this work. Credit should also be given to Professor N. Wiener, whose elegant solution of the problems of filtering and prediction of stationary ensembles has considerably influenced the writer's thinking in this field.

APPENDIX 5

Let S_1 be any measurable subset of the g ensemble, and S_2 the subset of the f ensemble which gives S_1 under the operation T. Then

$$S_1 = TS_2.$$

Let H^λ be the operator which shifts all functions in a set by the time λ. Then

$$H^\lambda S_1 = H^\lambda T S_2 = T H^\lambda S_2$$

since T is invariant and therefore commutes with H^λ. Hence if $m[S]$ is the probability measure of the set S

$$\begin{aligned} m[H^\lambda S_1] &= m[T H^\lambda S_2] = m[H^\lambda S_2] \\ &= m[S_2] = m[S_1] \end{aligned}$$

where the second equality is by definition of measure in the g space, the third since the f ensemble is stationary, and the last by definition of g measure again.

To prove that the ergodic property is preserved under invariant operations, let S_1 be a subset of the g ensemble which is invariant under H^λ, and let S_2 be the set of all functions f which transform into S_1. Then

$$H^\lambda S_1 = H^\lambda T S_2 = T H^\lambda S_2 = S_1$$

so that $H^\lambda S_2$ is included in S_2 for all λ. Now, since

$$m[H^\lambda S_2] = m[S_1]$$

this implies

$$H^\lambda S_2 = S_2$$

for all λ with $m[S_2] \neq 0, 1$. This contradiction shows that S_1 does not exist.

APPENDIX 6

The upper bound, $\overline{N}_3 \leq N_1 + N_2$, is due to the fact that the maximum possible entropy for a power $N_1 + N_2$ occurs when we have a white noise of this power. In this case the entropy power is $N_1 + N_2$.

To obtain the lower bound, suppose we have two distributions in n dimensions $p(x_i)$ and $q(x_i)$ with entropy powers $\overline{N}_1$ and $\overline{N}_2$. What form should p and q have to minimize the entropy power $\overline{N}_3$ of their convolution $r(x_i)$:

$$r(x_i) = \int p(y_i) q(x_i - y_i)\, dy_i.$$

The entropy H_3 of r is given by

$$H_3 = -\int r(x_i) \log r(x_i)\, dx_i.$$

We wish to minimize this subject to the constraints

$$H_1 = -\int p(x_i) \log p(x_i)\, dx_i$$

$$H_2 = -\int q(x_i) \log q(x_i)\, dx_i.$$

We consider then

$$U = -\int \left[r(x)\log r(x) + \lambda p(x)\log p(x) + \mu q(x)\log q(x) \right] dx$$
$$\delta U = -\int \left[[1+\log r(x)]\delta r(x) + \lambda[1+\log p(x)]\delta p(x) + \mu[1+\log q(x)]\delta q(x) \right] dx.$$

If $p(x)$ is varied at a particular argument $x_i = s_i$, the variation in $r(x)$ is

$$\delta r(x) = q(x_i - s_i)$$

and

$$\delta U = -\int q(x_i - s_i)\log r(x_i)\, dx_i - \lambda \log p(s_i) = 0$$

and similarly when q is varied. Hence the conditions for a minimum are

$$\int q(x_i - s_i)\log r(x_i)\, dx_i = -\lambda \log p(s_i)$$
$$\int p(x_i - s_i)\log r(x_i)\, dx_i = -\mu \log q(s_i).$$

If we multiply the first by $p(s_i)$ and the second by $q(s_i)$ and integrate with respect to s_i we obtain

$$H_3 = -\lambda H_1$$
$$H_3 = -\mu H_2$$

or solving for λ and μ and replacing in the equations

$$H_1 \int q(x_i - s_i)\log r(x_i)\, dx_i = -H_3 \log p(s_i)$$
$$H_2 \int p(x_i - s_i)\log r(x_i)\, dx_i = -H_3 \log q(s_i).$$

Now suppose $p(x_i)$ and $q(x_i)$ are normal

$$p(x_i) = \frac{|A_{ij}|^{n/2}}{(2\pi)^{n/2}} \exp -\tfrac{1}{2}\sum A_{ij}x_i x_j$$
$$q(x_i) = \frac{|B_{ij}|^{n/2}}{(2\pi)^{n/2}} \exp -\tfrac{1}{2}\sum B_{ij}x_i x_j.$$

Then $r(x_i)$ will also be normal with quadratic form C_{ij}. If the inverses of these forms are a_{ij}, b_{ij}, c_{ij} then

$$c_{ij} = a_{ij} + b_{ij}.$$

We wish to show that these functions satisfy the minimizing conditions if and only if $a_{ij} = Kb_{ij}$ and thus give the minimum H_3 under the constraints. First we have

$$\log r(x_i) = \frac{n}{2}\log\frac{1}{2\pi}|C_{ij}| - \tfrac{1}{2}\sum C_{ij}x_i x_j$$
$$\int q(x_i - s_i)\log r(x_i)\, dx_i = \frac{n}{2}\log\frac{1}{2\pi}|C_{ij}| - \tfrac{1}{2}\sum C_{ij}s_i s_j - \tfrac{1}{2}\sum C_{ij}b_{ij}.$$

This should equal

$$\frac{H_3}{H_1}\left[\frac{n}{2}\log\frac{1}{2\pi}|A_{ij}| - \tfrac{1}{2}\sum A_{ij}s_i s_j\right]$$

which requires $A_{ij} = \dfrac{H_1}{H_3}C_{ij}$. In this case $A_{ij} = \dfrac{H_1}{H_2}B_{ij}$ and both equations reduce to identities.

APPENDIX 7

The following will indicate a more general and more rigorous approach to the central definitions of communication theory. Consider a probability measure space whose elements are ordered pairs (x,y). The variables x, y are to be identified as the possible transmitted and received signals of some long duration T. Let us call the set of all points whose x belongs to a subset S_1 of x points the strip over S_1, and similarly the set whose y belong to S_2 the strip over S_2. We divide x and y into a collection of non-overlapping measurable subsets X_i and Y_i approximate to the rate of transmission R by

$$R_1 = \frac{1}{T}\sum_i P(X_i,Y_i)\log\frac{P(X_i,Y_i)}{P(X_i)P(Y_i)}$$

where

$$\begin{aligned} P(X_i) &\quad \text{is the probability measure of the strip over } X_i \\ P(Y_i) &\quad \text{is the probability measure of the strip over } Y_i \\ P(X_i,Y_i) &\quad \text{is the probability measure of the intersection of the strips.} \end{aligned}$$

A further subdivision can never decrease R_1. For let X_1 be divided into $X_1 = X_1' + X_1''$ and let

$$\begin{aligned} P(Y_1) &= a & P(X_1) &= b+c \\ P(X_1') &= b & P(X_1',Y_1) &= d \\ P(X_1'') &= c & P(X_1'',Y_1) &= e \end{aligned}$$
$$P(X_1,Y_1) = d+e.$$

Then in the sum we have replaced (for the X_1, Y_1 intersection)

$$(d+e)\log\frac{d+e}{a(b+c)} \quad \text{by} \quad d\log\frac{d}{ab} + e\log\frac{e}{ac}.$$

It is easily shown that with the limitation we have on b, c, d, e,

$$\left[\frac{d+e}{b+c}\right]^{d+e} \le \frac{d^d e^e}{b^d c^e}$$

and consequently the sum is increased. Thus the various possible subdivisions form a directed set, with R monotonic increasing with refinement of the subdivision. We may define R unambiguously as the least upper bound for R_1 and write it

$$R = \frac{1}{T}\iint P(x,y)\log\frac{P(x,y)}{P(x)P(y)}\,dx\,dy.$$

This integral, understood in the above sense, includes both the continuous and discrete cases and of course many others which cannot be represented in either form. It is trivial in this formulation that if x and u are in one-to-one correspondence, the rate from u to y is equal to that from x to y. If v is any function of y (not necessarily with an inverse) then the rate from x to y is greater than or equal to that from x to v since, in the calculation of the approximations, the subdivisions of y are essentially a finer subdivision of those for v. More generally if y and v are related not functionally but statistically, i.e., we have a probability measure space (y,v), then $R(x,v) \le R(x,y)$. This means that any operation applied to the received signal, even though it involves statistical elements, does not increase R.

Another notion which should be defined precisely in an abstract formulation of the theory is that of "dimension rate," that is the average number of dimensions required per second to specify a member of an ensemble. In the band limited case $2W$ numbers per second are sufficient. A general definition can be framed as follows. Let $f_\alpha(t)$ be an ensemble of functions and let $\rho_T[f_\alpha(t), f_\beta(t)]$ be a metric measuring

the "distance" from f_α to f_β over the time T (for example the R.M.S. discrepancy over this interval.) Let $N(\epsilon,\delta,T)$ be the least number of elements f which can be chosen such that all elements of the ensemble apart from a set of measure δ are within the distance ϵ of at least one of those chosen. Thus we are covering the space to within ϵ apart from a set of small measure δ. We define the dimension rate λ for the ensemble by the triple limit

$$\lambda = \operatorname*{Lim}_{\delta\to 0} \operatorname*{Lim}_{\epsilon\to 0} \operatorname*{Lim}_{T\to\infty} \frac{\log N(\epsilon,\delta,T)}{T\log\epsilon}.$$

This is a generalization of the measure type definitions of dimension in topology, and agrees with the intuitive dimension rate for simple ensembles where the desired result is obvious.

A New Approach to Linear Filtering and Prediction Problems[1]

R. E. KALMAN
Research Institute for Advanced Study,[2]
Baltimore, Md.

The classical filtering and prediction problem is re-examined using the Bode-Shannon representation of random processes and the "state transition" method of analysis of dynamic systems. New results are:

(1) The formulation and methods of solution of the problem apply without modification to stationary and nonstationary statistics and to growing-memory and infinite-memory filters.

(2) A nonlinear difference (or differential) equation is derived for the covariance matrix of the optimal estimation error. From the solution of this equation the coefficients of the difference (or differential) equation of the optimal linear filter are obtained without further calculations.

(3) The filtering problem is shown to be the dual of the noise-free regulator problem.

The new method developed here is applied to two well-known problems, confirming and extending earlier results.

The discussion is largely self-contained and proceeds from first principles; basic concepts of the theory of random processes are reviewed in the Appendix.

Introduction

AN IMPORTANT class of theoretical and practical problems in communication and control is of a statistical nature. Such problems are: (i) Prediction of random signals; (ii) separation of random signals from random noise; (iii) detection of signals of known form (pulses, sinusoids) in the presence of random noise.

In his pioneering work, Wiener [1][3] showed that problems (i) and (ii) lead to the so-called Wiener-Hopf integral equation; he also gave a method (spectral factorization) for the solution of this integral equation in the practically important special case of stationary statistics and rational spectra.

Many extensions and generalizations followed Wiener's basic work. Zadeh and Ragazzini solved the finite-memory case [2]. Concurrently and independently of Bode and Shannon [3], they also gave a simplified method [2] of solution. Booton discussed the nonstationary Wiener-Hopf equation [4]. These results are now in standard texts [5-6]. A somewhat different approach along these main lines has been given recently by Darlington [7]. For extensions to sampled signals, see, e.g., Franklin [8], Lees [9]. Another approach based on the eigenfunctions of the Wiener-Hopf equation (which applies also to nonstationary problems whereas the preceding methods in general don't), has been pioneered by Davis [10] and applied by many others, e.g., Shinbrot [11], Blum [12], Pugachev [13], Solodovnikov [14].

In all these works, the objective is to obtain the specification of a linear dynamic system (Wiener filter) which accomplishes the prediction, separation, or detection of a random signal.[4]

Present methods for solving the Wiener problem are subject to a number of limitations which seriously curtail their practical usefulness:

(1) The optimal filter is specified by its impulse response. It is not a simple task to synthesize the filter from such data.

(2) Numerical determination of the optimal impulse response is often quite involved and poorly suited to machine computation. The situation gets rapidly worse with increasing complexity of the problem.

(3) Important generalizations (e.g., growing-memory filters, nonstationary prediction) require new derivations, frequently of considerable difficulty to the nonspecialist.

(4) The mathematics of the derivations are not transparent. Fundamental assumptions and their consequences tend to be obscured.

This paper introduces a new look at this whole assemblage of problems, sidestepping the difficulties just mentioned. The following are the highlights of the paper:

(5) *Optimal Estimates and Orthogonal Projections.* The Wiener problem is approached from the point of view of conditional distributions and expectations. In this way, basic facts of the Wiener theory are quickly obtained; the scope of the results and the fundamental assumptions appear clearly. It is seen that all statistical calculations and results are based on first and second order averages; no other statistical data are needed. Thus difficulty (4) is eliminated. This method is well known in probability theory (see pp. 75–78 and 148–155 of Doob [15] and pp. 455–464 of Loève [16]) but has not yet been used extensively in engineering.

(6) *Models for Random Processes.* Following, in particular, Bode and Shannon [3], arbitrary random signals are represented (up to second order average statistical properties) as the output of a linear dynamic system excited by independent or uncorrelated random signals ("white noise"). This is a standard trick in the engineering applications of the Wiener theory [2–7]. The approach taken here differs from the conventional one only in the way in which linear dynamic systems are described. We shall emphasize the concepts of *state* and *state transition;* in other words, linear systems will be specified by systems of first-order difference (or differential) equations. This point of view is

[1] This research was supported in part by the U. S. Air Force Office of Scientific Research under Contract AF 49 (638)-382.

[2] 7212 Bellona Ave.

[3] Numbers in brackets designate References at end of paper.

[4] Of course, in general these tasks may be done better by nonlinear filters. At present, however, little or nothing is known about how to obtain (both theoretically and practically) these nonlinear filters.

Contributed by the Instruments and Regulators Division and presented at the Instruments and Regulators Conference, March 29– Apri1 2, 1959, of THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS.

NOTE: Statements and opinions advanced in papers are to be understood as individual expressions of their authors and not those of the Society. Manuscript received at ASME Headquarters, February 24, 1959. Paper No. 59—IRD-11.

natural and also necessary in order to take advantage of the simplifications mentioned under (5).

(7) *Solution of the Wiener Problem.* With the state-transition method, a single derivation covers a large variety of problems: growing and infinite memory filters, stationary and nonstationary statistics, etc.; difficulty (3) disappears. Having guessed the "state" of the estimation (i.e., filtering or prediction) problem correctly, one is led to a nonlinear difference (or differential) equation for the covariance matrix of the optimal estimation error. This is vaguely analogous to the Wiener-Hopf equation. Solution of the equation for the covariance matrix starts at the time t_0 when the first observation is taken; at each later time t the solution of the equation represents the covariance of the optimal prediction error given observations in the interval (t_0, t). From the covariance matrix at time t we obtain at once, without further calculations, the coefficients (in general, time-varying) characterizing the optimal linear filter.

(8) *The Dual Problem.* The new formulation of the Wiener problem brings it into contact with the growing new theory of control systems based on the "state" point of view [17–24]. It turns out, *surprisingly*, that the Wiener problem is the *dual* of the noise-free optimal regulator problem, which has been solved previously by the author, using the state-transition method to great advantage [18, 23, 24]. The mathematical background of the two problems is identical—this has been suspected all along, but until now the analogies have never been made explicit.

(9) *Applications.* The power of the new method is most apparent in theoretical investigations and in numerical answers to complex practical problems. In the latter case, it is best to resort to machine computation. Examples of this type will be discussed later. To provide some feel for applications, two standard examples from nonstationary prediction are included; in these cases the solution of the nonlinear difference equation mentioned under (7) above can be obtained even in closed form.

For easy reference, the main results are displayed in the form of theorems. Only Theorems 3 and 4 are original. The next section and the Appendix serve mainly to review well-known material in a form suitable for the present purposes.

Notation Conventions

Throughout the paper, we shall deal mainly with *discrete* (or *sampled)* dynamic systems; in other words, signals will be observed at equally spaced points in time *(sampling instants).* By suitable choice of the time scale, the constant intervals between successive sampling instants *(sampling periods)* may be chosen as unity. Thus variables referring to time, such as t, t_0, τ, T will always be integers. The restriction to discrete dynamic systems is not at all essential (at least from the engineering point of view); by using the discreteness, however, we can keep the mathematics rigorous and yet elementary. Vectors will be denoted by small bold-face letters: $\mathbf{a}$, $\mathbf{b}$, ..., $\mathbf{u}$, $\mathbf{x}$, $\mathbf{y}$, ... A *vector* or more precisely an *n-vector* is a set of n numbers x_1, ... x_n; the x_i are the *co-ordinates* or *components* of the vector $\mathbf{x}$.

Matrices will be denoted by capital bold-face letters: $\mathbf{A}$, $\mathbf{B}$, $\mathbf{Q}$, $\mathbf{\Phi}$, $\mathbf{\Psi}$, …; they are $m \times n$ arrays of elements a_{ij}, b_{ij}, q_{ij},... The *transpose* (interchanging rows and columns) of a matrix will be denoted by the prime. In manipulating formulas, it will be convenient to regard a vector as a matrix with a single column.

Using the conventional definition of matrix multiplication, we write the *scalar product* of two n-vectors $\mathbf{x}$, $\mathbf{y}$ as

$$\mathbf{x'y} = \sum_{i=1}^{n} x_i y_i = \mathbf{y'x}$$

The scalar product is clearly a scalar, i.e., not a vector, quantity. Similarly, the quadratic form associated with the $n \times n$ matrix $\mathbf{Q}$ is,

$$\mathbf{x'Qx} = \sum_{i,j=1}^{n} x_i q_{ij} x_j$$

We define the expression $\mathbf{xy'}$ where $\mathbf{x'}$ is an m-vector and $\mathbf{y}$ is an n-vector to be the $m \times n$ matrix with elements $x_i y_j$.

We write $E(\mathbf{x}) = E\mathbf{x}$ for the expected value of the random vector $\mathbf{x}$ (see Appendix). It is usually convenient to omit the brackets after E. This does not result in confusion in simple cases since constants and the operator E commute. Thus $E\mathbf{xy'}$ = matrix with elements $E(x_i y_j)$; $E\mathbf{x}E\mathbf{y'}$ = matrix with elements $E(x_i)E(y_j)$.

For ease of reference, a list of the principal symbols used is given below.

Optimal Estimates

t	time in general, present time.
t_0	time at which observations start.
$x_1(t)$, $x_2(t)$	basic random variables.
$y(t)$	observed random variable.
$x_1^*(t_1\|t)$	optimal estimate of $x_1(t_1)$ given $y(t_0)$, …, $y(t)$.
L	loss function (non random function of its argument).
ε	estimation error (random variable).

Orthogonal Projections

$\mathcal{Y}(t)$	linear manifold generated by the random variables $y(t_0)$, …, $y(t)$.
$\bar{x}(t_1\|t)$	orthogonal projection of $x(t_1)$ on $\mathcal{Y}(t)$.
$\tilde{x}(t_1\|t)$	component of $x(t_1)$ orthogonal to $\mathcal{Y}(t)$.

Models for Random Processes

$\mathbf{\Phi}(t+1; t)$	transition matrix
$\mathbf{Q}(t)$	covariance of random excitation

Solution of the Wiener Problem

$\mathbf{x}(t)$	basic random variable.
$\mathbf{y}(t)$	observed random variable.
$\mathcal{Y}(t)$	linear manifold generated by $\mathbf{y}(t_0)$, …, $\mathbf{y}(t)$.
$\mathcal{Z}(t)$	linear manifold generated by $\mathbf{y}(t\|t-1)$.
$\mathbf{x}^*(t_1\|t)$	optimal estimate of $\mathbf{x}(t_1)$ given $\mathcal{Y}(t)$.
$\tilde{\mathbf{x}}(t_1\|t)$	error in optimal estimate of $\mathbf{x}(t_1)$ given $\mathcal{Y}(t)$.

Optimal Estimates

To have a concrete description or the type of problems to be studied, consider the following situation. We are given signal $x_1(t)$ and noise $x_2(t)$. Only the sum $y(t) = x_1(t) + x_2(t)$ can be observed. Suppose we have observed and know exactly the values of $y(t_0)$, ..., $y(t)$. What can we infer from this knowledge in regard to the (unobservable) value of the signal at $t = t_1$, where t_1 may be less than, equal to, or greater than t? If $t_1 < t$, this is a *data-smoothing (interpolation)* problem. If $t_1 = t$, this is called *filtering.* If $t_1 > t$, we have a *prediction* problem. Since our treatment will be general enough to include these and similar problems, we shall use hereafter the collective term *estimation.*

As was pointed out by Wiener [1], the natural setting of the estimation problem belongs to the realm of probability theory and statistics. Thus signal, noise, and their sum will be random variables, and consequently they may be regarded as random processes. From the probabilistic description of the random processes we can determine the probability with which a particular sample of the signal and noise will occur. For any given set of measured values $\eta(t_0)$, ..., $\eta(t)$ of the random variable $y(t)$ one can then also determine, in principle, the probability of simultaneous occurrence of various values $\xi_1(t)$ of the random variable $x_1(t_1)$. This is the conditional probability distribution function

$$Pr[x_1(t_1) \le \xi_1 | y(t_0) = \eta(t_0), \ldots, y(t) = \eta(t)] = F(\xi_1) \qquad (1)$$

Evidently, $F(\xi_1)$ represents all the information which the measurement of the random variables $y(t_0)$, ..., $y(t)$ has conveyed about the random variable $x_1(t_1)$. Any statistical estimate of the random variable $x_1(t_1)$ will be some function of this distribution and therefore a (nonrandom) function of the random variables $y(t_0)$, ..., $y(t)$. This statistical estimate is denoted by $X_1(t_1|t)$, or by just $X_1(t_1)$ or X_1 when the set of observed random variables or the time at which the estimate is required are clear from context.

Suppose now that X_1 is given as a fixed function of the random variables $y(t_0)$, ..., $y(t)$. Then X_1 is itself a random variable and its actual value is known whenever the actual values of $y(t_0)$, ..., $y(t)$ are known. In general, the actual value of $X_1(t_1)$ will be different from the (unknown) actual value of $x_1(t_1)$. To arrive at a rational way of determining X_1, it is natural to assign a *penalty* or *loss* for incorrect estimates. Clearly, the loss should be a (i) positive, (ii) nondecreasing function of the *estimation error* $\varepsilon = x_1(t_1) - X_1(t_1)$. Thus we define a *loss function* by

$$L(0) = 0$$

$$L(\varepsilon_2) \ge L(\varepsilon_1) \ge 0 \quad \text{when} \quad \varepsilon_2 \ge \varepsilon_1 \ge 0 \qquad (2)$$

$$L(\varepsilon) = L(-\varepsilon)$$

Some common examples of loss functions are: $L(\varepsilon) = a\varepsilon^2$, $a\varepsilon^4$, $a|\varepsilon|$, $a[1 - \exp(-\varepsilon^2)]$, etc., where a is a positive constant.

One (but by no means the only) natural way of choosing the random variable X_1 is to require that this choice should minimize the average loss or risk

$$E\{L[x_1(t_1) - X_1(t_1)]\} = E[E\{L[x(t_1) - X_1(t_1)]|y(t_0), \ldots, y(t)\}] \qquad (3)$$

Since the first expectation on the right-hand side of (3) does not depend on the choice of X_1 but only on $y(t_0)$, ..., $y(t)$, it is clear that minimizing (3) is equivalent to minimizing

$$E\{L[x_1(t_1) - X_1(t_1)]|y(t_0), ..., y(t)\} \qquad (4)$$

Under just slight additional assumptions, optimal estimates can be characterized in a simple way.

Theorem 1. *Assume that L is of type* (2) *and that the conditional distribution function* $F(\xi)$ *defined by* (1) *is:*

(*A*) *symmetric about the mean* $\bar{\xi}$:

$$F(\xi - \bar{\xi}) = 1 - F(\bar{\xi} - \xi)$$

(*B*) *convex for* $\xi \le \bar{\xi}$:

$$F(\lambda\xi_1 + (1 - \lambda)\xi_2) \le \lambda F(\xi_1) + (1 - \lambda)F(\xi_2)$$

for all $\xi_1, \xi_2 \le \bar{\xi}$ *and* $0 \le \lambda \le 1$

Then the random variable $x_1^*(t_1|t)$ *which minimizes the average loss* (3) *is the conditional expectation*

$$x_1^*(t_1|t) = E[x_1(t_1)|y(t_0), \ldots, y(t)] \qquad (5)$$

Proof: As pointed out recently by Sherman [25], this theorem follows immediately from a well-known lemma in probability theory.

Corollary. *If the random processes* $\{x_1(t)\}$, $\{x_2(t)\}$, and $\{y(t)\}$ *are gaussian, Theorem 1 holds.*

Proof: By Theorem 5, (*A*) (see Appendix), conditional distributions on a gaussian random process are gaussian. Hence the requirements of Theorem 1 are always satisfied.

In the control system literature, this theorem appears sometimes in a form which is more restrictive in one way and more general in another way:

Theorem I-a. *If* $L(\varepsilon) = \varepsilon^2$, *then Theorem 1 is true without assumptions* (*A*) and (*B*).

Proof: Expand the conditional expectation (4):

$$E[x_1^2(t_1)|y(t_0), \ldots, y(t)] - 2X_1(t_1)E[x_1(t_1)|y(t_0), \ldots, y(t)] + X_1^2(t_1)$$

and differentiate with respect to $X_1(t_1)$. This is not a completely rigorous argument; for a simple rigorous proof see Doob [15], pp. 77–78.

Remarks. (*a*) As far as the author is aware, it is not known what is the most general class of random processes $\{x_1(t)\}$, $\{x_2(t)\}$ for which the conditional distribution function satisfies the requirements of Theorem 1.

(*b*) Aside from the note of Sherman, Theorem 1 apparently has never been stated explicitly in the control systems literature. In fact, one finds many statements to the effect that loss functions of the general type (2) cannot be conveniently handled mathematically.

(*c*) In the sequel, we shall be dealing mainly with vector-valued random variables. In that case, the estimation problem is stated as: Given a vector-valued random process $\{\mathbf{x}(t)\}$ and observed random variables $\mathbf{y}(t_0)$, ..., $\mathbf{y}(t)$, where $\mathbf{y}(t) = \mathbf{M}\mathbf{x}(t)$ ($\mathbf{M}$ being a singular matrix; in other words, not all co-ordinates of $\mathbf{x}(t)$ can be observed), find an estimate $\mathbf{X}(t_1)$ which minimizes the expected loss $E[L(\|\mathbf{x}(t_1) - \mathbf{X}(t_1)\|)]$, $\| \ \|$ being the norm of a vector.

Theorem 1 remains true in the vector case also, provided we re- quire that the conditional distribution function of the n co-ordi- nates of the vector $\mathbf{x}(t_1)$,

$$Pr[x_1(t_1) \le \xi_1, \ldots, x_n(t_1) \le \xi_n | \mathbf{y}(t_0), \ldots, \mathbf{y}(t)] = F(\xi_1, \ldots, \xi_n)$$

be symmetric with respect to the n variables $\xi_1 - \bar{\xi}_1, \ldots, \xi_n - \bar{\xi}_n$ and convex in the region where all of these variables are negative.

Orthogonal Projections

The explicit calculation of the optimal estimate as a function of the observed variables is, in general, impossible. There is an important exception: The processes $\{x_1(t)\}$, $\{x_2(t)\}$ are gaussian.

On the other hand, if we attempt to get an optimal estimate under the restriction $L(\varepsilon) = \varepsilon^2$ and the additional requirement that the estimate be a linear function of the observed random variables, we get an estimate which is identical with the optimal estimate in the gaussian case, without the assumption of linearity or quadratic loss function. This shows that results obtainable by linear estimation can be bettered by nonlinear estimation only when (i) the random processes are nongaussian and even then (in view of Theorem 5, (*C*)) only (ii) by considering at least third-order probability distribution functions.

In the special cases just mentioned, the explicit solution of the estimation problem is most easily understood with the help of a geometric picture. This is the subject of the present section.

Consider the (real-valued) random variables $y(t_0)$, …, $y(t)$. The set of all linear combinations of these random variables with real coefficients

$$\sum_{i=t_0}^{t} a_i y(i) \qquad (6)$$

forms a *vector space (linear manifold)* which we denote by $\mathcal{Y}(t)$. We regard, abstractly, any expression of the form (6) as "point" or "vector" in $\mathcal{Y}(t)$; this use of the word "vector" should not be confused, of course, with "vector-valued" random variables, etc. Since we do not want to fix the value of t (i.e., the total number of possible observations), $\mathcal{Y}(t)$ should be regarded as a finite-dimensional subspace of the space of all possible observations.

Given any two vectors u, v in $\mathcal{Y}(t)$ (i.e., random variables expressible in the form (6)), we say that u and v are *orthogonal* if $Euv = 0$. Using the Schmidt orthogonalization procedure, as described for instance by Doob [15], p. 151, or by Loève [16], p. 459, it is easy to select an *orthonormal basis* in $\mathcal{Y}(t)$. By this is meant a set of vectors $e_{t_0}, \ldots, e_t$ in $\mathcal{Y}(t)$ such that any vector in $\mathcal{Y}(t)$ can be expressed as a unique linear combination of $e_{t_0}, \ldots, e_t$ and

$$\left.\begin{aligned} Ee_ie_j = \delta_{ij} = 1 \quad &\text{if} \quad i = j \\ = 0 \quad &\text{if} \quad i \neq j \end{aligned}\right\} \quad (i, j = t_0, \ldots, t) \tag{7}$$

Thus any vector $\bar{x}$ in $\mathcal{Y}(t)$. is given by

$$\bar{x} = \sum_{i=t_0}^{t} a_i e_i$$

and so the coefficients a_i can be immediately determined with the aid of (7):

$$E\bar{x}e_j = E\left(\sum_{i=t_0}^{t} a_i e_i\right)e_j = \sum_{i=t_0}^{t} a_i E e_i e_j = \sum_{i=t_0}^{t} a_i \delta_{ij} = a_j \tag{8}$$

It follows further that any random variable x (not necessarily in $\mathcal{Y}(t)$) can be uniquely decomposed into two parts: a part $\bar{x}$ in $\mathcal{Y}(t)$ and a part $\tilde{x}$ orthogonal to $\mathcal{Y}(t)$ (i.e., orthogonal to every vector in $'\mathcal{Y}(t)$). In fact, we can write

$$x = \bar{x} + \tilde{x} = \sum_{i=t_0}^{t} (Exe_i)e_i + \tilde{x} \tag{9}$$

Thus $\bar{x}$ is uniquely determined by equation (9) and is obviously a vector in $\mathcal{Y}(t)$. Therefore $\tilde{x}$ is also uniquely determined; it remains to check that it is orthogonal to $\mathcal{Y}(t)$:

$$E\tilde{x}e_i = E(x - \bar{x})e_i = Exe_i - E\bar{x}e_i$$

Now the co-ordinates of $\bar{x}$ with respect to the basis $e_{t_0}, \ldots, e_t$, are given either in the form $E\bar{x}e_i$ (as in (8)) or in the form Exe_i (as in (9)). Since the co-ordinates are unique, $Exe_i = E\bar{x}e_i$ $(i = t_0, \ldots, t)$; hence $E\tilde{x}e_i = 0$ and $\tilde{x}$ is orthogonal to every base vector e_i; and therefore to $\mathcal{Y}(t)$. We call $\bar{x}$ the *orthogonal projection* of x on $\mathcal{Y}(t)$.

There is another way in which the orthogonal projection can be characterized: $\bar{x}$ is that vector in $\mathcal{Y}(t)$ (i.e., that *linear* function of the random variables $y(t_0), \ldots, y(t)$) which minimizes the quadratic loss function. In fact, if $\bar{w}$ is any other vector in $\mathcal{Y}(t)$, we have

$$E(x - \bar{w})^2 = E(\tilde{x} + \bar{x} - \bar{w})^2 = E[(x - \bar{x}) + (\bar{x} - \bar{w})]^2$$

Since $\tilde{x}$ is orthogonal to every vector in $\mathcal{Y}(t)$ and in particular to $\bar{x} - \bar{w}$ we have

$$E(x - \bar{w})^2 = E(x - \bar{x})^2 + E(\bar{x} - \bar{w})^2 \geq E(x - \bar{x})^2 \tag{10}$$

This shows that, if $\bar{w}$ also minimizes the quadratic loss, we must have $E(\bar{x} - \bar{w})^2 = 0$ which means that the random variables $\bar{x}$ and $\bar{w}$ are equal (except possibly for a set of events whose probability is zero).

These results may be summarized as follows:

Theorem 2. Let $\{x(t)\}$, $\{y(t)\}$ *random processes with zero mean* (*i.e.*, $Ex(t) = Ey(t) = 0$ *for all* t). *We observe* $y(t_0), \ldots, y(t)$.

If either

(*A*) *the random processes* $\{x(t)\}$, $\{y(t)\}$ *are gaussian; or*

(*B*) *the optimal estimate is restricted to be a linear function of the observed random variables and* $L(\varepsilon) = \varepsilon^2$;

then

$$\begin{aligned} x^*(t_1|t) &= \textit{optimal estimate of } x(t_1) \text{ given } y(t_0), \ldots, y(t) \\ &= \textit{orthogonal projection } \bar{x}\,(t_1|t) \textit{ of } x(t_1) \text{ on } \mathcal{Y}(t). \end{aligned} \tag{11}$$

These results are well-known though not easily accessible in the control systems literature. See Doob [15], pp. 75–78, or Pugachev [26]. It is sometimes convenient to denote the orthogonal projection by

$$\bar{x}(t_1 \mid t) \equiv x^*(t_1 \mid t) = \hat{E}[x(t_1) \mid \mathcal{Y}(t)]$$

The notation $\hat{E}$ is motivated by part (*b*) of the theorem: If the stochastic processes in question are gaussian, then orthogonal projection is actually identical with conditional expectation.

Proof. (*A*) This is a direct consequence of the remarks in connection with (10).

(*B*) Since $x(t)$, $y(t)$ are random variables with zero mean, it is clear from formula (9) that the orthogonal part $\tilde{x}\,(t_1|t)$ of $x(t_1)$ with respect to the linear manifold $\mathcal{Y}(t)$ is also a random variable with zero mean. Orthogonal random variables with zero mean are uncorrelated; if they are also gaussian then (by Theorem 5 (*B*)) they are independent. Thus

$$\begin{aligned} 0 = E\,\tilde{x}\,(t_1|t) &= E[\,\tilde{x}\,(t_1|t)|y(t_0), \ldots, y(t)] \\ &= E[x\,(t_1) - \bar{x}\,(t_1|t)|y(t_0), \ldots, y(t)] \\ &= E[x\,(t_1)|y(t_0), \ldots, y(t)] - \bar{x}\,(t_1|t) = 0 \end{aligned}$$

Remarks. (*d*) A rigorous formulation of the contents of this section as $t \to \infty$ requires some elementary notions from the theory of Hilbert space. See Doob [15] and Loève [16].

(*e*) The physical interpretation of Theorem 2 is largely a matter of taste. If we are not worried about the assumption of gaussianness, part (*A*) shows that the orthogonal projection is the optimal estimate for all reasonable loss functions. If we do worry about gaussianness, even if we are resigned to consider only linear estimates, we know that orthogonal projections are *not* the optimal estimate for many reasonable loss functions. Since in practice it is difficult to ascertain to what degree of approximation a random process of physical origin is gaussian, it is hard to decide whether Theorem 2 has very broad or very limited significance.

(*f*) Theorem 2 is immediately generalized for the case of vector-valued random variables. In fact, we define the linear manifold $\mathcal{Y}(t)$ generated by $\mathbf{y}(t_0), \ldots, \mathbf{y}(t)$ to be the set of all linear combinations

$$\sum_{i=t_0}^{t} \sum_{j=1}^{m} a_{ij} y_j(i)$$

of all m co-ordinates of each of the random vectors $\mathbf{y}(t_0), \ldots, \mathbf{y}(t)$. The rest of the story proceeds as before.

(*g*) Theorem 2 states in effect that the optimal estimate under conditions (*A*) or (*B*) is a linear combination of all previous observations. In other words, the optimal estimate can be regarded as the output of a linear filter, with the input being the actually occurring values of the observable random variables; Theorem 2 gives a way of computing the impulse response of the optimal filter. As pointed out before, knowledge of this impulse response is not a complete solution of the problem; for this reason, no explicit formulas for the calculation of the impulse response will be given.

Models for Random Processes

In dealing with physical phenomena, it is not sufficient to give an empirical description but one must have also some idea of the underlying causes. Without being able to separate in some sense causes and effects, i.e., without the assumption of causality, one can hardly hope for useful results.

It is a fairly generally accepted fact that primary macroscopic sources of random phenomena are independent gaussian processes.[5] A well-known example is the noise voltage produced in a resistor due to thermal agitation. In most cases, *observed* random phenomena are not describable by independent random variables. The statistical dependence (correlation) between random signals observed at different times is usually explained by the presence of a dynamic system between the primary random source and the observer. *Thus a random function of time may be thought of as the output of a dynamic system excited by an independent gaussian random process.*

An important property of gaussian random signals is that they remain gaussian after passing through a linear system (Theorem 5 (*A*)). Assuming independent gaussian primary random sources, if the observed random signal is also gaussian, we may assume that the dynamic system between the observer and the primary source is *linear*. This conclusion may be forced on us also because of lack of detailed knowledge of the statistical properties of the observed random signal: Given any random process with known first and second-order averages, we can find a gaussian random process with the same properties (Theorem 5 (*C*)). Thus gaussian distributions and linear dynamics are natural, mutually plausible assumptions particularly when the statistical data are scant.

How is a dynamic system (linear or nonlinear) described? The fundamental concept is the notion of the *state*. By this is meant, intuitively, some quantitative information (a set of numbers, a function, etc.) which is the least amount of data one has to know about the past behavior of the system in order to predict its future behavior. The dynamics is then described in terms of *state transitions*, i.e., one must specify how one state is transformed into another as time passes.

A *linear* dynamic system may be described in general by the vector differential equation

$$\left.\begin{aligned} d\mathbf{x}/dt &= \mathbf{F}(t)\mathbf{x} + \mathbf{D}(t)\mathbf{u}(t) \\ \text{and} \qquad & \\ \mathbf{y}(t) &= \mathbf{M}(t)\mathbf{x}(t) \end{aligned}\right\} \tag{12}$$

where $\mathbf{x}$ is an n-vector, the *state* of the system (the components x_i of $\mathbf{x}$ are called *state variables*); $\mathbf{u}(t)$ is an m-vector ($m \le n$) representing the *inputs* to the system; $\mathbf{F}(t)$ and $\mathbf{D}(t)$ are $n \times n$, respectively, $n \times m$ matrices. If all coefficients of $\mathbf{F}(t)$, $\mathbf{D}(t)$, $\mathbf{M}(t)$ are constants, we say that the dynamic system (12) is *time-invariant* or *stationary*. Finally, $\mathbf{y}(t)$ is a *p-vector* denoting the outputs of the system; $\mathbf{M}(t)$ is an $n \times p$ matrix; $p \le n$

The physical interpretation of (12) has been discussed in detail elsewhere [18, 20, 23]. A look at the block diagram in Fig. 1 may be helpful. This is not an ordinary but a matrix block diagram (as revealed by the fat lines indicating signal flow). The integrator in

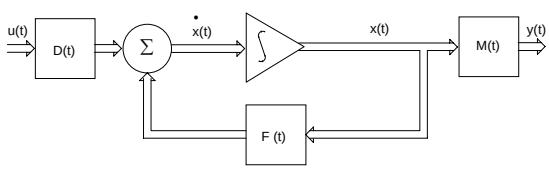


Fig 1. Matrix block diagram of the general linear continuous-dynamic system

[5] The probability distributions will be gaussian because macroscopic random effects may be thought of as the superposition of very many microscopic random effects; under very general conditions, such aggregate effects tend to be gaussian, regardless of the statistical properties of the microscopic effects. The assumption of independence in this context is motivated by the fact that microscopic phenomena tend to take place much more rapidly than macroscopic phenomena; thus primary random sources would appear to be independent on a macroscopic time scale.

Fig. 1 actually stands for n integrators such that the output of each is a state variable; $\mathbf{F}(t)$ indicates how the outputs of the integrators are fed back to the inputs of the integrators. Thus $f_{ij}(t)$ is the coefficient with which the output of the jth integrator is fed back to the input of the ith integrator. It is not hard to relate this formalism to more conventional methods of linear system analysis.

If we assume that the system (12) is stationary and that $\mathbf{u}(t)$ is constant during each sampling period, that is

$$\mathbf{u}(t + \tau) = \mathbf{u}(t); \quad 0 \le \tau < 1, \quad t = 0, 1, \ldots \tag{13}$$

then (12) can be readily transformed into the more convenient discrete form.

$$\mathbf{x}(t + 1) = \mathbf{\Phi}(1)\mathbf{x}(t) + \mathbf{\Delta}(1)\mathbf{u}(t); \quad t = 0, 1, \ldots$$

where [18, 20]

$$\mathbf{\Phi}(1) = \exp \mathbf{F} = \sum_{i=0}^{\infty} \mathbf{F}^i/i! \quad (\mathbf{F}^0 = \text{unit matrix})$$

and

$$\mathbf{\Delta}(1) = \left(\int_0^1 \exp \mathbf{F}\tau d\tau \right) \mathbf{D}$$

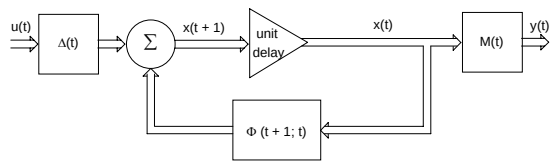


Fig 2. Matrix block diagram of the general linear discrete-dynamic system

See Fig. 2. One could also express $\exp \mathbf{F}\tau$ in closed form using Laplace transform methods [18, 20, 22, 24]. If $\mathbf{u}(t)$ satisfies (13) but the system (12) is nonstationary, we can write analogously

$$\left.\begin{aligned} \mathbf{x}(t + 1) &= \mathbf{\Phi}(t + 1; t) + \mathbf{\Delta}(t)\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{M}(t)\mathbf{x}(t) \end{aligned}\right\} \quad t = 0, 1, \ldots \tag{14}$$

but of course now $\mathbf{\Phi}(t + 1; t)$, $\mathbf{\Delta}(t)$ cannot be expressed in general in closed form. Equations of type (14) are encountered frequently also in the study of complicated sampled-data systems [22]. See Fig. 2

$\mathbf{\Phi}(t + 1; t)$ is the *transition matrix* of the system (12) or (14). The notation $\mathbf{\Phi}(t_2; t_1)$ (t_2, t_1 = integers) indicates transition from time t_1 to time t_2. Evidently $\mathbf{\Phi}(t; t) = \mathbf{I}$ = unit matrix. If the system (12) is stationary then $\mathbf{\Phi}(t + 1; t) = \mathbf{\Phi}(t + 1 - t) = \mathbf{\Phi}(1)$ = const. Note also the product rule: $\mathbf{\Phi}(t; s)\mathbf{\Phi}(s; r) = \mathbf{\Phi}(t; r)$ and the inverse rule $\mathbf{\Phi}^{-1}(t; s) = \mathbf{\Phi}(s; t)$, where t, s, r are integers. In a stationary system, $\mathbf{\Phi}(t; \tau) = \exp \mathbf{F}(t - \tau)$.

As a result of the preceding discussion, we shall represent random phenomena by the model

$$\mathbf{x}(t + 1) = \mathbf{\Phi}(t + 1; t)\mathbf{x}(t) + \mathbf{u}(t) \tag{15}$$

where $\{\mathbf{u}(t)\}$ is a vector-valued, independent, gaussian random process, with zero mean, which is completely described by (in view of Theorem 5 (*C*))

$$E\mathbf{u}(t) = \mathbf{0} \quad \text{for all } t;$$

$$E\mathbf{u}(t)\mathbf{u}'(s) = \mathbf{0} \quad \text{if } t \ne s$$

$$E\mathbf{u}(t)\mathbf{u}'(t) = \mathbf{G}(t).$$

Of course (Theorem 5 (*A*)), $\mathbf{x}(t)$ is then also a gaussian random process with zero mean, but it is no longer independent. In fact, if we consider (15) in the steady state (assuming it is a stable system), in other words, if we neglect the initial state $\mathbf{x}(t_0)$, then

$$\mathrm{x}(t) = \sum_{r=-\infty}^{t-1} \mathbf{\Phi}(t;\, r+1)\mathbf{u}(r).$$

Therefore if $t \geq s$ we have

$$E\mathbf{x}(t)\mathbf{x}'(s) = \sum_{r=-\infty}^{s-1} \mathbf{\Phi}(t;\, r+1)\mathbf{Q}(r)\,\mathbf{\Phi}'(s;\, r+1).$$

Thus if we assume a linear dynamic model and know the statistical properties of the gaussian random excitation, it is easy to find the corresponding statistical properties of the gaussian random process $\{\mathbf{x}(t)\}$.

In real life, however, the situation is usually reversed. One is given the covariance matrix $E\mathbf{x}(t)\mathbf{x}'(s)$ (or rather, one attempts to estimate the matrix from limited statistical data) and the problem is to get (15) and the statistical properties of $\mathbf{u}(t)$. This is a subtle and presently largely unsolved problem in experimentation and data reduction. As in the vast majority of the engineering literature on the Wiener problem, we shall find it convenient to start with the model (15) and regard the problem of obtaining the model itself as a separate question. To be sure, the two problems *should* be optimized jointly if possible; the author is not aware, however, of any study of the *joint* optimization problem.

In summary, the following assumptions are made about random processes:

Physical random phenomena may be thought of as *due to primary random sources exciting dynamic systems. The primary sources are assumed to be independent gaussian random processes with zero mean; the dynamic systems will be linear. The random processes are therefore described by models such as* (15). *The question of how the numbers specifying the model are obtained from experimental data will not be considered.*

Solution of the Wiener problem

Let us now define the principal problem of the paper.

Problem I. *Consider the dynamic model*

$$\mathbf{x}(t+1) = \mathbf{\Phi}(t+1;\, t)\mathbf{x}(t) + \mathbf{u}(t) \tag{16}$$

$$\mathbf{y}(t) = \mathbf{M}(t)\mathbf{x}(t) \tag{17}$$

where $\mathbf{u}(t)$ is *an independent gaussian random process of n-vectors with zero mean,* $\mathbf{x}(t)$ is *an n-vector,* $\mathbf{y}(t)$ is *a p-vector* ($p \leq n$), $\mathbf{\Phi}(t+1;\, t)$, $\mathbf{M}(t)$ *are* $n \times n$, *resp.* $p \times n$, *matrices whose elements are nonrandom functions of time.*

Given the observed values of $\mathbf{y}(t_0)$, ..., $\mathbf{y}(t)$ *find an estimate* $\mathbf{x}^*(t_1|t)$ *of* $\mathbf{x}(t_1)$ *which minimizes the expected loss.* (See Fig. 2, where $\mathbf{\Delta}(t) = \mathbf{I}$.)

This problem includes as a special case the problems of filtering, prediction, and data smoothing mentioned earlier. It includes also the problem of reconstructing all the state variables of a linear dynamic system from noisy observations of some of the state variables ($p < n$!).

From Theorem 2-*a* we know that the solution of Problem I is simply the orthogonal projection of $\mathbf{x}(t_1)$ on the linear manifold $\mathcal{Y}(t)$ generated by the observed random variables. As remarked in the Introduction, this is to be accomplished by means of a linear (not necessarily stationary!) dynamic system of the general form (14). With this in mind, we proceed as follows.

Assume that $\mathbf{y}(t_0)$, ..., $\mathbf{y}(t-1)$ have been measured, i.e., that $\mathcal{Y}(t-1)$ is known. Next, at time t, the random variable $\mathbf{y}(t)$ is measured. As before let $\tilde{\mathbf{y}}(t|t-1)$ be the component of $\mathbf{y}(t)$ orthogonal to $\mathcal{Y}(t-1)$. If $\tilde{\mathbf{y}}(t|t-1) \equiv 0$, which means that the values of all components of this random vector are zero for almost every possible event, then $\mathcal{Y}(t)$ is obviously the same as $\mathcal{Y}(t-1)$ and therefore the measurement of $\mathbf{y}(t)$ does not convey any additional information. This is not likely to happen in a physically meaningful situation. In any case, $\tilde{\mathbf{y}}(t|t-1)$ generates a linear manifold (possibly 0) which we denote by $\mathcal{Z}(t)$. By definition, $\mathcal{Y}(t-1)$ and $\mathcal{Z}(t)$ taken together are the same manifold as $\mathcal{Y}(t)$, and every vector in $\mathcal{Z}(t)$ is orthogonal to every vector in $\mathcal{Y}(t-1)$.

Assuming by induction that $\mathbf{x}^*(t_1-1|t-1)$ is known, we can write:

$$\begin{aligned}\mathbf{x}^*(t_1|t) &= \hat{E}\,[\mathbf{x}(t_1)|\mathcal{Y}(t)] = \hat{E}\,[\mathbf{x}(t_1)|\mathcal{Y}(t-1)] + \hat{E}\,[\mathbf{x}(t_1)|\mathcal{Z}(t)] \\ &= \mathbf{\Phi}(t+1;\, t)\,\mathbf{x}^*(t_1-1|t-1) + \hat{E}\,[\mathbf{u}(t_1-1)|\mathcal{Y}(t-1)] \\ &\qquad + \hat{E}\,[\mathbf{x}(t_1)|\mathcal{Z}(t)] \end{aligned} \tag{18}$$

where the last line is obtained using (16).

Let $t_1 = t + s$, where s is any integer. If $s \geq 0$, then $\mathbf{u}(t_1-1)$ is independent of $\mathcal{Y}(t-1)$. This is because $\mathbf{u}(t_1-1) = \mathbf{u}(t+s-1)$ is then independent of $\mathbf{u}(t-2)$, $\mathbf{u}(t-3)$, ... and therefore by (16–17), independent of $\mathbf{y}(t_0)$, ..., $\mathbf{y}(t-1)$, hence independent of $\mathcal{Y}(t-1)$. Since, for all t, $\mathbf{u}(t_0)$ has zero mean by assumption, it follows that $\mathbf{u}(t_1-1)$ ($s \geq 0$) is orthogonal to $\mathcal{Y}(t-1)$. Thus if $s \geq 0$, the second term on the right-hand side of (18) vanishes; if $s < 0$, considerable complications result in evaluating this term. We shall consider only the case $t_1 \geq t$. Furthermore, it will suffice to consider in detail only the case $t_1 = t + 1$ since the other cases can be easily reduced to this one.

The last term in (18) must be a linear operation on the random variable $\tilde{\mathbf{y}}(t|t-1)$:

$$\hat{E}\,[\mathbf{x}(t+1)|\mathcal{Z}(t)] = \mathbf{\Delta}^*(t)\,\tilde{\mathbf{y}}(t|t-1) \tag{19}$$

where $\mathbf{\Delta}^*(t)$ is an $n \times p$ matrix, and the star refers to "optimal filtering".

The component of $\mathbf{y}(t)$ lying in $\mathcal{Y}(t-1)$ is $\bar{\mathbf{y}}(t|t-1) = \mathbf{M}(t)\mathbf{x}^*(t|t-1)$. Hence

$$\tilde{\mathbf{y}}(t|t-1) = \mathbf{y}(t) - \bar{\mathbf{y}}(t|t-1) = \mathbf{y}(t) - \mathbf{M}(t)\mathbf{x}^*(t|t-1). \tag{20}$$

Combining (18-20) (see Fig. 3) we obtain

$$\mathbf{x}^*(t+1|t) = \mathbf{\Phi}^*(t+1;\, t)\mathbf{x}^*(t|t-1) + \mathbf{\Delta}^*(t)\mathbf{y}(t) \tag{21}$$

where

$$\mathbf{\Phi}^*(t+1;\, t) = \mathbf{\Phi}(t+1;\, t) - \mathbf{\Delta}^*(t)\mathbf{M}(t) \tag{22}$$

Thus optimal estimation is performed by a linear dynamic system of the same form as (14). The state of the estimator is the previous estimate, the input is the last measured value of the observable random variable $\mathbf{y}(t)$, the transition matrix is given by (22). Notice that physical realization of the optimal filter requires only (*i*) the model of the random process (*ii*) the operator $\mathbf{\Delta}^*(t)$.

The estimation error is also governed by a linear dynamic system. In fact,

$$\begin{aligned}\tilde{\mathbf{x}}(t+1|t) &= \mathbf{x}(t+1) - \mathbf{x}^*(t+1|t) \\ &= \mathbf{\Phi}(t+1;\, t)\mathbf{x}(t) + \mathbf{u}(t) - \mathbf{\Phi}^*(t+1;\, t)\mathbf{x}^*(t|t-1) \\ &\qquad - \mathbf{\Delta}^*(t)\mathbf{M}(t)\mathbf{x}(t)\end{aligned}$$

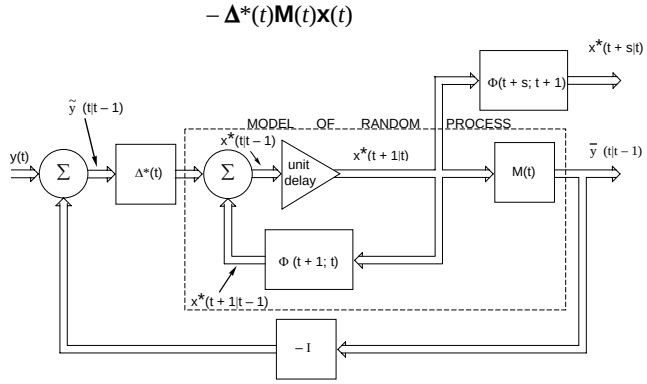


Fig. 3 Matrix block diagram of optimal filter

$$= \mathbf{\Phi}^*(t+1;\, t)\, \tilde{\mathbf{x}}\,(t|t-1) + \mathbf{u}(t) \tag{23}$$

Thus $\mathbf{\Phi}^*$ is also the transition matrix of the linear dynamic system governing the error.

From (23) we obtain at once a recursion relation for the covariance matrix $\mathbf{P}^*(t)$ of the optimal error $\tilde{\mathbf{x}}\,(t|t-1)$. Noting that $\mathbf{u}(t)$ is independent of $\mathbf{x}(t)$ and therefore of $\tilde{\mathbf{x}}\,(t|t-1)$ we get

$$\begin{aligned}\mathbf{P}^*(t+1) &= E\,\tilde{\mathbf{x}}\,(t+1|t)\,\tilde{\mathbf{x}}\,'(t+1|t)\\ &= \mathbf{\Phi}^*(t+1;\, t)E\,\tilde{\mathbf{x}}\,(t|t-1)\,\tilde{\mathbf{x}}\,'(t|t-1)\mathbf{\Phi}^{*\prime}\,(t+1;\, t) + \mathbf{Q}(t)\\ &= \mathbf{\Phi}^*(t+1;\, t)E\,\tilde{\mathbf{x}}\,(t|t-1)\,\tilde{\mathbf{x}}\,'(t|t-1)\mathbf{\Phi}'(t+1;\, t) + \mathbf{Q}(t)\\ &= \mathbf{\Phi}^*(t+1;\, t)\mathbf{P}^*(t)\mathbf{\Phi}'(t+1;\, t) + \mathbf{Q}(t)\end{aligned} \tag{24}$$

where $\mathbf{Q}(t) = E\mathbf{u}(t)\mathbf{u}'(t)$.

There remains the problem of obtaining an explicit formula for $\mathbf{\Delta}^*$ (and thus also for $\mathbf{\Phi}^*$). Since,

$$\tilde{\mathbf{x}}\,(t+1)|Z(t)) = \mathbf{x}(t+1) - \hat{E}\,[\mathbf{x}(t+1)|\mathcal{Z}(t)]$$

is orthogonal to $\tilde{\mathbf{y}}\,(t\,|t-1)$, it follows that by (19) that

$$\begin{aligned}0 &= E[\mathbf{x}(t+1) - \mathbf{\Delta}^*(t)\,\tilde{\mathbf{y}}\,(t|t-1)]\,\tilde{\mathbf{y}}\,'(t|t-1)\\ &= E\mathbf{x}(t+1)\,\tilde{\mathbf{y}}\,'(t|t-1) - \mathbf{\Delta}^*(t)E\;\tilde{\mathbf{y}}\,(t|t-1)\,\tilde{\mathbf{y}}\,'(t|t-1).\end{aligned}$$

Noting that $\bar{\mathbf{x}}\,(t+1|t-1)$ is orthogonal to $\mathcal{Z}(t)$, the definition of $\mathbf{P}(t)$ given earlier, and (17), it follows further

$$\begin{aligned}0 &= E\,\tilde{\mathbf{x}}\,(t+1|t-1)\,\tilde{\mathbf{y}}\,'(t|t-1) - \mathbf{\Delta}^*(t)\mathbf{M}(t)\mathbf{P}^*(t)\mathbf{M}'(t)\\ &= E[\mathbf{\Phi}(t+1;\, t)\,\tilde{\mathbf{x}}\,(t|t-1) + \mathbf{u}(t|t-1)]\,\tilde{\mathbf{x}}\,'(t|t-1)\mathbf{M}'(t)\\ &\qquad - \mathbf{\Delta}^*(t)\mathbf{M}(t)\mathbf{P}^*(t)\mathbf{M}'(t).\end{aligned}$$

Finally, since $\mathbf{u}(t)$ is independent of $\mathbf{x}(t)$,

$$0 = \mathbf{\Phi}(t+1;\, t)\mathbf{P}^*(t)\mathbf{M}'(t) - \mathbf{\Delta}^*(t)\mathbf{M}(t)\mathbf{P}^*(t)\mathbf{M}'(t).$$

Now the matrix $\mathbf{M}(t)\mathbf{P}^*(t)\mathbf{M}'(t)$will be positive definite and hence invertible whenever $\mathbf{P}^*(t)$ is positive definite, provided that none of the rows of $\mathbf{M}(t)$ are linearly dependent at any time, in other words, that none of the observed scalar random variables $y_1(t)$, ..., $y_m(t)$, is a linear combination of the others. Under these circumstances we get finally:

$$\mathbf{\Delta}^*(t) = \mathbf{\Phi}(t+1;\, t)\mathbf{P}^*(t)\mathbf{M}'(t)[\mathbf{M}(t)\mathbf{P}^*(t)\mathbf{M}'(t)]^{-1} \tag{25}$$

Since observations start at t_0, $\tilde{\mathbf{x}}\,(t_0|t_0-1) = \mathbf{x}(t_0)$; to begin the iterative evaluation of $\mathbf{P}^*(t)$ by means of equation (24), we must obviously specify $\mathbf{P}^*(t_0) = E\mathbf{x}(t_0)\mathbf{x}'(t_0)$. Assuming this matrix is positive definite, equation (25) then yields $\mathbf{\Delta}^*(t_0)$; equation (22) $\mathbf{\Phi}^*(t_0+1;\, t_0)$, and equation (24) $\mathbf{P}^*(t_0+1)$, completing the cycle. If now $\mathbf{Q}(t)$ is positive definite, then all the $\mathbf{P}^*(t)$ will be positive definite and the requirements in deriving (25) will be satisfied at each step.

Now we remove the restriction that $t_1 = t+1$. Since $\mathbf{u}(t)$ is orthogonal to $\mathcal{Y}(t)$, we have

$$\mathbf{x}^*(t+1|t) = \hat{E}\,[\mathbf{\Phi}(t+1;\, t)\mathbf{x}(t) + \mathbf{u}(t)|\mathcal{Y}(t)] = \mathbf{\Phi}(t+1;\, t)\mathbf{x}^*(t|t)$$

Hence if $\mathbf{\Phi}(t+1;\, t)$ has an inverse $\mathbf{\Phi}(t;\, t+1)$ (which is always the case when $\mathbf{\Phi}$ is the transition matrix of a dynamic system describable by a differential equation) we have

$$\mathbf{x}^*(t|t) = \mathbf{\Phi}(t;\, t+1)\mathbf{x}^*(t+1|t)$$

If $t_1 \geq t+1$, we first observe by repeated application of (16) that

$$\mathbf{x}(t+s) = \mathbf{\Phi}(t+s;\, t+1)\mathbf{x}(t+1) + \sum_{r1}^{s-1} \mathbf{\Phi}(t+s;\, t+r)\mathbf{u}(t+r) \qquad (s \geq 1)$$

Since $\mathbf{u}(t+s-1)$, …, $\mathbf{u}(t+1)$ are all orthogonal to $\mathcal{Y}(t)$,

$$\begin{aligned}\mathbf{x}^*(t+s|t) &= \hat{E}\,[\mathbf{x}(t+s)|\mathcal{Y}(t)]\\ &= \hat{E}\,[\mathbf{\Phi}(t+s;\, \mathrm{t}+1)\mathbf{x}(t+1)|\mathcal{Y}(t)]\\ &= \mathbf{\Phi}(t+s;\, \mathrm{t}+1)\mathbf{x}^*(t+1|t) \qquad (s \geq 1)\end{aligned}$$

If $s < 0$, the results are similar, but $\mathbf{x}^*(t-s|t)$ will have $(1-s)(n-p)$ co-ordinates.

The results of this section may be summarized as follows:

Theorem 3. (*Solution of the Wiener Problem*)

Consider Problem I. The optimal estimate $\mathbf{x}^*(t+1|t)$ *of* $\mathbf{x}(t+1)$ *given* $\mathbf{y}(t_0)$, ..., $\mathbf{y}(t)$ is *generated by the linear dynamic system*

$$\mathbf{x}^*(t+1|t) = \mathbf{\Phi}^*(t+1;\, \mathrm{t})\mathbf{x}^*(t|t-1) + \mathbf{\Delta}^*(t)\mathbf{y}(t) \tag{21}$$

The estimation error is given by

$$\tilde{\mathbf{x}}\,(t+1|t) = \mathbf{\Phi}^*(t+1;\, t)\,\tilde{\mathbf{x}}\,(t|t-1) + \mathbf{u}(t) \tag{23}$$

The covariance matrix of the estimation error is

$$\mathrm{cov}\;\tilde{\mathbf{x}}\,(t|t-1) = E\,\tilde{\mathbf{x}}\,(t|t-1)\,\tilde{\mathbf{x}}\,'(t|t-1) = \mathbf{P}^*(t) \tag{26}$$

The expected quadratic loss is

$$\sum_{i=1}^{n} E\tilde{x}_i^{\,2}(t\,|\,t-1) \;=\; \mathrm{trace}\;\mathbf{P}^*(t) \tag{27}$$

The matrices $\mathbf{\Delta}^*(t)$, $\mathbf{\Phi}^*(t+1;\, t)$, $\mathbf{P}^*(t)$ *are generated by the recursion relations*

$$\left.\begin{aligned}\mathbf{\Delta}^*(t) &= \mathbf{\Phi}(t+1;\, t)\mathbf{P}^*(t)\mathbf{M}'(t)[\mathbf{M}(t)\mathbf{P}^*(t)\mathbf{M}'(t)]^{-1} && (28)\\ \mathbf{\Phi}^*(t+1;\, t) &= \mathbf{\Phi}(t+1;\, t) - \mathbf{\Delta}^*(t)\mathbf{M}(t) && (29)\\ \mathbf{P}^*(t+1) &= \mathbf{\Phi}^*(t+1;\, t)\mathbf{P}^*(t)\mathbf{\Phi}'(t+1;\, t) + \mathbf{Q}(t) && (30)\end{aligned}\right\}\; t \geq t_0$$

In order to carry out the iterations, one must specify the covariance $\mathbf{P}^*(t_0)$ *of* $\mathbf{x}(t_0)$ *and the covariance* $\mathbf{Q}(t)$ *of* $\mathbf{u}(t)$. *Finally, for any* $s \geq 0$, *if* $\mathbf{\Phi}$ is *invertible*

$$\begin{aligned}\mathbf{x}^*(t+s|t) &= \mathbf{\Phi}(t+s;\, \mathrm{t}+1)\mathbf{x}^*(t+1)|t)\\ &= \mathbf{\Phi}(t+s;\, \mathrm{t}+1)\mathbf{\Phi}^*(t+1;\, t)\mathbf{\Phi}(t;\, t+s-1)\\ &\qquad \times \mathbf{x}^*(t+s-1|t-1)\\ &\quad + \mathbf{\Phi}(t+s;\, t+1)\mathbf{\Delta}^*(t)\mathbf{y}(t)\end{aligned} \tag{31}$$

so that the estimate $\mathbf{x}^*(t+s|t)$ $(s \geq 0)$ is *also given by a linear dynamic system of the type* (21).

Remarks. (*h*) Eliminating $\mathbf{\Delta}^*$ and $\mathbf{\Phi}^*$ from (28–30), a nonlinear difference equation is obtained for $\mathbf{P}^*(t)$:

$$\begin{aligned}\mathbf{P}^*(t+1) \;=\; &\mathbf{\Phi}(t+1;\, t)\{\mathbf{P}^*(t) - \mathbf{P}^*(t)\mathbf{M}'(t)[\mathbf{M}(t)\mathbf{P}^*(t)\mathbf{M}'(t)]^{-1}\\ &\times \mathbf{P}^*(t)\mathbf{M}(t)\}\mathbf{\Phi}'(t+1;\, t) + \mathbf{Q}(t) \qquad t \geq t_0\end{aligned} \tag{32}$$

This equation is linear only if $\mathbf{M}(t)$ is invertible but then the problem is trivial since all components of the random vector $\mathbf{x}(t)$ are observable $\mathbf{P}^*(t+1) = \mathbf{Q}(t)$. Observe that equation (32) plays a role in the present theory analogous to that of the Wiener-Hopf equation in the conventional theory.

Once $\mathbf{P}^*(t)$ has been computed via (32) starting at $t = t_0$, the explicit specification of the optimal linear filter is immediately available from formulas (29-30). Of course, the solution of Equation (32), or of its differential-equation equivalent, is a much simpler task than solution of the Wiener-Hopf equation.

(*i*) The results stated in Theorem 3 do not resolve completely Problem I. Little has been said, for instance, about the physical significance of the assumptions needed to obtain equation (25), the convergence and stability of the nonlinear difference equation (32), the stability of the optimal filter (21), etc. This can actually be done in a completely satisfactory way, but must be left to a future paper. In this connection, the principal guide and

tool turns out to be the duality theorem mentioned briefly in the next section. See [29].

(*j*) By letting the sampling period (equal to one so far) approach zero, the method can be used to obtain the specification of a differential equation for the optimal filter. To do this, i.e., to pass from equation (14) to equation (12), requires computing the logarithm $\mathbf{F}^*$ of the matrix $\mathbf{\Phi}^*$. But this can be done only if $\mathbf{\Phi}^*$ is nonsingular—which is easily seen *not* to be the case. This is because it is sufficient for the optimal filter to have $n - p$ state variables, rather than n, as the formalism of equation (22) would seem to imply. By appropriate modifications, therefore, equation (22) can be reduced to an equivalent set of only $n - p$ equations whose transition matrix is nonsingular. Details of this type will be covered in later publications.

(*k*) The dynamic system (21) is, in general, nonstationary. This is due to two things: (1) The time dependence of $\mathbf{\Phi}(t + 1; t)$ and $\mathbf{M}(t)$; (2) the fact that the estimation starts at $t = t_0$ and improves as more data are accumulated. If $\mathbf{\Phi}$, $\mathbf{M}$ are constants, it can be shown that (21) becomes a stationary dynamic system in the limit $t \to \infty$. This is the case treated by the classical Wiener theory.

(*l*) It is noteworthy that the derivations given are not affected by the nonstationarity of the model for $\mathbf{x}(t)$ or the finiteness of available data. In fact, as far as the author is aware, the only explicit recursion relations given before for the growing-memory filter are due to Blum [12]. However, his results are much more complicated than ours.

(*m*) By inspection of Fig. 3 we see that the optimal filter is a feedback system, and that the signal after the first summer is white noise since $\tilde{\mathbf{y}}(t|t - 1)$ is obviously an orthogonal random process. This corresponds to some well-known results in Wiener filtering, see, e.g., Smith [28], Chapter 6, Fig. 6–4. However, this is apparently the first *rigorous* proof that every Wiener filter is realizable by means of a feedback system. Moreover, it will be shown in another paper that such a filter is always *stable,* under very mild assumptions on the model (16–17). See [29].

The Dual Problem

Let us now consider another problem which is conceptually very different from optimal estimation, namely, the noise-free regulator problem. In the simplest cases, this is:

Problem II. *Consider the dynamic system*

$$\mathbf{x}(t + 1) = \hat{\mathbf{\Phi}}(t + 1; t)\mathbf{x}(t) + \hat{\mathbf{M}}(t)\mathbf{u}(t) \tag{33}$$

where $\mathbf{x}(t)$ *is an n-vector,* $\mathbf{u}(t)$ *is an m-vector* ($m \le n$), $\hat{\mathbf{\Phi}}$, $\hat{\mathbf{M}}$ *are* $n \times n$ *resp.* $n \times m$ *matrices whose elements are nonrandom functions of time. Given any state* $\mathbf{x}(t)$ *at time t, we are to find a sequence* $\mathbf{u}(t)$, ..., $\mathbf{u}(T)$ *of control vectors which minimizes the performance index*

$$V[\mathbf{x}(t)] = \sum_{\tau=t}^{T+1} \mathbf{x}'(\tau)\mathbf{Q}(\tau)\mathbf{x}(\tau)$$

Where $\hat{\mathbf{Q}}(t)$ *is a positive definite matrix whose elements are nonrandom functions of time.* See Fig. 2, where $\mathbf{\Delta} = \hat{\mathbf{M}}$ and $\mathbf{M} = \mathbf{I}$.

Probabilistic considerations play no part in Problem II; it is implicitly assumed that every state variable can be measured exactly at each instant t, $t + 1$, ..., T. It is customary to call $T \ge t$ the *terminal time* (it may be infinity).

The first general solution of the noise-free regulator problem is due to the author [18]. The main result is that the optimal control vectors $\mathbf{u}^*(t)$ are nonstationary linear functions of $\mathbf{x}(t)$. After a change in notation, the formulas of the Appendix, Reference [18] (see also Reference [23]) are as follows:

$$\mathbf{u}^*(t) = -\hat{\mathbf{\Delta}}^*(t)\mathbf{x}(t) \tag{34}$$

Under optimal control as given by (34), the "closed-loop" equations for the system are (see Fig. 4)

$$\mathbf{x}(t + 1) = \hat{\mathbf{\Phi}}^*(t + 1; t)\mathbf{x}(t)$$

and the minimum performance index at time t is given by

$$V^*[\mathbf{x}(t)] = \mathbf{x}'(t)\mathbf{P}^*(t - 1)\mathbf{x}(t)$$

The matrices $\hat{\mathbf{\Delta}}^*(t)$, $\hat{\mathbf{\Phi}}^*(t + 1; t)$, $\hat{\mathbf{P}}^*(t)$ are determined by the recursion relations:

$$\left.\begin{aligned}
\hat{\mathbf{\Delta}}^*(t) &= [\hat{\mathbf{M}}'(t)\,\hat{\mathbf{P}}^*(t)\,\hat{\mathbf{M}}(t)]^{-1}\,\hat{\mathbf{M}}'(t)\,\hat{\mathbf{P}}^*(t)\,\hat{\mathbf{\Phi}}(t + 1; t) && (35)\\
\hat{\mathbf{\Phi}}^*(t + 1; t) &= \hat{\mathbf{\Phi}}(t + 1; t) - \hat{\mathbf{M}}(t)\,\hat{\mathbf{\Delta}}^*(t) && (36)\\
\hat{\mathbf{P}}^*(t - 1) &= \hat{\mathbf{\Phi}}'(t + 1; t)\,\hat{\mathbf{P}}^*(t)\,\hat{\mathbf{\Phi}}^*(t + 1; t) + \hat{\mathbf{Q}}(t) && (37)
\end{aligned}\right\} t \le T$$

Initially we must set $\hat{\mathbf{P}}^*(T) = \hat{\mathbf{Q}}(T + 1)$.

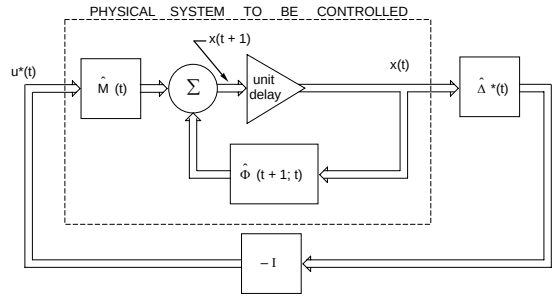


Fig. 4 Matrix block diagram of optimal controller

Comparing equations (35–37) with (28–30) and Fig. 3 with Fig. 4 we notice some interesting things which are expressed precisely by

Theorem 4. (*Duality Theorem*) *Problem I and Problem II are duals of each other in the following sense:*

Let $\tau \ge 0$. *Replace every matrix* $\mathbf{X}(t) = \mathbf{X}(t_0 + \tau)$ *in* (28–30) *by* $\hat{\mathbf{X}}'(t) = \hat{\mathbf{X}}'(T - \tau)$. *Then One has* (35–37). *Conversely, replace every matrix* $\hat{\mathbf{X}}(T - \tau)$ *in* (35–37) *by* $\mathbf{X}'(t_0 + \tau)$. *Then one has* (28–30).

Proof. Carry out the substitutions. For ease of reference, the dualities between the two problems are given in detail in Table 1.

Table 1

	Problem I	Problem II
1	$\mathbf{x}(t)$ (unobservable) state variables of random process.	$\mathbf{x}(t)$ (observable) state variables of plant to be regulated.
2	$\mathbf{y}(t)$ observed random variables.	$\mathbf{u}(t)$ control variables
3	t_0 first observation.	T last control action.
4	$\mathbf{\Phi}(t_0 + \tau + 1; t_0 + \tau)$ transition matrix.	$\hat{\mathbf{\Phi}}(T - \tau + 1; T - \tau)$ transition matrix.
5	$\mathbf{P}^*(t_0 + \tau)$ covariance of optimized estimation error.	$\hat{\mathbf{P}}^*(T - \tau)$ matrix of quadratic form for performance index under optimal regulation.
6	$\mathbf{\Delta}^*(t_0 + \tau)$ weighting of observation for optimal estimation.	$\hat{\mathbf{\Delta}}^*(T - \tau)$ weighting of state for optimal control.
7	$\mathbf{\Phi}^*(t_0 + \tau + 1; t_0 + \tau)$ transition matrix for optimal estimation error.	$\hat{\mathbf{\Phi}}^*(T - \tau + 1; T - \tau)$ transition matrix under optimal regulation.
8	$\mathbf{M}(t_0 + \tau)$ effect of state on observation.	$\hat{\mathbf{M}}(T - \tau)$ effect of control vectors on state.
9	$\mathbf{Q}(t_0 + \tau)$ covariance of random excitation.	$\hat{\mathbf{Q}}(T - \tau)$ matrix of quadratic form defining error criterion.

Remarks. (*n*) The *mathematical* significance of the duality between Problem I and Problem II is that both problems reduce to the solution of the Wiener-Hopf-like equation (32).

(*o*) The *physical* significance of the duality is intriguing. Why are observations and control dual quantities?

Recent research [29] has shown that the essence of the Duality Theorem lies in the duality of constraints at the output (represented by the matrix $\hat{\mathbf{M}}(t)$ in Problem I) and constraints at the input (represented by the matrix $\hat{\mathbf{M}}(t)$ in Problem II).

(*p*) Applications of Wiener's methods to the solution of noise-free regulator problem have been known for a long time; see the recent textbook of Newton, Gould, and Kaiser [27]. However, the connections between the two problems, and in particular the duality, have apparently never been stated precisely before.

(*q*) The duality theorem offers a powerful tool for developing more deeply the theory (as opposed to the computation) of Wiener filters, as mentioned in Remark (*i*). This will be published elsewhere [29].

Applications

The power of the new approach to the Wiener problem, as expressed by Theorem 3, is most obvious when the data of the problem are given in numerical form. In that case, one simply performs the numerical computations required by (28–30). Resuits of such calculations, in some cases of practical engineering interest, will be published elsewhere.

When the answers are desired in closed analytic form, the iterations (28–30) may lead to very unwieldy expressions. In a few cases, $\mathbf{\Delta}^*$ and $\mathbf{\Phi}^*$ can be put into "closed form." Without discussing here how (if at all) such closed forms can be obtained, we now give two examples indicative of the type of results to be expected.

Example 1. Consider the problem mentioned under "Optimal Estimates." Let $x_1(t)$ be the signal and $x_2(t)$ the noise. We assume the model:

$$x_1(t+1) = \phi_{11}(t+1;\, t)x_1(t) + u_1(t)$$

$$x_2(t+1) = u_2(t)$$

$$y_1(t) = x_1(t) + x_2(t)$$

The specific data for which we desire a solution of the estimation problem are as follows:

1 $t_1 = t + 1;\ t_0 = 0$
2 $Ex_1^2(0) = 0$, i.e., $x_1(0) = 0$
3 $Eu_1^2(t) = a^2$, $Eu^2(t) = b^2$, $Eu_1(t)\, u_2(t) = 0$ (for all t)
4 $\phi_{11}(t+1;\, t) = \phi_{11} = \text{const.}$

A simple calculation shows that the following matrices satisfy the difference equations (28–30), for all $t \geq t_0$:

$$\mathbf{\Delta}^*(t) = \begin{bmatrix} \phi_{11}C(t) \\ 0 \end{bmatrix}$$

$$\mathbf{\Phi}^*(t+1;\, t) = \begin{bmatrix} \phi_{11}[1-C(t)] & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{P}^*(t+1) = \begin{bmatrix} a^2 + \phi_{11}^2 b^2 C(t) & 0 \\ 0 & b^2 \end{bmatrix}$$

where $$C(t+1) = 1 - \frac{b^2}{a^2 + b^2 + \phi_{11}^2 b^2 C(t)} \qquad t \geq 0 \tag{38}$$

Since it was assumed that $x_1(0) = 0$, neither $x_1(1)$ nor $x_2(1)$ can be predicted from the measurement of $y_1(0)$. Hence the measurement at time $t = 0$ is useless, which shows that we should set $C(0) = 0$. This fact, with the iterations (38), completely determines the function $C(t)$. The nonlinear difference equation (38) plays the role of the Wiener-Hopf equation.

If $b^2/a^2 \ll 1$, then $C(t) \approx 1$ which is essentially pure prediction. If $b^2/a^2 \gg 1$, then $C(t) \approx 0$, and we depend mainly on $x_1^*(t|t-1)$ for the estimation of $x_1^*(t+1|t)$ and assign only very small weight to the measurement $y_1(t)$; this is what one would expect when the measured data are very noisy.

In any case, $x_2^*(t|t-1) = 0$ at all times; one cannot predict independent noise! This means that ϕ^*_{12} can be set equal to zero. The optimal predictor is a first-order dynamic system. See Remark (*j*).

To find the stationary Wiener filter, let $t = \infty$ on both sides of (38), solve the resulting quadratic equation in $C(\infty)$, etc.

Example 2. A number or particles leave the origin at time $t_0 = 0$ with random velocities; after $t = 0$, each particle moves with a constant (unknown) velocity. Suppose that the position of one of these particles is measured, the data being contaminated by stationary, additive, correlated noise. What is the optimal estimate of the position and velocity of the particle at the time of the last measurement?

Let $x_1(t)$ be the position and $x_2(t)$ the velocity of the particle; $x_3(t)$ is the noise. The problem is then represented by the model,

$$x_1(t+1) = x_1(t) + x_2(t)$$
$$x_2(t+1) = x_2(t)$$
$$x_3(t+1) = \phi_{33}(t+1;\, t)x_3(t) + u_3(t)$$
$$y_1(t) = x_1(t) + x_3(t)$$

and the additional conditions

1 $t_1 = t;\ t_0 = 0$
2 $Ex_1^2(0) = Ex_2(0) = 0$, $Ex_2^2(0) = a^2 > 0$;
3 $Eu_3(t) = 0$, $Eu_3^2(t) = b^2$.
4 $\phi_{33}(t+1;\, t) = \phi_{33} = \text{const.}$

According to Theorem 3, $\mathbf{x}^*(t|t)$ is calculated using the dynamic system (31).

First we solve the problem of predicting the position and velocity of the particle one step ahead. Simple considerations show that

$$\mathbf{P}^*(1) = \begin{bmatrix} a^2 & a^2 & 0 \\ a^2 & a^2 & 0 \\ 0 & 0 & b^2 \end{bmatrix} \quad \text{and} \quad \mathbf{\Delta}^*(0) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

It is then easy to check by substitution into equations (28–30) that

$$\mathbf{P}^*(t) = \frac{b^2}{C_1(t-1)} \times \begin{bmatrix} t^2 & t & -\phi_{33}t(t-1) \\ t & 1 & -\phi_{33}(t-1) \\ -\phi_{33}t(t-1) & -\phi_{33}(t-1) & \phi_{33}^2(t-1)^2 + C_1(t-1) \end{bmatrix}$$

is the correct expression for the covariance matrix of the prediction error $\tilde{\mathbf{x}}(t|t-1)$ for all $t \geq 1$, provided that we define

$$C_1(0) = b^2/a^2$$
$$C_1(t) = C_1(t-1) + [t - \phi_{33}(t-1)]^2, \quad t \geq 1$$

It is interesting to note that the results just obtained are valid also when ϕ_{33} depends on t. This is true also in Example 1. In conventional treatments of such problems there *seems* to be an essential difference between the cases of stationary and nonstationary noise. This misleading impression created by the conventional theory is due to the very special *methods* used in solving the Wiener-Hopf equation.

Introducing the abbreviation

$$C_2(0) = 0$$
$$C_2(t) = t - \phi_{33}(t-1), \quad t \geq 1$$

and observing that

$$\text{cov}\ \tilde{\mathbf{x}}(t+1|t) = \mathbf{P}^*(t+1)$$
$$= \mathbf{\Phi}(t+1;\, t)[\text{cov}\ \tilde{\mathbf{x}}(t|t)]\mathbf{\Phi}'(t+1;\, t) + \mathbf{Q}(t)$$

the matrices occurring in equation (31) and the covariance matrix of $\tilde{\mathbf{x}}(t|t)$ are found after simple calculations. We have, for all $t \geq 0$,

$$\mathbf{\Phi}(t;\, t+1)\mathbf{\Delta}^*(t) = \frac{1}{C_1(t)}\begin{bmatrix} tC_2(t) \\ C_2(t) \\ C_1(t) - tC_2(t) \end{bmatrix}$$

$$\mathbf{\Phi}(t;\, t+1)\mathbf{\Phi}^*(t+1;\, t)\mathbf{\Phi}(t+1;\, t)$$

$$= \frac{1}{C_1(t)}\begin{bmatrix} C_1(t) - tC_2(t) & C_1(t) - tC_3(t) & -\phi_{33}tC_2(t) \\ -C_2(t) & C_1(t) - C_2(t) & -\phi_{33}C_2(t) \\ -C_1(t) + tC_2(t) & -C_1(t) + tC_2(t) & +\phi_{33}tC_2(t) \end{bmatrix}$$

and

$$\operatorname{cov}\ \tilde{\mathbf{x}}(t|t) = E\,\tilde{\mathbf{x}}(t|t)\,\tilde{\mathbf{x}}'(t|t) = \frac{b^2}{C_1(t)}\begin{bmatrix} t^2 & t & -t^2 \\ t & 1 & -t \\ -t^2 & -t & t^2 \end{bmatrix}$$

To gain some insight into the behavior of this system, let us examine the limiting case $t \to \infty$ of a large number of observations. Then $C_1(t)$ obeys approximately the differential equation

$$dC_1(t)/dt \approx C_2^2(t) \qquad (t >> 1)$$

from which we find

$$C_1(t) \approx (1 - \phi_{33})^2 t^3/3 + \phi_{33}(1 - \phi_{33})t^2 + \phi_{33}^2 t + b^2/a^2 \qquad (t >> 1) \qquad (39)$$

Using (39), we get further,

$$\mathbf{\Phi}^{-1}\mathbf{\Phi}^*\mathbf{\Phi} \approx \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ -1 & -1 & 0 \end{bmatrix} \text{ and } \mathbf{\Phi}^{-1}\mathbf{\Delta}^* \approx \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad (t >> 1)$$

Thus as the number of observations becomes large, we depend almost exclusively on $x_1^*(t|t)$ and $x_2^*(t|t)$ to estimate $x_1^*(t+1|t+1)$ and $x_2^*(t+1|t+1)$. Current observations are used almost exclusively to estimate the noise

$$x_3^*(t|t) \approx y_1^*(t) - x_1^*(t|t) \qquad (t >> 1)$$

One would of course expect something like this since the problem is analogous to fitting a straight line to an increasing number of points.

As a second check on the reasonableness of the results given, observe that the case $t >> 1$ is essentially the same as prediction based on continuous observations. Setting $\phi_{33} = 0$, we have

$$E\,\tilde{x}_1^{\,2}(t\,|\,t) \approx \frac{a^2b^2t^2}{b^2 + a^2t^3/3} \qquad (t >> 1;\ \phi_{33} = 0)$$

which is identical with the result obtained by Shinbrot [11], Example 1, and Solodovnikov [14], Example 2, in their treatment of the Wiener problem in the finite-length, continuous-data case, using an approach entirely different from ours.

Conclusions

This paper formulates and solves the Wiener problem from the "state" point of view. On the one hand, this leads to a very general treatment including cases which cause difficulties when attacked by other methods. On the other hand, the Wiener problem is shown to be closely connected with other problems in the theory of control. Much remains to be done to exploit these connections.

References

1 N. Wiener, "The Extrapolation, Interpolation and Smoothing of Stationary Time Series," John Wiley & Sons, Inc., New York, N.Y., 1949.

2 L. A. Zadeh and J. R. Ragazzini, "An Extension of Wiener's Theory of Prediction," *Journal of Applied Physics*, vol. 21, 1950, pp. 645–655.

3 H. W. Bode and C. E. Shannon, "A Simplified Derivation of Linear Least-Squares Smoothing and Prediction Theory," *Proceedinqs IRE*, vol. 38, 1950, pp. 417–425.

4 R. C. Booton, "An Optimization Theory for Time-Varying Linear Systems With Nonstationary Statistical Inputs," *Proceedings IRE*, vol. 40, 1952, pp. 977–981.

5 J. H. Laning and R. H. Battin, "Random Processes in Automatic Control," McGraw-Hill Book Company, Inc., New York, N. Y., 1956.

6 W. B. Davenport, Jr., and W. L. Root, "An Introduction to the Theory of Random Signals and Noise," McGraw-Hill Book Company, Inc., New York, N. Y., 1958.

7 S. Darlington, "Linear Least-Squares Smoothing and Prediction, With Applications," *Bell System Tech. Journal*, vol. 37, 1958, pp.1221–1294.

8 G. Franklin, "The Optimum Synthesis of Sampled-Data Systems" Doctoral dissertation, Dept. of Elect. Engr., Columbia University, 1955.

9 A. B. Lees, "Interpolation and Extrapolation of Sampled Data," *Trans. IRE Prof. Grou.p on Information Theory*, IT-2, 1956, pp. 173–175.

10 R. C. Davis. "On the Theory of Prediction of Nonstationary Stochastic Processes," *Journal of Applied Physics*, vol. 23. 1952, pp. 1047–1053.

11 M. Shinbrot, "Optimization of Tirne-Varying Linear Systems With Nonstationary Inputs," TRANS. ASME, vol. 80, 1958. pp. 457–462.

12 M. Blum, "Recursion Formulas for Growing Memory Digital Filters," *Trans. IRE Prof. Group on Information Theory*. IT-4. 1958, pp. 24–30.

13 V. S. Pugachev, "The Use of Canonical Expansions of Random Functions in Determining an Optimum Linear System," *Automatics and Remote Control* (USSR), vol. 17, 1956, pp. 489–499; translation pp. 545–556.

14 V. V. Solodovnikov and A. M. Batkov. "On the Theory of Self-Optimizing Systems (in German and Russian)," Proc. Heidelberg Conference on Automatic Control, 1956. pp. 308–323.

15 J. L. Doob, "Stochastic Processes," John Wiley & Sons, Inc., New York, N. Y., 1955.

16 M. Loève, "Probability Theory," Van Nostrand Company, Inc., New York, N. Y., 1955.

17 R. E. Bellman, I. Glicksberg, and 0. A. Gross, "Some Aspects of the Mathematical Theory of Control Processes," RAND Report R-313, 1958, 244 pp.

18 R. E. Kalman and R. W. Koepcke, "Optimal Synthesis of Linear Sampling Control Systems Using Generalized Performance Indexes," TRANS. ASME, vol. 80, 1958, pp. 1820–1826.

19 J. E. Bertram, "Effect of Quantization in Sampled-Feedback Systems," *Trans. AlEE*, vol. 77, II, 1958, pp. 177–182.

20 R. E. Kalman and J. E. Bertram, "General Synthesis Procedure for Computer Control of Single and Multi-Loop Linear Systems" *Trans. AlEE*, vol. 77, II, 1958, pp. 602–609.

21 C. W. Merriam, III, "A Class of Optimum Control Systems," *Journal of the Franklin Institute*, vol. 267, 1959, pp. 267–281.

22 R. E. Kalman and J. E. Bertram, "A Unified Approach to the Theory of Sampling Systems," *Journal of the Franklin Institute*, vol. 267, 1959, pp. 405–436.

23 R. E. Kalman and R. W. Koepcke, "The Role of Digital Computers in the Dynamic Optimization of Chemical Reactors," Proc. Western Joint Computer Conference, 1959, pp. 107–116.

24 R. E. Kalman, "Dynamic Optimization of Linear Control Systems, I. Theory," to appear.

25 S. Sherman, "Non-Mean-Square Error Criteria," *Trans. IRE Prof. Group on Information Theory*, IT-4, 1958, pp. 125–126.

26 V. S. Pugachev, "On a Possible General Solution of the Problem of Determining Optimum Dynamic Systems," *Automatics and Remote Control* (USSR), vol. 17, 1956, pp. 585–589.

27 G. C. Newton, Jr., L. A. Gould, and J. F. Kaiser, "Analytical Design of Linear Feedback Controls," John Wiley & Sons, Inc., New York, N. Y., 1957.

28 0. J. M. Smith, "Feedback Control Systems," McGraw-Hill Book Company, Inc., New York, N. Y., 1958.

29 R. E. Kalman, "On the General Theory of Control Systems," Proceedings First International Conference on Automatic Control, Moscow, USSR, 1960.

APPENDIX
RANDOM PROCESSES: BASIC CONCEPTS

For convenience of the reader, we review here some elementary definitions and facts about probability and random processes. Everything is presented with the utmost possible simplicity; for greater depth and breadth, consult Laning and Battin [5] or Doob [15].

A *random variable* is a function whose values depend on the outcome of a chance event. The *values* of a random variable may be any convenient mathematical entities; real or complex numbers, vectors, etc. For simplicity, we shall consider here only real-valued random variables, but this is no real restriction. Random variables will be denoted by $x, y, \ldots$ and their values by ξ, η, …. Sums, products, and functions of random variables are also random variables.

A random variable x can be explicitly defined by stating the probability that x is less than or equal to some real constant ξ. This is expressed symbolically by writing

$$Pr(x \le \xi) = F_x(\xi);\ F_x(-\infty) = 0,\ F_x(+\infty) = 1$$

$F_x(\xi)$ is called the *probability distribution function* of the random variable x. When $F_x(\xi)$ is differentiable with respect to ξ, then $f_x(\xi) = dF_x(\xi)/d\xi$ is called the *probability density function* of x.

The *expected value* (*mathematical expectation*, *statistical average*, *ensemble average*, *mean*, etc., are commonly used synonyms) of any nonrandom function $g(x)$ of a random variable x is defined by

$$Eg(x) = E[g(x)] = \int_{-\infty}^{\infty} g(\xi)dF_x(\xi) = \int_{-\infty}^{\infty} g(\xi) f_x(\xi)d\xi \qquad (40)$$

As indicated, it is often convenient to omit the brackets after the symbol E. A sequence of random variables (finite or infinite)

$$\{x(t)\} = \ldots, x(-1), x(0), x(1), \ldots \qquad (41)$$

is called a *discrete* (or *discrete-parameter*) *random* (or *stochastic*) *process*. One particular set of observed values of the random process (41)

$$\ldots, \xi(-1), \xi(0), \xi(1), \ldots$$

is called a *realization* (or a *sample function)* of the process. Intuitively, a random process is simply a set of random variables which are indexed in such a way as to bring the notion of time into the picture.

A random process is *uncorrelated* if

$$Ex(t)x(s) = Ex(t)Ex(s) \qquad (t \ne s)$$

If, furthermore,

$$Ex(t)x(s) = 0 \qquad (t \ne s)$$

then the random process is *orthogonal*. Any uncorrelated random process can be changed into orthogonal random process by replacing $x(t)$ by $x'(t) = x(t) - Ex(t)$ since then

$$Ex'(t)x'(s) = E[x(t) - Ex(t)]\cdot[x(s) - Ex(s)] = Ex(t)x(s) - Ex(t)Ex(s) = 0$$

It is useful to remember that, if a random process is orthogonal, then

$$E[x(t_1) + x(t_2) + \ldots]^2 = Ex^2(t_1) + Ex^2(t_2) + \ldots \ (t_1 \ne t_2 \ne \ldots)$$

If $\mathbf{x}$ is a vector-valued random variable with components $x_1, \ldots, x_n$ (which are of course random variables), the matrix

$$[E(x_i - Ex_i)(x_j - Ex_j)] = E(\mathbf{x} - E\mathbf{x})(\mathbf{x}' - E\mathbf{x}') = \operatorname{cov} \mathbf{x} \qquad (42)$$

is called the *covariance matrix of x*.

A random process may be specified explicitly by stating the probability of simultaneous occurrence of any finite number of events of the type
$x(t_1) \le \xi_1, \ldots, x(t_n) \le \xi_n$; $(t_1 \ne \ldots \ne t_n)$, i.e.,

$$Pr[(x(t_1) \le \xi_1, \ldots, x(t_n) \le \xi_n)] = F_{x(t_1), \ldots, x(t_n)}(\xi_1, \ldots, \xi_n) \qquad (43)$$

where $F_{x(t_1), \ldots, x(t_n)}$ is called the *joint probability distribution function* of the random variables $x(t_1), \ldots, x(t_n)$. The *joint probability density function* is then

$$f_{x(t_1), \ldots, x(t_n)}(\xi_1, \ldots, \xi_n) = \partial^n F_{n(t_1), \ldots, x(t_n)}/\partial\xi_1, \ldots, \partial\xi_n$$

provided the required derivatives exist. The expected value $Eg[x(t_1), \ldots, x(t_n)]$ of any nonrandom function of n random variables is defined by an n-fold integral analogous to (40).

A random process is *independent* if for any finite $t_1 \ne \ldots \ne t_n$, (43) is equal to the product of the first-order distributions

$$Pr[x(t_1) \le \xi_1] \ldots Pr[x(t_n) \le \xi_n]$$

If a set of random variables is independent, then they are obviously also uncorrelated. The converse is not true in general. For a set of more than 2 random variables to be independent, it is not sufficient that any pair of random variables be independent.

Frequently it is of interest to consider the probability distribution of a random variable $x(t_{n+1})$ of a random process given the actual values $\xi(t_1), \ldots, \xi(t_n)$ with which the random variables $x(t_1), \ldots, x(t_n)$ have occurred. This is denoted by

$$Pr[x(t_{n+1}) \le \xi_{n+1} | x(t_1) = \xi_1, \ldots, x(t_n) = \xi_n] = \frac{\int_{-\infty}^{\xi_{n+1}} f_{x(t_1),\ldots,x(t_{n+1})}(\xi_1,\ldots,\xi_{n+1})d\xi_{n+1}}{f_{x(t_1),\ldots,x(t_n)}(\xi_1,\ldots,\xi_n)} \qquad (44)$$

which is called the *conditional probability distribution function* of $x(t_{n+1})$ given $x(t_1), \ldots, x(t_n)$. The *conditional expectation*

$$E\{g[x(t_{n+1})]|x(t_1), \ldots, x(t_n)\}$$

is defined analogously to (40). The conditional expectation is a random variable; it follows that

$$E[E\{g[x(t_{n+1})]|x(t_1), \ldots, x(t_n)\}] = E\{g[x(t_{n+1})]\}$$

In all cases of interest in this paper, integrals of the type (40) or (44) need never be evaluated explicitly, only the *concept* of the expected value is needed.

A random variable x is *gaussian* (or *normally distributed)* if

$$f_x(\xi) = \frac{1}{[2\pi E(x - Ex)^2]^{1/2}} \exp\left[-\frac{1}{2}\frac{(\xi - Ex)^2}{E(x - Ex)^2}\right]$$

which is the well-known bell-shaped curve. Similarly, a random vector $\mathbf{x}$ is *gaussian* if

$$f_x(\boldsymbol{\xi}) = \frac{1}{(2\pi)^{n/2}(det\ \mathbf{C})^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{\xi} - E\mathbf{x})'\mathbf{C}^{-1}(\boldsymbol{\xi} - E\mathbf{x})\right]$$

where $\mathbf{C}^{-1}$ is the inverse of the covariance matrix (42) of $\mathbf{x}$. A *gaussian random process* is defined similarly.

The importance of gaussian random variables and processes is largely due to the following facts:

Theorem 5. (*A*) *Linear functions* (*and therefore conditional expectations*) *on a gaussian random process are gaussian random variables.*

(*B*) *Orthogonal gaussian random variables are independent.*

(*C*) *Given any random process with means* $Ex(t)$ *and covariances* $Ex(t)x(s)$, *there exists a unique gaussian random process with the same means and covariances.*

Transactions of the ASME–Journal of Basic Engineering, 82 (Series D): 35-45. Copyright © 1960 by ASME

Explanation of this transcription, John Lukesh, 20 January 2002.

Using a photo copy of R. E. Kalman's 1960 paper from an original of the ASME "Journal of Basic Engineering", March 1960 issue, I did my best to make an accurate version of this rather significant piece, in an up-to-date computer file format. For this I was able to choose page formatting and type font spacings that resulted in a document that is a close match to the original. (All pages start and stop at about the same point, for example; even most individual lines of text do.) I used a recent version of Word for Windows and a recent Hewlett Packard scanner with OCR (optical character recognition) software. The OCR software is very good on plain text, even distinguishing between italic versus regular characters quite reliably, but it does not do well with subscripts, superscripts, and special fonts, which were quite prevalent in the original paper. And I found there was no point in trying to work from the OCR results for equations. A lot of manual labor was involved.

Since I wanted to make a faithful reproduction of the original, I did not make any changes to correct (what I believed were) mistakes in it. For example, equation (32) has a $\mathbf{P}^*(t)\mathbf{M}(t)$ product that should be reversed, I think. I left this, and some other things that I thought were mistakes in the original, as is. (I didn't find very many other problems with the original.) There may, *of course*, be problems with my transcription. The plain text OCR results, which didn't require much editing, are pretty accurate I think. But the subscripts etc and the equations which I copied essentially manually, are suspect. I've reviewed the resulting document quite carefully, several times finding mistakes in what I did each time. The last time there were five, four cosmetic and one fairly inconsequential. There are probably more. I would be very pleased to know about it if any reader of this finds some of them; jlukesh@deltanet.com.

Deterministic Nonperiodic Flow[1]

EDWARD N. LORENZ

Massachusetts Institute of Technology

(Manuscript received 18 November 1962, in revised form 7 January 1963)

ABSTRACT

Finite systems of deterministic ordinary nonlinear differential equations may be designed to represent forced dissipative hydrodynamic flow. Solutions of these equations can be identified with trajectories in phase space. For those systems with bounded solutions, it is found that nonperiodic solutions are ordinarily unstable with respect to small modifications, so that slightly differing initial states can evolve into considerably different states. Systems with bounded solutions are shown to possess bounded numerical solutions.

A simple system representing cellular convection is solved numerically. All of the solutions are found to be unstable, and almost all of them are nonperiodic.

The feasibility of very-long-range weather prediction is examined in the light of these results.

1. Introduction

Certain hydrodynamical systems exhibit steady-state flow patterns, while others oscillate in a regular periodic fashion. Still others vary in an irregular, seemingly haphazard manner, and, even when observed for long periods of time, do not appear to repeat their previous history.

These modes of behavior may all be observed in the familiar rotating-basin experiments, described by Fultz, *et al.* (1959) and Hide (1958). In these experiments, a cylindrical vessel containing water is rotated about its axis, and is heated near its rim and cooled near its center in a steady symmetrical fashion. Under certain conditions the resulting flow is as symmetric and steady as the heating which gives rise to it. Under different conditions a system of regularly spaced waves develops, and progresses at a uniform speed without changing its shape. Under still different conditions an irregular flow pattern forms, and moves and changes its shape in an irregular nonperiodic manner.

Lack of periodicity is very common in natural systems, and is one of the distinguishing features of turbulent flow. Because instantaneous turbulent flow patterns are so irregular, attention is often confined to the statistics of turbulence, which, in contrast to the details of turbulence, often behave in a regular well-organized manner. The short-range weather forecaster, however, is forced willy-nilly to predict the details of the large-scale turbulent eddies—the cyclones and anticyclones—which continually arrange themselves into new patterns. Thus there are occasions when more than the statistics of irregular flow are of very real concern.

In this study we shall work with systems of deterministic equations which are idealizations of hydrodynamical systems. We shall be interested principally in nonperiodic solutions, i.e., solutions which never repeat their past history exactly, and where all approximate repetitions are of finite duration. Thus we shall be involved with the ultimate behavior of the solutions, as opposed to the transient behavior associated with arbitrary initial conditions.

A closed hydrodynamical system of finite mass may ostensibly be treated mathematically as a finite collection of molecules—usually a very large finite collection—in which case the governing laws are expressible as a finite set of ordinary differential equations. These equations are generally highly intractable, and the set of molecules is usually approximated by a continuous distribution of mass. The governing laws are then expressed as a set of partial differential equations, containing such quantities as velocity, density, and pressure as dependent variables.

It is sometimes possible to obtain particular solutions of these equations analytically, especially when the solutions are periodic or invariant with time, and, indeed, much work has been devoted to obtaining such solutions by one scheme or another. Ordinarily, however, nonperiodic solutions cannot readily be determined except by numerical procedures. Such procedures involve replacing the continuous variables by a new finite set of functions of time, which may perhaps be the values of the continuous variables at a chosen grid of points, or the coefficients in the expansions of these variables in series of orthogonal functions. The governing laws then become a finite set of ordinary differential

[1] The research reported in this work has been sponsored by the Geophysics Research Directorate of the Air Force Cambridge Research Center, under Contract No. AF 19(604)-4969.

equations again, although a far simpler set than the one which governs individual molecular motions.

In any real hydrodynamical system, viscous dissipation is always occurring, unless the system is moving as a solid, and thermal dissipation is always occurring, unless the system is at constant temperature. For certain purposes many systems may be treated as conservative systems, in which the total energy, or some other quantity, does not vary with time. In seeking the ultimate behavior of a system, the use of conservative equations is unsatisfactory, since the ultimate value of any conservative quantity would then have to equal the arbitrarily chosen initial value. This difficulty may be obviated by including the dissipative processes, thereby making the equations nonconservative, and also including external mechanical or thermal forcing, thus preventing the system from ultimately reaching a state of rest. If the system is to be deterministic, the forcing functions, if not constant with time, must themselves vary according to some deterministic rule.

In this work, then, we shall deal specifically with finite systems of deterministic ordinary differential equations, designed to represent forced dissipative hydrodynamical systems. We shall study the properties of nonperiodic solutions of these equations.

It is not obvious that such solutions can exist at all. Indeed, in dissipative systems governed by finite sets of *linear* equations, a constant forcing leads ultimately to a constant response, while a periodic forcing leads to a periodic response. Hence, nonperiodic flow has sometimes been regarded as the result of nonperiodic or random forcing.

The reasoning leading to these concludions is not applicable when the governing equations are nonlinear. If the equations contain terms representing advection—the transport of some property of a fluid by the motion of the fluid itself—a constant forcing can lead to a variable response. In the rotating-basin experiments already mentioned, both periodic and nonperiodic flow result from thermal forcing which, within the limits of experimental control, is constant. Exact periodic solutions of simplified systems of equations, representing dissipative flow with constant thermal forcing, have been obtained analytically by the writer (1962a). The writer (1962b) has also found nonperiodic solutions of similar systems of equations by numerical means.

2. Phase space

Consider a system whose state may be described by M variables $X_1, \cdots, X_M$. Let the system be governed by the set of equations

$$dX_i/dt = F_i(X_1, \cdots X_M), \quad i=1, \cdots, M, \tag{1}$$

where time t is the single independent variable, and the functions F_i possess continuous first partial derivatives. Such a system may be studied by means of *phase space*—an M-dimensional Euclidean space Γ whose coordinates are $X_1, \cdots, X_M$. Each *point* in phase space represents a possible instantaneous state of the system. A state which is varying in accordance with (1) is represented by a moving *particle* in phase space, traveling along a *trajectory* in phase space. For completeness, the position of a stationary particle, representing a steady state, is included as a trajectory.

Phase space has been a useful concept in treating finite systems, and has been used by such mathematicians as Gibbs (1902) in his development of statistical mechanics, Poincaré (1881) in his treatment of the solutions of differential equations, and Birkhoff (1927) in his treatise on dynamical systems.

From the theory of differential equations (e,g., Ford 1933, ch. 6), it follows, since the partial derivatives $\partial F_i/\partial X_j$ are continuous, that if t_0 is any time, and if $X_{10}, \cdots X_{M0}$ is any point in Γ, equations (1) possess a unique solution

$$X_i = f_i(X_{10}, \cdots, X_{M0}, t), \quad i=1, \cdots, M, \tag{2}$$

valid throughout some time interval containing t_0, and satisfying the condition

$$f_i(X_{10}, \cdots, X_{M0}, t_0) = X_{i0}, \quad i=1, \cdots, M. \tag{3}$$

The functions f_i are continuous in $X_{10}, \cdots, X_{M0}$ and t. Hence there is a unique trajectory through each point of Γ. Two or more trajectories may, however, approach the same point or the same curve asymptotically as $t \to \infty$ or as $t \to -\infty$. Moreover, since the functions f_i are continuous, the passage of time defines a continuous deformation of any region of Γ into another region.

In the familiar case of a conservative system, where some positive definite quantity Q, which may represent some form of energy, is invariant with time, each trajectory is confined to one or another of the surfaces of constant Q. These surfaces may take the form of closed concentric shells.

If, on the other hand, there is dissipation and forcing, and if, whenever Q equals or exceeds some fixed value Q_1, the dissipation acts to diminish Q more rapidly then the forcing can increase Q, then $(-dQ/dt)$ has a positive lower bound where $Q \geqq Q_1$, and each trajectory must ultimately become trapped in the region where $Q < Q_1$. Trajectories representing forced dissipative flow may therefore differ considerably from those representing conservative flow.

Forced dissipative systems of this sort are typified by the system

$$dX_i/dt = \sum_{j,k} a_{ijk} X_j X_k - \sum_j b_{ij} X_j + c_i, \tag{4}$$

where $\sum a_{ijk} X_i X_j X_k$ vanishes identically, $\sum b_{ij} X_i X_j$ is positive definite, and $c_1, \cdots, c_M$ are constants. If

$$Q = \tfrac{1}{2} \sum_i X_i^2, \tag{5}$$

and if $e_1, \cdots, e_M$ are the roots of the equations

$$\sum_j (b_{ij}+b_{ji})e_j=c_i, \tag{6}$$

it follows from (4) that

$$dQ/dt=\sum_{i,j} b_{ij}e_ie_j-\sum_{i,j} b_{ij}(X_i-e_i)(X_j-e_j). \tag{7}$$

The right side of (7) vanishes only on the surface of an ellipsoid E, and is positive only in the interior of E. The surfaces of constant Q are concentric spheres. If S denotes a particular one of these spheres whose interior R contains the ellipsoid E, it is evident that each trajectory eventually becomes trapped within R.

3. The instability of nonperiodic flow

In this section we shall establish one of the most important properties of deterministic nonperiodic flow, namely, its instability with respect to modifications of small amplitude. We shall find it convenient to do this by identifying the solutions of the governing equations with trajectories in phase space. We shall use such symbols as $P(t)$ (variable argument) to denote trajectories, and such symbols as P or $P(t_0)$ (no argument or constant argument) to denote points, the latter symbol denoting the specific point through which $P(t)$ passes at time t_0.

We shall deal with a phase space Γ in which a unique trajectory passes through each point, and where the passage of time defines a continuous deformation of any region of Γ into another region, so that if the points $P_1(t_0)$, $P_2(t_0)$, $\cdots$approach $P_0(t_0)$ as a limit, the points $P_1(t_0+\tau)$, $P_2(t_0+\tau)$, $\cdots$must approach $P_0(t_0+\tau)$ as a limit. We shall furthermore require that the trajectories be uniformly bounded as $t\rightarrow\infty$; that is, there must be a bounded region R, such that every trajectory ultimately remains with R. Our procedure is influenced by the work of Birkhoff (1927) on dynamical systems, but differs in that Birkhoff was concerned mainly with conservative systems. A rather detailed treatment of dynamical systems has been given by Nemytskii and Stepanov (1960), and rigorous proofs of some of the theorems which we shall present are to be found in that source.

We shall first classify the trajectories in three different manners, namely, according to the absence or presence of transient properties, according to the stability or instability of the trajectories with respect to small modifications, and according to the presence or absence of periodic behavior.

Since any trajectory $P(t)$ is bounded, it must possess at least one *limit point* P_0, a point which it approaches arbitrarily closely arbitrarily often. More precisely, P_0 is a limit point of $P(t)$ if for any $\epsilon>0$ and any time t_1 there exists a time $t_2(\epsilon,t_1)>t_1$ such that $|P(t_2)-P_0|<\epsilon$. Here absolute-value signs denote distance in phase space. Because Γ is continuously deformed as t varies, every point on the trajectory through P_0 is also a limit point of $P(t)$, and the set of limit points of $P(t)$ forms a trajectory, or a set of trajectories, called the *limiting trajectories* of $P(t)$. A limiting trajectory is obviously contained within R in its entirety.

If a trajectory is contained among its own limiting trajectories, it will be called *central*; otherwise it will be called *noncentral*. A central trajectory passes arbitrarily closely arbitrarily often to any point through which it has previously passed, and, in this sense at least, separate sufficiently long segments of a central trajectory are statistically similar. A noncentral trajectory remains a certain distance away from any point through which it has previously passed. It must approach its entire set of limit points asymptotically, although it need not approach any particular limiting trajectory asymptotically. Its instantaneous distance from its closest limit point is therefore a transient quantity, which becomes arbitrarily small as $t\rightarrow\infty$.

A trajectory $P(t)$ will be called *stable at a point* $P(t_1)$ if any other trajectory passing sufficiently close to $P(t_1)$ at time t_1 remains close to $P(t)$ as $t\rightarrow\infty$; i.e., $P(t)$ is stable at $P(t_1)$ if for any $\epsilon>0$ there exists a $\delta(\epsilon,t_1)>0$ such that if $|P_1(t_1)-P(t_1)|<\delta$ and $t_2>t_1$, $|P_1(t_2)-P(t_2)|<\epsilon$. Otherwise $P(t)$ will be called *unstable* at $P(t_1)$. Because Γ is continuously deformed as t varies, a trajectory which is stable at one point is stable at every point, and will be called a *stable* trajectory. A trajectory unstable at one point is unstable at every point, and will be called an *unstable* trajectory. In the special case that $P(t)$ is confined to one point, this definition of stability coincides with the familiar concept of stability of steady flow.

A stable trajectory $P(t)$ will be called uniformly stable if the distance within which a neighboring trajectory must approach a point $P(t_1)$, in order to be certain of remaining close to $P(t)$ as $t\rightarrow\infty$, itself possesses a positive lower bound as $t_1\rightarrow\infty$; i.e., $P(t)$ is uniformly stable if for any $\epsilon>0$ there exists a $\delta(\epsilon)>0$ and a time $t_0(\epsilon)$ such that if $t_1>t_0$ and $|P_1(t_1)-P(t_1)|<\delta$ and $t_2>t_1$, $|P_1(t_2)-P(t_2)|<\epsilon$. A limiting trajectory $P_0(t)$ of a uniformly stable trajectory $P(t)$ must be uniformly stable itself, since all trajectories passing sufficiently close to $P_0(t)$ must pass arbitrarily close to some point of $P(t)$ and so must remain close to $P(t)$, and hence to $P_0(t)$, as $t\rightarrow\infty$.

Since each point lies on a unique trajectory, any trajectory passing through a point through which it has previously passed must continue to repeat its past behavior, and so must be *periodic*. A trajectory $P(t)$ will be called *quasi-periodic* if for some arbitrarily large time interval τ, $P(t+\tau)$ ultimately remains arbitrarily close to $P(t)$, i.e., $P(t)$ is quasi-periodic if for any $\epsilon>0$ and for any time interval τ_0, there exists a $\tau(\epsilon,\tau_0)>\tau_0$ and a time $t_1(\epsilon,\tau_0)$ such that if $t_2>t_1$, $|P(t_2+\tau)-P(t_2)|$

$<\epsilon$. Periodic trajectories are special cases of quasi-periodic trajectories.

A trajectory which is not quasi-periodic will be called *nonperiodic*. If $P(t)$ is nonperiodic, $P(t_1+\tau)$ may be arbitrarily close to $P(t_1)$ for some time t_1 and some arbitrarily large time interval τ, but, if this is so, $P(t+\tau)$ cannot remain arbitrarily close to $P(t)$ as $t\rightarrow\infty$. Nonperiodic trajectories are of course representations of deterministic nonperiodic flow, and form the principal subject of this paper.

Periodic trajectories are obviously central. Quasi-periodic central trajectories include multiple periodic trajectories with incommensurable periods, while quasi-periodic noncentral trajectories include those which approach periodic trajectories asymptotically. Nonperiodic trajectories may be central or noncentral.

We can now establish the theorem that a trajectory with a stable limiting trajectory is quasi-periodic. For if $P_0(t)$ is a limiting trajectory of $P(t)$, two distinct points $P(t_1)$ and $P(t_1+\tau)$, with τ arbitrarily large, may be found arbitrary close to any point $P_0(t_0)$. Since $P_0(t)$ is stable, $P(t)$ and $P(t+\tau)$ must remain arbitrarily close to $P_0(t+t_0-t_1)$, and hence to each other, as $t\rightarrow\infty$, and $P(t)$ is quasi-periodic.

It follows immediately that a stable central trajectory is quasi-periodic, or, equivalently, that a nonperiodic central trajectory is unstable.

The result has far-reaching consequences when the system being considered is an observable nonperiodic system whose future state we may desire to predict. It implies that two states differing by imperceptible amounts may eventually evolve into two considerably different states. If, then, there is any error whatever in observing the present state—and in any real system such errors seem inevitable—an acceptable prediction of an instantaneous state in the distant future may well be impossible.

As for noncentral trajectories, it follows that a uniformly stable noncentral trajectory is quasi-periodic, or, equivalently, a nonperiodic noncentral trajectory is not uniformly stable. The possibility of a nonperiodic noncentral trajectory which is stable but not uniformly stable still exists. To the writer, at least, such trajectories, although possible on paper, do not seem characteristic of real hydrodynamical phenomena. Any claim that atmospheric flow, for example, is represented by a trajectory of this sort would lead to the improbable conclusion that we ought to master long-range forecasting as soon as possible, because, the longer we wait, the more difficult our task will become.

In summary, we have shown that, subject to the conditions of uniqueness, continuity, and boundedness prescribed at the beginning of this section, a central trajectory, which in a certain sense is free of transient properties, is unstable if it is nonperiodic. A noncentral trajectory, which is characterized by transient properties, is not uniformly stable if it is nonperiodic, and, if it is stable at all, its very stability is one of its transient properties, which tends to die out as time progresses. In view of the impossibility of measuring initial conditions precisely, and thereby distinguishing between a central trajectory and a nearby noncentral trajectory, all nonperiodic trajectories are effectively unstable from the point of view of practical prediction.

4. Numerical integration of nonconservative systems

The theorems of the last section can be of importance only if nonperiodic solutions of equations of the type considered actually exist. Since statistically stationary nonperiodic functions of time are not easily described analytically, particular nonperiodic solutions can probably be found most readily by numerical procedures. In this section we shall examine a numerical-integration procedure which is especially applicable to systems of equations of the form (4). In a later section we shall use this procedure to determine a nonperiodic solution of a simple set of equations.

To solve (1) numerically we may choose an initial time t_0 and a time increment Δt, and let

$$X_{i,n}=X_i(t_0+n\Delta t). \tag{8}$$

We then introduce the auxiliary approximations

$$X_{i(n+1)}=X_{i,n}+F_i(P_n)\Delta t, \tag{9}$$

$$X_{i((n+2))}=X_{i(n+1)}+F_i(P_{(n+1)})\Delta t, \tag{10}$$

where P_n and $P_{(n+1)}$ are the points whose coordinates are

$$(X_{1,n},\cdots,X_{M,n}) \quad\text{and}\quad (X_{1(n+1)},\cdots,X_{M(n+1)}).$$

The simplest numerical procedure for obtaining approximate solutions of (1) is the forward-difference procedure,

$$X_{i,n+1}=X_{i(n+1)}. \tag{11}$$

In many instances better approximations to the solutions of (1) may be obtained by a centered-difference procedure

$$X_{i,n+1}=X_{i,n-1}+2F_i(P_n)\Delta t. \tag{12}$$

This procedure is unsuitable, however, when the deterministic nature of (1) is a matter of concern, since the values of $X_{1,n},\cdots,X_{M,n}$ do not uniquely determine the values of $X_{1,n+1},\cdots,X_{M,n+1}$.

A procedure which largely overcomes the disadvantages of both the forward-difference and centered-difference procedures is the double-approximation procedure, defined by the relation

$$X_{i,n+1}=X_{i,n}+\tfrac{1}{2}[F_i(P_n)+F_i(P_{(n+1)})]\Delta t. \tag{13}$$

Here the coefficient of Δt is an approximation to the time derivative of X_i at time $t_0+(n+\frac{1}{2})\Delta t$. From (9) and (10), it follows that (13) may be rewritten

$$X_{i,n+1}=\tfrac{1}{2}(X_{i,n}+X_{i((n+2))}). \tag{14}$$

A convenient scheme for automatic computation is the successive evaluation of $X_{i(n+1)}$, $X_{i((n+2))}$, and $X_{i,n+1}$ according to (9), (10) and (14). We have used this procedure in all the computations described in this study.

In phase space a numerical solution of (1) must be represented by a jumping particle rather than a continuously moving particle. Moreover, if a digital computer is instructed to represent each number in its memory by a preassigned fixed number of bits, only certain discrete points in phase space will ever be occupied. If the numerical solution is bounded, repetitions must eventually occur, so that, strictly speaking, every numerical solution is periodic. In practice this consideration may be disregarded, if the number of different possible states is far greater than the number of iterations ever likely to be performed. The necessity for repetition could be avoided altogether by the somewhat uneconomical procedure of letting the precision of computation increase as n increases.

Consider now numerical solutions of equations (4), obtained by the forward-difference procedure (11). For such solutions,

$$Q_{n+1}=Q_n+(dQ/dt)_n\Delta t+\tfrac{1}{2}\sum_i F_i^2(P_n)\Delta t^2. \qquad (15)$$

Let S' be any surface of constant Q whose interior R' contains the ellipsoid E where dQ/dt vanishes, and let S be any surface of constant Q whose interior R contains S'.

Since $\sum F_i^2$ and dQ/dt both possess upper bounds in R', we may choose Δt so small that P_{n+1} lies in R if P_n lies in R'. Likewise, since $\sum F_i^2$ possesses an upper bound and dQ/dt possesses a *negative* upper bound in $R-R'$, we may choose Δt so small that $Q_{n+1}<Q_n$ if P_n lies in $R-R'$. Hence Δt may be chosen so small that any jumping particle which has entered R remains trapped within R, and the numerical solution does not blow up. A blow-up may still occur, however, if initially the particle is exterior to R.

Consider now the double-approximation procedure (14). The previous arguments imply not only that $P_{(n+1)}$ lies within R if P_n lies within R, but also that $P_{((n+2))}$ lies within R if $P_{(n+1)}$ lies within R. Since the region R is convex, it follows that P_{n+1}, as given by (14), lies within R if P_n lies within R. Hence if Δt is chosen so small that the forward-difference procedure does not blow up, the double-approximation procedure also does not blow up.

We note in passing that if we apply the forward-difference procedure to a conservative system where $dQ/dt=0$ everywhere,

$$Q_{n+1}=Q_n+\tfrac{1}{2}\sum_i F_i^2(P_n)\Delta t^2. \qquad (16)$$

In this case, for any fixed choice of Δt the numerical solution ultimately goes to infinity, unless it is asymptotically approaching a steady state. A similar result holds when the double-approximation procedure (14) is applied to a conservative system.

5. The convection equations of Saltzman

In this section we shall introduce a system of three ordinary differential equations whose solutions afford the simplest example of deterministic nonperiodic flow of which the writer is aware. The system is a simplification of one derived by Saltzman (1962) to study finite-amplitude convection. Although our present interest is in the nonperiodic nature of its solutions, rather than in its contributions to the convection problem, we shall describe its physical background briefly.

Rayleigh (1916) studied the flow occurring in a layer of fluid of uniform depth H, when the temperature difference between the upper and lower surfaces is maintained at a constant value ΔT. Such a system possesses a steady-state solution in which there is no motion, and the temperature varies linearly with depth, If this solution is unstable, convection should develop.

In the case where all motions are parallel to the $x-z$-plane, and no variations in the direction of the y-axis occur, the governing equations may be written (see Saltzman, 1962)

$$\frac{\partial}{\partial t}\nabla^2\psi=-\frac{\partial(\psi,\nabla^2\psi)}{\partial(x,z)}+\nu\nabla^4\psi+g\alpha\frac{\partial\theta}{\partial x}, \qquad (17)$$

$$\frac{\partial}{\partial t}\theta=-\frac{\partial(\psi,\theta)}{\partial(x,z)}+\frac{\Delta T}{H}\frac{\partial\psi}{\partial x}+\kappa\nabla^2\theta. \qquad (18)$$

Here ψ is a stream function for the two-dimensional motion, θ is the departure of temperature from that occurring in the state of no convection, and the constants g, α, ν, and κ denote, respectively, the acceleration of gravity, the coefficient of thermal expansion, the kinematic viscosity, and the thermal conductivity. The problem is most tractable when both the upper and lower boundaries are taken to be free, in which case ψ and $\nabla^2\psi$ vanish at both boundaries.

Rayleigh found that fields of motion of the form

$$\psi=\psi_0\sin(\pi aH^{-1}x)\sin(\pi H^{-1}z), \qquad (19)$$

$$\theta=\theta_0\cos(\pi aH^{-1}x)\sin(\pi H^{-1}z), \qquad (20)$$

would develop if the quantity

$$R_a=g\alpha H^3\Delta T\nu^{-1}\kappa^{-1}, \qquad (21)$$

now called the *Rayleigh number*, exceeded a critical value

$$R_c=\pi^4a^{-2}(1+a^2)^3. \qquad (22)$$

The minimum value of R_c, namely $27\pi^4/4$, occurs when $a^2=\frac{1}{2}$.

Saltzman (1962) derived a set of ordinary differential equations by expanding ψ and θ in double Fourier series in x and z, with functions of t alone for coefficients, and

substituting these series into (17) and (18). He arranged the right-hand sides of the resulting equations in double-Fourier-series form, by replacing products of trigonometric functions of x (or z) by sums of trigonometric functions, and then equated coefficients of similar functions of x and z. He then reduced the resulting infinite system to a finite system by omitting reference to all but a specified finite set of functions of t, in the manner proposed by the writer (1960).

He then obtained time-dependent solutions by numerical integration. In certain cases all except three of the dependent variables eventually tended to zero, and these three variables underwent irregular, apparently nonperiodic fluctuations.

These same solutions would have been obtained if the series had at the start been truncated to include a total of three terms. Accordingly, in this study we shall let

$$a(1+a^2)^{-1}\kappa^{-1}\psi = X\sqrt{2}\sin(\pi a H^{-1}x)\sin(\pi H^{-1}z), \tag{23}$$

$$\pi R_c^{-1}R_a\Delta T^{-1}\theta = Y\sqrt{2}\cos(\pi a H^{-1}x)\sin(\pi H^{-1}z) - Z\sin(2\pi H^{-1}z), \tag{24}$$

where X, Y, and Z are functions of time alone. When expressions (23) and (24) are substituted into (17) and (18), and trigonometric terms other than those occurring in (23) and (24) are omitted, we obtain the equations

$$X^{\cdot} = -\sigma X + \sigma Y, \tag{25}$$

$$Y^{\cdot} = -XZ + rX - Y, \tag{26}$$

$$Z^{\cdot} = XY - bZ. \tag{27}$$

Here a dot denotes a derivative with respect to the dimensionless time $\tau = \pi^2 H^{-2}(1+a^2)\kappa t$, while $\sigma = \kappa^{-1}\nu$ is the *Prandtl number*, $r = R_c^{-1}R_a$, and $b = 4(1+a^2)^{-1}$. Except for multiplicative constants, our variables X, Y, and Z are the same as Saltzman's variables A, D, and G. Equations (25), (26), and (27) are the convection equations whose solutions we shall study.

In these equations X is proportional to the intensity of the convective motion, while Y is proportional to the temperature difference between the ascending and descending currents, similar signs of X and Y denoting that warm fluid is rising and cold fluid is descending. The variable Z is proportional to the distortion of the vertical temperature profile from linearity, a positive value indicating that the strongest gradients occur near the boundaries.

Equations (25)–(27) may give realistic results when the Rayleigh number is slightly supercritical, but their solutions cannot be expected to resemble those of (17) and (18) when strong convection occurs, in view of the extreme truncation.

6. Applications of linear theory

Although equations (25)–(27), as they stand, do not have the form of (4), a number of linear transformations will convert them to this form. One of the simplest of these is the transformation

$$X' = X, \quad Y' = Y, \quad Z' = Z - r - \sigma. \tag{28}$$

Solutions of (25)–(27) therefore remain bounded within a region R as $\tau \to \infty$, and the general results of Sections 2, 3 and 4 apply to these equations.

The stability of a solution $X(\tau)$, $Y(\tau)$, $Z(\tau)$ may be formally investigated by considering the behavior of small superposed perturbations $x_0(\tau)$, $y_0(\tau)$, $z_0(\tau)$. Such perturbations are temporarily governed by the linearized equations

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}^{\cdot} = \begin{bmatrix} -\sigma & \sigma & 0 \\ (r-Z) & -1 & -X \\ Y & X & -b \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}. \tag{29}$$

Since the coefficients in (29) vary with time, unless the basic state X, Y, Z is a steady-state solution of (25)–(27), a general solution of (29) is not feasible. However, the variation of the volume V_0 of a small region in phase space, as each point in the region is displaced in accordance with (25)–(27), is determined by the diagonal sum of the matrix of coefficients; specifically

$$V_0^{\cdot} = -(\sigma+b+1)V_0. \tag{30}$$

This is perhaps most readily seen by visualizing the motion in phase space as the flow of a fluid, whose divergence is

$$\frac{\partial X^{\cdot}}{\partial X} + \frac{\partial Y^{\cdot}}{\partial Y} + \frac{\partial Z^{\cdot}}{\partial Z} = -(\sigma+b+1). \tag{31}$$

Hence each small volume shrinks to zero as $\tau \to \infty$, at a rate independent of X, Y, and Z. This does not imply that each small volume shrinks to a point; it may simply become flattened into a surface. It follows that the volume of the region initially enclosed by the surface S shrinks to zero at this same rate, so that all trajectories ultimately become confined to a specific subspace having zero volume. This subspace contains all those trajectories which lie entirely within R, and so contains all central trajectories.

Equations (25)–(27) possess the steady-state solution $X=Y=Z=0$, representing the state of no convection. With this basic solution, the characteristic equation of the matrix in (29) is

$$[\lambda+b][\lambda^2+(\sigma+1)\lambda+\sigma(1-r)]=0. \tag{32}$$

This equation has three real roots when $r>0$; all are negative when $r<1$, but one is positive when $r>1$. The criterion for the onset of convection is therefore $r=1$, or $R_a=R_c$, in agreement with Rayleigh's result.

When $r>1$, equations (25)–(27) possess two additional steady-state solutions $X=Y=\pm\sqrt{b(r-1)}$, $Z=r-1$.

For either of these solutions, the characteristic equation of the matrix in (29) is

$$\lambda^3+(\sigma+b+1)\lambda^2+(r+\sigma)b\lambda+2\sigma b(r-1)=0. \quad (33)$$

This equation possesses one real negative root and two complex conjugate roots when $r>1$; the complex conjugate roots are pure imaginary if the product of the coefficients of λ^2 and λ equals the constant term, or

$$r=\sigma(\sigma+b+3)(\sigma-b-1)^{-1}. \quad (34)$$

This is the critical value of r for the instability of steady convection. Thus if $\sigma<b+1$, no positive value of r satisfies (34), and steady convection is always stable, but if $\sigma>b+1$, steady convection is unstable for sufficiently high Rayleigh numbers. This result of course applies only to idealized convection governed by (25)–(27), and not to the solutions of the partial differential equations (17) and (18).

The presence of complex roots of (34) shows that if unstable steady convection is disturbed, the motion will oscillate in intensity. What happens when the disturbances become large is not revealed by linear theory. To investigate finite-amplitude convection, and to study the subspace to which trajectories are ultimately confined, we turn to numerical integration.

TABLE 1. Numerical solution of the convection equations. Values of X, Y, Z are given at every fifth iteration N, for the first 160 iterations.

N	X	Y	Z
0000	0000	0010	0000
0005	0004	0012	0000
0010	0009	0020	0000
0015	0016	0036	0002
0020	0030	0066	0007
0025	0054	0115	0024
0030	0093	0192	0074
0035	0150	0268	0201
0040	0195	0234	0397
0045	0174	0055	0483
0050	0097	−0067	0415
0055	0025	−0093	0340
0060	−0020	−0089	0298
0065	−0046	−0084	0275
0070	−0061	−0083	0262
0075	−0070	−0086	0256
0080	−0077	−0091	0255
0085	−0084	−0095	0258
0090	−0089	−0098	0266
0095	−0093	−0098	0275
0100	−0094	−0093	0283
0105	−0092	−0086	0297
0110	−0088	−0079	0286
0115	−0083	−0073	0281
0120	−0078	−0070	0273
0125	−0075	−0071	0264
0130	−0074	−0075	0257
0135	−0076	−0080	0252
0140	−0079	−0087	0251
0145	−0083	−0093	0254
0150	−0088	−0098	0262
0155	−0092	−0099	0271
0160	−0094	−0096	0281

7. Numerical integration of the convection equations

To obtain numerical solutions of the convection equations, we must choose numerical values for the constants. Following Saltzman (1962), we shall let $\sigma=10$ and $a^2=\frac{1}{2}$, so that $b=8/3$. The critical Rayleigh number for instability of steady convection then occurs when $r=470/19=24.74$.

We shall choose the slightly supercritical value $r=28$. The states of steady convection are then represented by the points $(6\sqrt{2}, 6\sqrt{2}, 27)$ and $(-6\sqrt{2}, -6\sqrt{2}, 27)$ in phase space, while the state of no convection corresponds to the origin (0,0,0).

We have used the double-approximation procedure for numerical integration, defined by (9), (10), and (14). The value $\Delta\tau=0.01$ has been chosen for the dimensionless time increment. The computations have been performed on a Royal McBee LGP-30 electronic com-

TABLE 2. Numerical solution of the convection equations Values of X, Y, Z are given at every iteration N for which Z possesses a relative maximum, for the first 6000 iterations.

N	X	Y	Z	N	X	Y	Z
0045	0174	0055	0483	3029	0117	0075	0352
0107	−0091	−0083	0287	3098	0123	0076	0365
0168	−0092	−0084	0288	3171	0134	0082	0383
0230	−0092	−0084	0289	3268	0155	0069	0435
0292	−0092	−0083	0290	3333	−0114	−0079	0342
0354	−0093	−0083	0292	3400	−0117	−0077	0350
0416	−0093	−0083	0293	3468	−0125	−0083	0361
0478	−0094	−0082	0295	3541	−0129	−0073	0378
0540	−0094	−0082	0296	3625	−0146	−0074	0413
0602	−0095	−0082	0298	3695	0127	0079	0370
0664	−0096	−0083	0300	3772	0136	0072	0394
0726	−0097	−0083	0302	3853	−0144	−0077	0407
0789	−0097	−0081	0304	3926	0129	0072	0380
0851	−0099	−0083	0307	4014	0148	0068	0421
0914	−0100	−0081	0309	4082	−0120	−0074	0359
0977	−0100	−0080	0312	4153	−0129	−0078	0375
1040	−0102	−0080	0315	4233	−0144	−0082	0404
1103	−0104	−0081	0319	4307	0135	0081	0385
1167	−0105	−0079	0323	4417	−0162	−0069	0450
1231	−0107	−0079	0328	4480	0106	0081	0324
1295	−0111	−0082	0333	4544	0109	0082	0329
1361	−0111	−0077	0339	4609	0110	0080	0334
1427	−0116	−0079	0347	4675	0112	0076	0341
1495	−0120	−0077	0357	4741	0118	0081	0349
1566	−0125	−0072	0371	4810	0120	0074	0360
1643	−0139	−0077	0396	4881	0130	0081	0376
1722	0140	0075	0401	4963	0141	0068	0406
1798	−0135	−0072	0391	5035	−0133	−0081	0381
1882	0146	0074	0413	5124	−0151	−0076	0422
1952	−0127	−0078	0370	5192	0119	0075	0358
2029	−0135	−0070	0393	5262	0129	0083	0372
2110	0146	0083	0408	5340	0140	0079	0397
2183	−0128	−0070	0379	5419	−0137	−0067	0399
2268	−0144	−0066	0415	5495	0140	0081	0394
2337	0126	0079	0368	5576	−0141	−0072	0405
2412	0137	0081	0389	5649	0135	0082	0384
2501	−0153	−0080	0423	5752	0160	0074	0443
2569	0119	0076	0357	5816	−0110	−0081	0332
2639	0129	0082	0371	5881	−0113	−0082	0339
2717	0136	0070	0395	5948	−0114	−0075	0346
2796	−0143	−0079	0402				
2871	0134	0076	0388				
2962	−0152	−0072	0426				

puting machine. Approximately one second per iteration, aside from output time, is required.

For initial conditions we have chosen a slight departure from the state of no convection, namely (0,1,0). Table 1 has been prepared by the computer. It gives the values of N (the number of iterations), X, Y, and Z at every fifth iteration for the first 160 iterations. In the printed output (but not in the computations) the values of X, Y, and Z are multiplied by ten, and then only those figures to the left of the decimal point are printed. Thus the states of steady convection would appear as 0084, 0084, 0270 and −0084, −0084, 0270, while the state of no convection would appear as 0000, 0000, 0000.

The initial instability of the state of rest is evident. All three variables grow rapidly, as the sinking cold fluid is replaced by even colder fluid from above, and the rising warm fluid by warmer fluid from below, so that by step 35 the strength of the convection far exceeds that of steady convection. Then Y diminishes as the warm fluid is carried over the top of the convective cells, so that by step 50, when X and Y have opposite signs, warm fluid is descending and cold fluid is ascending. The motion thereupon ceases and reverses its direction, as indicated by the negative values of X following step 60. By step 85 the system has reached a state not far from that of steady convection. Between steps 85 and 150 it executes a complete oscillation in its intensity, the slight amplification being almost indetectable.

The subsequent behavior of the system is illustrated in Fig. 1, which shows the behavior of Y for the first 3000 iterations. After reaching its early peak near step 35 and then approaching equilibrium near step 85, it undergoes systematic amplified oscillations until near step 1650. At this point a critical state is reached, and thereafter Y changes sign at seemingly irregular intervals, reaching sometimes one, sometimes two, and sometimes three or more extremes of one sign before changing sign again.

Fig. 2 shows the projections on the X-Y- and Y-Z-planes in phase space of the portion of the trajectory corresponding to iterations 1400–1900. The states of steady convection are denoted by C and C'. The first portion of the trajectory spirals outward from the vicinity of C', as the oscillations about the state of steady convection, which have been occurring since step 85, continue to grow. Eventually, near step 1650, it crosses the X-Z-plane, and is then deflected toward the neighborhood of C. It temporarily spirals about C, but crosses the X-Z-plane after one circuit, and returns to the neighborhood of C', where it soon joins the spiral over which it has previously traveled. Thereafter it crosses from one spiral to the other at irregular intervals.

Fig. 3, in which the coordinates are Y and Z, is based upon the printed values of X, Y, and Z at every fifth iteration for the first 6000 iterations. These values determine X as a smooth single-valued function of Y and Z over much of the range of Y and Z; they determine X

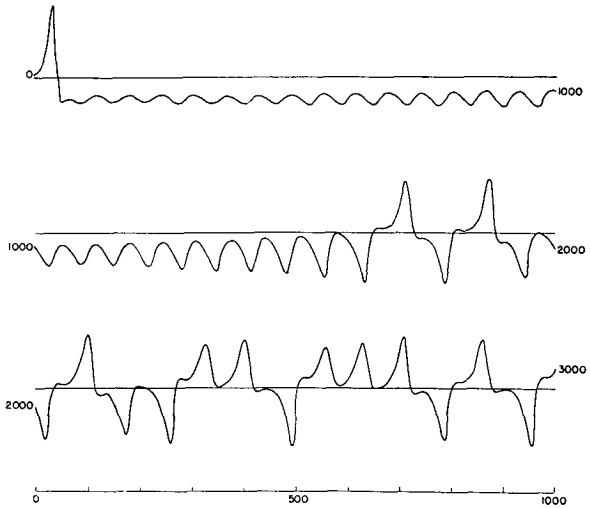


FIG. 1. Numerical solution of the convection equations. Graph of Y as a function of time for the first 1000 iterations (upper curve), second 1000 iterations (middle curve), and third 1000 iterations (lower curve).

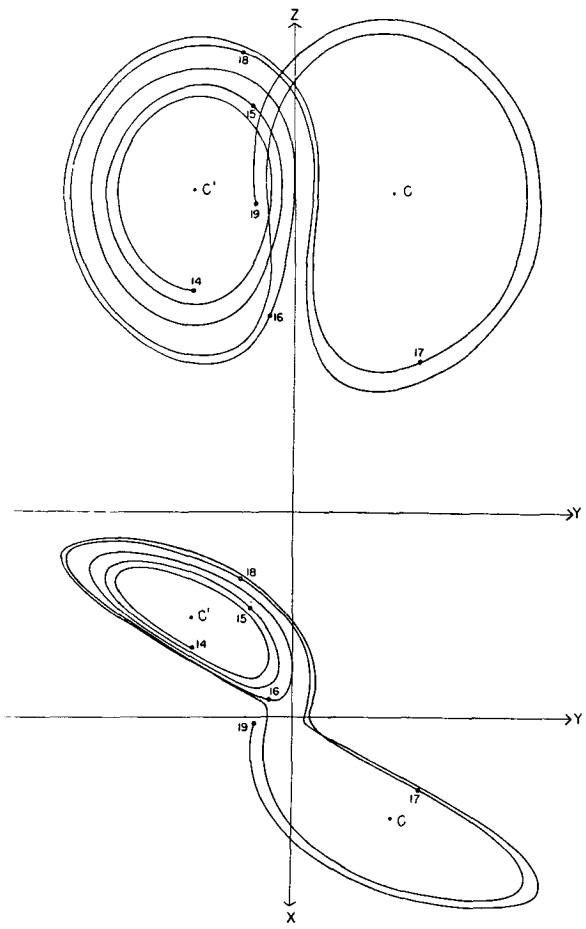


FIG. 2. Numerical solution of the convection equations. Projections on the X-Y-plane and the Y-Z-plane in phase space of the segment of the trajectory extending from iteration 1400 to iteration 1900. Numerals "14," "15," etc., denote positions at iterations 1400, 1500, etc. States of steady convection are denoted by C and C'.

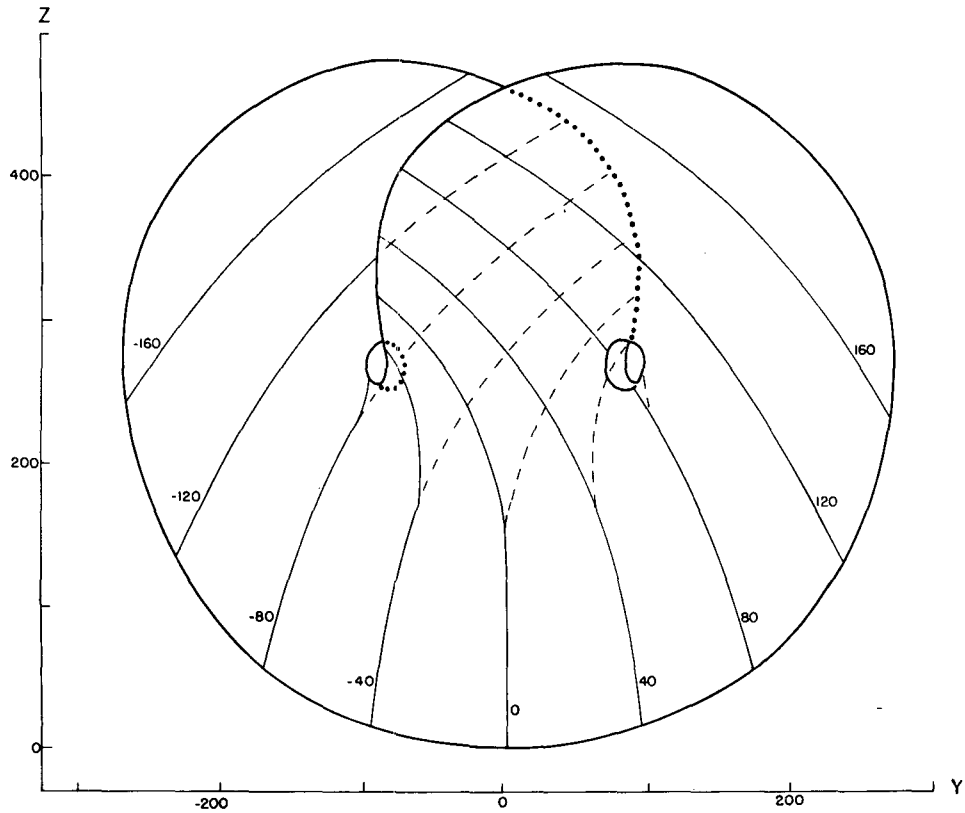


FIG. 3. Isopleths of X as a function of Y and Z (thin solid curves), and isopleths of the lower of two values of X, where two values occur (dashed curves), for approximate surfaces formed by all points on limiting trajectories. Heavy solid curve, and extensions as dotted curves, indicate natural boundaries of surfaces.

as one of two smooth single-valued functions over the remainder of the range. In Fig. 3 the thin solid lines are isopleths of X, and where two values of X exist, the dashed lines are isopleths of the lower value. Thus, within the limits of accuracy of the printed values, the trajectory is confined to a pair of surfaces which appear to merge in the lower portion of Fig. 3. The spiral about C lies in the upper surface, while the spiral about C' lies in the lower surface. Thus it is possible for the trajectory to pass back and forth from one spiral to the other without intersecting itself.

Additional numerical solutions indicate that other trajectories, originating at points well removed from these surfaces, soon meet these surfaces. The surfaces therefore appear to be composed of all points lying on limiting trajectories.

Because the origin represents a steady state, no trajectory can pass through it. However, two trajectories emanate from it, i.e., approach it asymptotically as $\tau \rightarrow -\infty$. The heavy solid curve in Fig. 3, and its extensions as dotted curves, are formed by these two trajectories. Trajectories passing close to the origin will tend to follow the heavy curve, but will not cross it, so that the heavy curve forms a natural boundary to the region which a trajectory can ultimately occupy. The holes near C and C' also represent regions which cannot be occupied after they have once been abandoned.

Returning to Fig. 2, we find that the trajectory apparently leaves one spiral only after exceeding some critical distance from the center. Moreover, the extent to which this distance is exceeded appears to determine the point at which the next spiral is entered; this in turn seems to determine the number of circuits to be executed before changing spirals again.

It therefore seems that some single feature of a given circuit should predict the same feature of the following circuit. A suitable feature of this sort is the maximum value of Z, which occurs when a circuit is nearly completed. Table 2 has again been prepared by the computer, and shows the values of X, Y, and Z at only those iterations N for which Z has a relative maximum. The succession of circuits about C and C' is indicated by the succession of positive and negative values of X and Y. Evidently X and Y change signs following a maximum which exceeds some critical value printed as about 385.

Fig. 4 has been prepared from Table 2. The abscissa is M_n, the value of the nth maximum of Z, while the ordinate is M_{n+1}, the value of the following maximum. Each point represents a pair of successive values of Z taken from Table 2. Within the limits of the round-off

in tabulating Z, there is a precise two-to-one relation between M_n and M_{n+1}. The initial maximum $M_1=483$ is shown as if it had followed a maximum $M_0=385$, since maxima near 385 are followed by close approaches to the origin, and then by exceptionally large maxima.

It follows that an investigator, unaware of the nature of the governing equations, could formulate an empirical prediction scheme from the "data" pictured in Figs. 2 and 4. From the value of the most recent maximum of Z, values at future maxima may be obtained by repeated applications of Fig. 4. Values of X, Y, and Z between maxima of Z may be found from Fig. 2, by interpolating between neighboring curves. Of course, the accuracy of predictions made by this method is limited by the exactness of Figs. 2 and 4, and, as we shall see, by the accuracy with which the initial values of X, Y, and Z are observed.

Some of the implications of Fig. 4 are revealed by considering an idealized two-to-one correspondence between successive members of sequences M_0, M_1, $\cdots$, consisting of numbers between zero and one. These sequences satisfy the relations

$$\begin{aligned} M_{n+1}&=2M_n && \text{if} \quad M_n<\tfrac{1}{2} \\ M_{n+1} &\ \text{is undefined} && \text{if} \quad M_n=\tfrac{1}{2} \\ M_{n+1}&=2-2M_n && \text{if} \quad M_n>\tfrac{1}{2}. \end{aligned} \tag{35}$$

The correspondence defined by (35) is shown in Fig. 5, which is an idealization of Fig. 4. It follows from repeated applications of (35) that in any particular sequence,

$$M_n=m_n\pm 2^n M_0, \tag{36}$$

where m_n is an even integer.

Consider first a sequence where $M_0=u/2^p$, where u is odd. In this case $M_{p-1}=\frac{1}{2}$, and the sequence terminates. These sequences form a denumerable set, and correspond to the trajectories which score direct hits upon the state of no convection.

Next consider a sequence where $M_0=u/2^p v$, where u and v are relatively prime odd numbers. Then if $k>0$, $M_{p+1+k}=u_k/v$, where u_k and v are relatively prime and u_k is even. Since for any v the number of proper fractions u_k/v is finite, repetitions must occur, and the sequence is periodic. These sequences also form a denumerable set, and correspond to periodic trajectories.

The periodic sequences having a given number of distinct values, or phases, are readily tabulated. In particular there are a single one-phase, a single two-phase, and two three-phase sequences, namely,

$$\begin{aligned} &2/3, \cdots, \\ &2/5, 4/5, \cdots, \\ &2/7, 4/7, 6/7, \cdots, \\ &2/9, 4/9, 8/9, \cdots. \end{aligned}$$

The two three-phase sequences differ qualitatively in that the former possesses two numbers, and the latter only one number, exceeding $\frac{1}{2}$. Thus the trajectory corresponding to the former makes two circuits about C, followed by one about C' (or vice versa). The trajectory corresponding to the latter makes three circuits about C, followed by three about C', so that actually only Z varies in three phases, while X and Y vary in six.

Now consider a sequence where M_0 is not a rational fraction. In this case (36) shows that M_{n+k} cannot equal

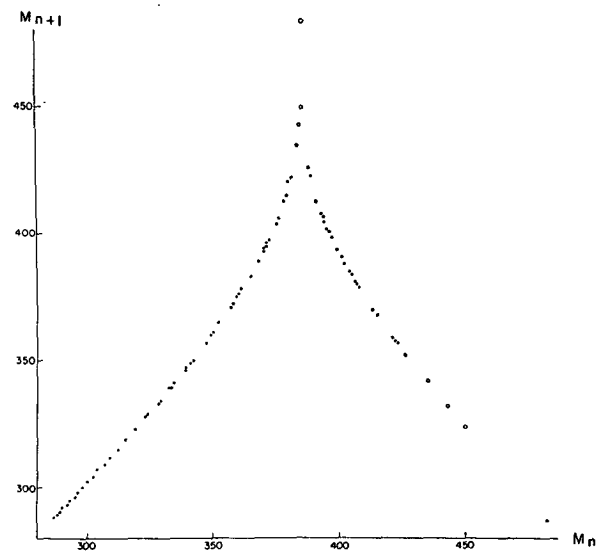


FIG. 4. Corresponding values of relative maximum of Z (abscissa) and subsequent relative maximum of Z (ordinate) occurring during the first 6000 iterations.

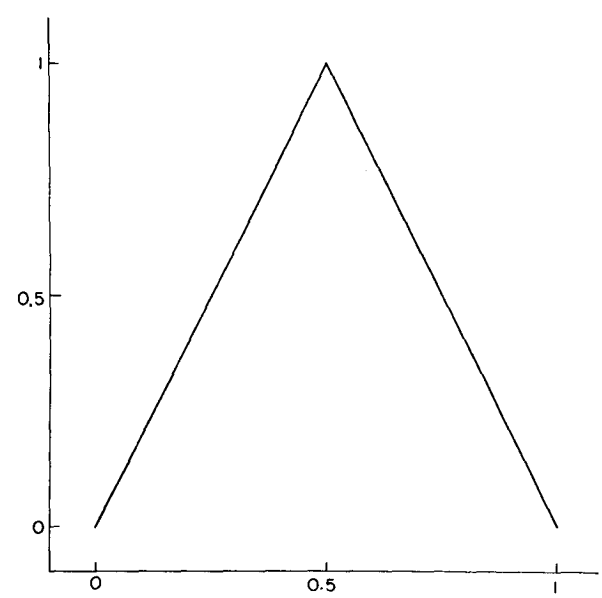


FIG. 5. The function $M_{n+1}=2M_n$ if $M_n<\frac{1}{2}$, $M_{n+1}=2-2M_n$ if $M_n>\frac{1}{2}$, serving as an idealization of the locus of points in Fig. 4.

M_n if $k>0$, so that no repetitions occur. These sequences, which form a nondenumerable set, may conceivably approach periodic sequences asymptotically and be quasi-periodic, or they may be nonperiodic.

Finally, consider two sequences M_0, M_1, $\cdots$ and M_0', M_1', $\cdots$, where $M_0'=M_0+\epsilon$. Then for a given k, if ϵ is sufficiently small, $M_k'=M_k\pm 2^k\epsilon$. All sequences are therefore unstable with respect to small modifications. In particular, all periodic sequences are unstable, and no other sequences can approach them asymptotically. All sequences except a set of measure zero are therefore nonperiodic, and correspond to nonperiodic trajectories.

Returning to Fig. 4, we see that periodic sequences analogous to those tabulated above can be found. They are given approximately by

398, $\cdots$,
377, 410, $\cdots$,
369, 391, 414, $\cdots$,
362, 380, 419, $\cdots$.

The trajectories possessing these or other periodic sequences of maxima are presumably periodic or quasi-periodic themselves.

The above sequences are temporarily approached in the numerical solution by sequences beginning at iterations 5340, 4881, 3625, and 3926. Since the numerical solution eventually departs from each of these sequences, each is presumably unstable.

More generally, if $M_n'=M_n+\epsilon$, and if ϵ is sufficiently small, $M_{n+k}'=M_{n+k}+\Lambda\epsilon$, where Λ is the product of the slopes of the curve in Fig. 4 at the points whose abscissas are M_n, $\cdots$, M_{n+k-1}. Since the curve apparently has a slope whose magnitude exceeds unity everywhere, all sequences of maxima, and hence all trajectories, are unstable. In particular, the periodic trajectories, whose sequences of maxima form a denumerable set, are unstable, and only exceptional trajectories, having the same sequences of maxima, can approach them asymptotically. The remaining trajectories, whose sequences of maxima form a nondenumerable set, therefore represent deterministic nonperiodic flow.

These conclusions have been based upon a finite segment of a numerically determined solution. They cannot be regarded as mathematically proven, even though the evidence for them is strong. One apparent contradiction requires further examination.

It is difficult to reconcile the merging of two surfaces, one containing each spiral, with the inability of two trajectories to merge. It is not difficult, however, to explain the *apparent* merging of the surfaces. At two times τ_0 and τ_1, the volumes occupied by a specified set of particles satisfy the relation

$$V_0(\tau_1)=e^{-(\sigma+b+1)(\tau_1-\tau_0)}V_0(\tau_0), \tag{37}$$

according to (30). A typical circuit about C or C' requires about 70 iterations, so that, for such a circuit, $\tau_2=\tau_1+0.7$, and, since $\sigma+b+1=41/3$,

$$V_0(\tau_1)=0.00007V_0(\tau_0). \tag{38}$$

Two particles separated from each other in a suitable direction can therefore come together very rapidly, and appear to merge.

It would seem, then, that the two surfaces merely appear to merge, and remain distinct surfaces. Following these surfaces along a path parallel to a trajectory, and circling C or C', we see that each surface is really a pair of surfaces, so that, where they appear to merge, there are really four surfaces. Continuing this process for another circuit, we see that there are really eight surfaces, etc., and we finally conclude that there is an infinite complex of surfaces, each extremely close to one or the other of two merging surfaces.

The infinite set of values at which a line parallel to the X-axis intersects these surfaces may be likened to the set of all numbers between zero and one whose decimal expansions (or some other expansions besides binary) contain only zeros and ones. This set is plainly nondenumerable, in view of its correspondence to the set of all numbers between zero and one, expressed in binary. Nevertheless it forms a set of measure zero. The sequence of ones and zeros corresponding to a particular surface contains a history of the trajectories lying in that surface, a one or zero immediately to the right of the decimal point indicating that the last circuit was about C or C', respectively, a one or zero in second place giving the same information about the next to the last circuit, etc. Repeating decimal expansions represent periodic or quasi-periodic trajectories, and, since they define rational fractions, they form a denumerable set.

If one first visualizes this infinite complex of surfaces, it should not be difficult to picture nonperiodic deterministic trajectories embedded in these surfaces.

8. Conclusion

Certain mechanically or thermally forced nonconservative hydrodynamical systems may exhibit either periodic or irregular behavior when there is no obviously related periodicity or irregularity in the forcing process. Both periodic and nonperiodic flow are observed in some experimental models when the forcing process is held constant, within the limits of experimental control. Some finite systems of ordinary differential equations designed to represent these hydrodynamical systems possess periodic analytic solutions when the forcing is strictly constant. Other such systems have yielded nonperiodic numerical solutions.

A finite system of ordinary differential equations representing forced dissipative flow often has the property that all of its solutions are ultimately confined within the same bounds. We have studied in detail the properties of solutions of systems of this sort. Our principal results concern the instability of nonperiodic solutions. A nonperiodic solution with no transient com-

ponent must be unstable, in the sense that solutions temporarily approximating it do not continue to do so. A nonperiodic solution with a transient component is sometimes stable, but in this case its stability is one of its transient properties, which tends to die out.

To verify the existence of deterministic nonperiodic flow, we have obtained numerical solutions of a system of three ordinary differential equations designed to represent a convective process. These equations possess three steady-state solutions and a denumerably infinite set of periodic solutions. All solutions, and in particular the periodic solutions, are found to be unstable. The remaining solutions therefore cannot in general approach the periodic solutions asymptotically, and so are nonperiodic.

When our results concerning the instability of nonperiodic flow are applied to the atmosphere, which is ostensibly nonperiodic, they indicate that prediction of the sufficiently distant future is impossible by any method, unless the present conditions are known exactly. In view of the inevitable inaccuracy and incompleteness of weather observations, precise very-long-range forecasting would seem to be non-existent.

There remains the question as to whether our results really apply to the atmosphere. One does not usually regard the atmosphere as either deterministic or finite, and the lack of periodicity is not a mathematical certainty, since the atmosphere has not been observed forever.

The foundation of our principal result is the eventual necessity for any bounded system of finite dimensionality to come arbitrarily close to acquiring a state which it has previously assumed. If the system is stable, its future development will then remain arbitrarily close to its past history, and it will be quasi-periodic.

In the case of the atmosphere, the crucial point is then whether analogues must have occurred since the state of the atmosphere was first observed. By analogues, we mean specifically two or more states of the atmosphere, together with its environment, which resemble each other so closely that the differences may be ascribed to errors in observation. Thus, to be analogues, two states must be closely alike in regions where observations are accurate and plentiful, while they need not be at all alike in regions where there are no observations at all, whether these be regions of the atmosphere or the environment. If, however, some unobserved features are implicit in a succession of observed states, two successions of states must be nearly alike in order to be analogues.

If it is true that two analogues have occurred since atmospheric observation first began, it follows, since the atmosphere has not been observed to be periodic, that the successions of states following these analogues must eventually have differed, and no forecasting scheme could have given correct results both times. If, instead, analogues have not occurred during this period, some accurate very-long-range prediction scheme, using observations at present available, may exist. But, if it does exist, the atmosphere will acquire a quasi-periodic behavior, never to be lost, once an analogue occurs. This quasi-periodic behavior need not be established, though, even if very-long-range forecasting is feasible, if the variety of possible atmospheric states is so immense that analogues need never occur. It should be noted that these conclusions do not depend upon whether or not the atmosphere is deterministic.

There remains the very important question as to how long is "very-long-range." Our results do not give the answer for the atmosphere; conceivably it could be a few days or a few centuries. In an idealized system, whether it be the simple convective model described here, or a complicated system designed to resemble the atmosphere as closely as possible, the answer may be obtained by comparing pairs of numerical solutions having nearly identical initial conditions. In the case of the real atmosphere, if all other methods fail, we can wait for an analogue.

Acknowledgments. The writer is indebted to Dr. Barry Saltzman for bringing to his attention the existence of nonperiodic solutions of the convection equations. Special thanks are due to Miss Ellen Fetter for handling the many numerical computations and preparing the graphical presentations of the numerical material.

REFERENCES

Birkhoff, G. O., 1927: *Dynamical systems.* New York, Amer. Math. Soc., Colloq. Publ., 295 pp.

Ford, L. R., 1933: *Differential equations.* New York, McGraw-Hill, 264 pp.

Fultz, D., R. R. Long, G. V. Owens, W. Bohan, R. Kaylor and J. Weil, 1959: Studies of thermal convection in a rotating cylinder with some implications for large-scale atmospheric motions. *Meteor. Monog,* 4(21), Amer. Meteor. Soc., 104 pp.

Gibbs, J. W., 1902: *Elementary principles in statistical mechanics.* New York, Scribner, 207 pp.

Hide, R., 1958: An experimental study of thermal convection in a rotating liquid. *Phil. Trans. Roy. Soc. London,* (A), **250**, 441–478.

Lorenz, E. N., 1960: Maximum simplification of the dynamic equations. *Tellus,* **12**, 243–254.

——, 1962a: Simplified dynamic equations applied to the rotating-basin experiments. *J. atmos. Sci.,* **19**, 39–51.

——, 1962b: The statistical prediction of solutions of dynamic equations. *Proc. Internat. Symposium Numerical Weather Prediction,* Tokyo, 629–635.

Nemytskii, V. V., and V. V. Stepanov, 1960: *Qualitative theory of differential equations.* Princeton, Princeton Univ. Press, 523 pp.

Poincaré, H., 1881: Mémoire sur les courbes définies par une équation différentielle. *J. de Math.,* **7**, 375–442.

Rayleigh, Lord, 1916: On convective currents in a horizontal layer of fluid when the higher temperature is on the under side. *Phil. Mag.,* **32**, 529–546.

Saltzman, B., 1962: Finite amplitude free convection as an initial value problem—I. *J. atmos. Sci.,* **19**, 329–341.

Non-Cooperative Games
Author(s): John Nash
Source: *The Annals of Mathematics*, Second Series, Vol. 54, No. 2, (Sep., 1951), pp. 286-295
Published by: Annals of Mathematics
Stable URL: http://www.jstor.org/stable/1969529
Accessed: 01/05/2008 08:56

Your use of the JSTOR archive indicates your acceptance of JSTOR's Terms and Conditions of Use, available at http://www.jstor.org/page/info/about/policies/terms.jsp. JSTOR's Terms and Conditions of Use provides, in part, that unless you have obtained prior permission, you may not download an entire issue of a journal or multiple copies of articles, and you may use content in the JSTOR archive only for your personal, non-commercial use.

Please contact the publisher regarding any further use of this work. Publisher contact information may be obtained at http://www.jstor.org/action/showPublisher?publisherCode=annals.

Each copy of any part of a JSTOR transmission must contain the same copyright notice that appears on the screen or printed page of such transmission.

JSTOR is a not-for-profit organization founded in 1995 to build trusted digital archives for scholarship. We enable the scholarly community to preserve their work and the materials they rely upon, and to build a common research platform that promotes the discovery and use of these resources. For more information about JSTOR, please contact support@jstor.org.

http://www.jstor.org

ANNALS OF MATHEMATICS
Vol. 54, No. 2, September, 1951

NON-COOPERATIVE GAMES

JOHN NASH

(Received October 11, 1950)

Introduction

Von Neumann and Morgenstern have developed a very fruitful theory of two-person zero-sum games in their book *Theory of Games and Economic Behavior*. This book also contains a theory of n-person games of a type which we would call cooperative. This theory is based on an analysis of the interrelationships of the various coalitions which can be formed by the players of the game.

Our theory, in contradistinction, is based on the *absence* of coalitions in that it is assumed that each participant acts independently, without collaboration or communication with any of the others.

The notion of an *equilibrium point* is the basic ingredient in our theory. This notion yields a generalization of the concept of the solution of a two-person zero-sum game. It turns out that the set of equilibrium points of a two-person zero-sum game is simply the set of all pairs of opposing "good strategies."

In the immediately following sections we shall define equilibrium points and prove that a finite non-cooperative game always has at least one equilibrium point. We shall also introduce the notions of solvability and strong solvability of a non-cooperative game and prove a theorem on the geometrical structure of the set of equilibrium points of a solvable game.

As an example of the application of our theory we include a solution of a simplified three person poker game.

Formal Definitions and Terminology

In this section we define the basic concepts of this paper and set up standard terminology and notation. Important definitions will be preceded by a subtitle indicating the concept defined. The non-cooperative idea will be implicit, rather than explicit, below.

Finite Game:

For us an *n-person game* will be a set of n *players*, or *positions*, each with an associated finite set of *pure strategies*; and corresponding to each player, i, a *payoff function*, p_i, which maps the set of all n-tuples of pure strategies into the real numbers. When we use the term *n-tuple* we shall always mean a set of n items, with each item associated with a different player.

Mixed Strategy, s_i :

A *mixed strategy* of player i will be a collection of non-negative numbers which have unit sum and are in one to one correspondence with his pure strategies.

We write $s_i = \sum_\alpha c_{i\alpha}\pi_{i\alpha}$ with $c_{i\alpha} \geqq 0$ and $\sum_\alpha c_{i\alpha} = 1$ to represent such a mixed strategy, where the $\pi_{i\alpha}$'s are the pure strategies of player i. We regard the s_i's as points in a simplex whose vertices are the $\pi_{i\alpha}$'s. This simplex may be re-

garded as a convex subset of a real vector space, giving us a natural process of linear combination for the mixed strategies.

We shall use the suffixes i, j, k for players and α, β, γ to indicate various pure strategies of a player. The symbols s_i, t_i, and r_i, etc. will indicate mixed strategies; $\pi_{i\alpha}$ will indicate the ith player's α^{th} pure strategy, etc.

Payoff function, p_i:

The payoff function, p_i, used in the definition of a finite game above, has a unique extension to the n-tuples of mixed strategies which is linear in the mixed strategy of each player [n-linear]. This extension we shall also denote by p_i, writing $p_i(s_1, s_2, \cdots, s_n)$.

We shall write $\mathfrak{s}$ or $\mathfrak{t}$ to denote an n-tuple of mixed strategies and if $\mathfrak{s} = (s_1, s_2, \cdots, s_n)$ then $p_i(\mathfrak{s})$ shall mean $p_i(s_1, s_2, \cdots, s_n)$. Such an n-tuple, $\mathfrak{s}$, will also be regarded as a point in a vector space, the product space of the vector spaces containing the mixed strategies. And the set of all such n-tuples forms, of course, a convex polytope, the product of the simplices representing the mixed strategies.

For convenience we introduce the substitution notation $(\mathfrak{s}; t_i)$ to stand for $(s_1, s_2, \cdots, s_{i-1}, t_i, s_{i+1}, \cdots, s_n)$ where $\mathfrak{s} = (s_1, s_2, \cdots, s_n)$. The effect of successive substitutions $((\mathfrak{s}; t_i); r_j)$ we indicate by $(\mathfrak{s}; t_i; r_j)$, etc.

Equilibrium Point:

An n-tuple $\mathfrak{s}$ is an *equilibrium point* if and only if for every i

$$p_i(\mathfrak{s}) = \max_{\text{all } r_i\text{'s}} [p_i(\mathfrak{s}; r_i)]. \tag{1}$$

Thus an equilibrium point is an n-tuple $\mathfrak{s}$ such that each player's mixed strategy maximizes his payoff if the strategies of the others are held fixed. Thus each player's strategy is optimal against those of the others. We shall occasionally abbreviate equilibrium point by eq. pt.

We say that a mixed strategy s_i *uses* a pure strategy $\pi_{i\alpha}$ if $s_i = \sum_\beta c_{i\beta}\pi_{i\beta}$ and $c_{i\alpha} > 0$. If $\mathfrak{s} = (s_1, s_2, \cdots, s_n)$ and s_i uses $\pi_{i\alpha}$ we also say that $\mathfrak{s}$ uses $\pi_{i\alpha}$.

From the linearity of $p_i(s_1, \cdots, s_n)$ in s_i,

$$\max_{\text{all } r_i\text{'s}} [p_i(\mathfrak{s}; r_i)] = \max_\alpha [p_i(\mathfrak{s}; \pi_{i\alpha})]. \tag{2}$$

We define $p_{i\alpha}(\mathfrak{s}) = p_i(\mathfrak{s}; \pi_{i\alpha})$. Then we obtain the following trivial necessary and sufficient condition for $\mathfrak{s}$ to be an equilibrium point:

$$p_i(\mathfrak{s}) = \max_\alpha p_{i\alpha}(\mathfrak{s}). \tag{3}$$

If $\mathfrak{s} = (s_1, s_2, \cdots, s_n)$ and $s_i = \sum_\alpha c_{i\alpha}\pi_{i\alpha}$ then $p_i(\mathfrak{s}) = \sum_\alpha c_{i\alpha}p_{i\alpha}(\mathfrak{s})$, consequently for (3) to hold we must have $c_{i\alpha} = 0$ whenever $p_{i\alpha}(\mathfrak{s}) < \max_\beta p_{i\beta}(\mathfrak{s})$, which is to say that $\mathfrak{s}$ does not use $\pi_{i\alpha}$ unless it is an optimal pure strategy for player i. So we write

$$\text{if } \pi_{i\alpha} \text{ is used in } \mathfrak{s} \text{ then } p_{i\alpha}(\mathfrak{s}) = \max_\beta p_{i\beta}(\mathfrak{s}) \tag{4}$$

as another necessary and sufficient condition for an equilibrium point.

Since a criterion (3) for an eq. pt. can be expressed by the equating of n pairs of continuous functions on the space of n-tuples $\mathfrak{s}$ the eq. pts. obviously form a closed subset of this space. Actually, this subset is formed from a number of pieces of algebraic varieties, cut out by other algebraic varieties.

Existence of Equilibrium Points

A proof of this existence theorem based on Kakutani's generalized fixed point theorem was published in Proc. Nat. Acad. Sci. U. S. A., 36, pp. 48–49. The proof given here is a considerable improvement over that earlier version and is based directly on the Brouwer theorem. We proceed by constructing a continuous transformation T of the space of n-tuples such that the fixed points of T are the equilibrium points of the game.

THEOREM 1. *Every finite game has an equilibrium point.*

PROOF. Let $\mathfrak{s}$ be an n-tuple of mixed strategies, $p_i(\mathfrak{s})$ the corresponding pay-off to player i, and $p_{i\alpha}(\mathfrak{s})$ the pay-off to player i if he changes to his α^{th} pure strategy $\pi_{i\alpha}$ and the others continue to use their respective mixed strategies from $\mathfrak{s}$. We now define a set of continuous functions of $\mathfrak{s}$ by

$$\varphi_{i\alpha}(\mathfrak{s}) = \max\,(0, p_{i\alpha}(\mathfrak{s}) - p_i(\mathfrak{s}))$$

and for each component s_i of $\mathfrak{s}$ we define a modification s_i' by

$$s_i' = \frac{s_i + \sum_\alpha \varphi_{i\alpha}(\mathfrak{s})\pi_{i\alpha}}{1 + \sum_\alpha \varphi_{i\alpha}(\mathfrak{s})},$$

calling $\mathfrak{s}'$ the n-tuple $(s_1', s_2', s_3' \cdots s_n')$.

We must now show that the fixed points of the mapping $T\colon \mathfrak{s} \to \mathfrak{s}'$ are the equilibrium points.

First consider any n-tuple $\mathfrak{s}$. In $\mathfrak{s}$ the ith player's mixed strategy s_i will use certain of his pure strategies. Some one of these strategies, say $\pi_{i\alpha}$, must be "least profitable" so that $p_{i\alpha}(\mathfrak{s}) \leqq p_i(\mathfrak{s})$. This will make $\varphi_{i\alpha}(\mathfrak{s}) = 0$.

Now if this n-tuple $\mathfrak{s}$ happens to be fixed under T the proportion of $\pi_{i\alpha}$ used in s_i must not be decreased by T. Hence, for all β's, $\varphi_{i\beta}(\mathfrak{s})$ must be zero to prevent the denominator of the expression defining s_i' from exceeding 1.

Thus, if $\mathfrak{s}$ is fixed under T, for any i and β $\varphi_{i\beta}(\mathfrak{s}) = 0$. This means no player can improve his pay-off by moving to a pure strategy $\pi_{i\beta}$. But this is just a criterion for an eq. pt. [see (2)].

Conversely, if $\mathfrak{s}$ is an eq. pt. it is immediate that all φ's vanish, making $\mathfrak{s}$ a fixed point under T.

Since the space of n-tuples is a cell the Brouwer fixed point theorem requires that T must have at least one fixed point $\mathfrak{s}$, which must be an equilibrium point.

Symmetries of Games

An *automorphism*, or *symmetry*, of a game will be a permutation of its pure strategies which satisfies certain conditions, given below.

If two strategies belong to a single player they must go into two strategies belonging to a single player. Thus if ϕ is the permutation of the pure strategies it induces a permutation ψ of the players.

Each n-tuple of pure strategies is therefore permuted into another n-tuple of pure strategies. We may call χ the induced permutation of these n-tuples. Let ξ denote an n-tuple of pure strategies and $p_i(\xi)$ the payoff to player i when the n-tuple ξ is employed. We require that if

$$j = i^{\psi} \qquad \text{then } p_j(\xi^{\chi}) = p_i(\xi)$$

which completes the definition of a symmetry.

The permutation ϕ has a unique linear extension to the mixed strategies. If

$$s_i = \sum_{\alpha} c_{i\alpha}\pi_{i\alpha} \quad \text{we define} \quad (s_i)^{\phi} = \sum_{\alpha} c_{i\alpha}(\pi_{i\alpha})^{\phi}.$$

The extension of ϕ to the mixed strategies clearly generates an extension of χ to the n-tuples of mixed strategies. We shall also denote this by χ.

We define a *symmetric n-tuple* $\mathfrak{s}$ of a game by $\mathfrak{s}^{\chi} = \mathfrak{s}$ for all χ's

THEOREM 2. *Any finite game has a symmetric equilibrium point.*

PROOF. First we note that $s_{i0} = \sum_{\alpha}\pi_{i\alpha}/\sum_{\alpha} 1$ has the property $(s_{i0})^{\phi} = s_{j0}$ where $j = i^{\psi}$, so that the n-tuple $\mathfrak{s}_0 = (s_{10}, s_{20}, \cdots, s_{n0})$ is fixed under any χ; hence any game has at least one symmetric n-tuple.

If $\mathfrak{s} = (s_1, \cdots, s_n)$ and $\mathfrak{t} = (t_1, \cdots, t_n)$ are symmetric then

$$\frac{\mathfrak{s} + \mathfrak{t}}{2} = \left(\frac{s_1 + t_1}{2}, \frac{s_2 + t_2}{2}, \ldots, \frac{s_n + t_n}{2}\right)$$

is also symmetric because $\mathfrak{s}^{\chi} = \mathfrak{s} \longleftrightarrow s_j = (s_i)^{\phi}$, where $j = i^{\psi}$, hence

$$\frac{s_j + t_j}{2} = \frac{(s_i)^{\phi} + (t_i)^{\phi}}{2} = \left(\frac{s_i + t_i}{2}\right)^{\phi},$$

hence

$$\left(\frac{\mathfrak{s} + \mathfrak{t}}{2}\right)^{\chi} = \frac{\mathfrak{s} + \mathfrak{t}}{2}.$$

This shows that the set of symmetric n-tuples is a convex subset of the space of n-tubles since it is obviously closed.

Now observe that the mapping $T: \mathfrak{s} \to \mathfrak{s}'$ used in the proof of the existence theorem was intrinsically defined. Therefore, if $\mathfrak{s}_2 = T\mathfrak{s}_1$ and χ is derived from an automorphism of the game we will have $\mathfrak{s}_2^{\chi} = T\mathfrak{s}_1^{\chi}$. If $\mathfrak{s}_1$ is symmetric $\mathfrak{s}_1^{\chi} = \mathfrak{s}_1$ and therefore $\mathfrak{s}_2^{\chi} = T\mathfrak{s}_1 = \mathfrak{s}_2$. Consequently this mapping maps the set of symmetric n-tuples into itself.

Since this set is a cell there must be a symmetric fixed point $\mathfrak{s}$ which must be a symmetric equilibrium point.

Solutions

We define here solutions, strong solutions, and sub-solutions. A non-cooperative game does not always have a solution, but when it does the solution is unique. Strong solutions are solutions with special properties. Sub-solutions always exist and have many of the properties of solutions, but lack uniqueness.

S_1 will denote a set of mixed strategies of player i and $\mathfrak{S}$ a set of n-tuples of mixed strategies.

Solvability:

A game is *solvable* if its set, $\mathfrak{S}$, of equilibrium points satisfies the condition

$$(5) \qquad (\mathbf{t};\ r_i) \in \mathfrak{S} \quad \text{and} \quad \mathfrak{s} \in \mathfrak{S} \rightarrow (\mathfrak{s};\ r_i) \in \mathfrak{S} \qquad \text{for all } i\text{'s.}$$

This is called the *interchangeability* condition. The *solution* of a solvable game is its set, $\mathfrak{S}$, of equilibrium points.

Strong Solvability:

A game is *strongly solvable* if it has a solution, $\mathfrak{S}$, such that for all i's

$$\mathfrak{s} \in \mathfrak{S} \quad \text{and} \quad p_i(\mathfrak{s};\ r_i) = p_i(\mathfrak{s}) \rightarrow (\mathfrak{s};\ r_i) \in \mathfrak{S}$$

and then $\mathfrak{S}$ is called a *strong solution.*

Equilibrium Strategies:

In a solvable game let S_i be the set of all mixed strategies s_i such that for some $\mathbf{t}$ the n-tuple $(\mathbf{t};\ s_i)$ is an equilibrium point. [s_i is the ith component of some equilibrium point.] We call S_i the set of *equilibrium strategies* of player i.

Sub-solutions:

If $\mathfrak{S}$ is a subset of the set of equilibrium points of a game and satisfies condition (1); and if $\mathfrak{S}$ is maximal relative to this property then we call $\mathfrak{S}$ a *sub-solution.*

For any sub-solution $\mathfrak{S}$ we define the ith *factor set,* S_i, as the set of all s_i's such that $\mathfrak{S}$ contains $(\mathbf{t};\ s_i)$ for some $\mathbf{t}$.

Note that a sub-solution, when unique, is a solution; and its factor sets are the sets of equilibrium strategies.

THEOREM 3. *A sub-solution, $\mathfrak{S}$, is the set of all n-tuples $(s_1, s_2, \cdots, s_n)$ such that each $s_i \in S_i$ where S_i is the ith factor set of $\mathfrak{S}$. Geometrically, $\mathfrak{S}$ is the product of its factor sets.*

PROOF. Consider such an n-tuple $(s_1, s_2, \cdots, s_n)$. By definition $\exists\ \mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_n$ such that for each i $(\mathbf{t}_i\ ;\ s_i) \in \mathfrak{S}$. Using the condition (5) $n-1$ times we obtain successively $(\mathbf{t}_1\ ;\ s_1) \in \mathfrak{S}$, $(\mathbf{t}_1\ ;\ s_1\ ;\ s_2) \in \mathfrak{S}$, $\cdots$, $(\mathbf{t}_1\ ;\ s_1\ ;\ s_2\ ;\ \cdots\ ;\ s_n) \in \mathfrak{S}$ and the last is simply $(s_1, s_2, \cdots, s_n) \in \mathfrak{S}$, which we needed to show.

THEOREM 4. *The factor sets $S_1, S_2, \cdots, S_n$ of a sub-solution are closed and convex as subsets of the mixed strategy spaces.*

PROOF. It suffices to show two things:
(a) if s_i and $s_i' \in S_i$ then $s_i^* = (s_i + s_i')/2 \in S_i$; (b) if s_i^* is a limit point of S_i then $s_i^* \in S_i$.

Let $\mathbf{t} \in \mathfrak{S}$. Then we have $p_j(\mathbf{t};\ s_i) \geqq p_j(\mathbf{t};\ s_i\ ;\ r_j)$ and $p_j(\mathbf{t};\ s_i') \geqq p_j(\mathbf{t};\ s_i'\ ;\ r_j)$ for any r_j, by using the criterion of (1) for an eq. pt. Adding these inequalities, using the linearity of $p_j(s_1, \cdots, s_n)$ in s_i, and dividing by 2, we get $p_j(\mathbf{t};\ s_i^*) \geqq$

$p_j(\mathbf{t}; s_i^* ; r_j)$ since $s_i^* = (s_i + s_i')/2$. From this we know that $(\mathbf{t}; s_i)$ is an eq. pt. for any $\mathbf{t} \in \mathfrak{S}$. If the set of all such eq. pts. $(\mathbf{t}; s_i^*)$ is added to $\mathfrak{S}$ the augmented set clearly satisfies condition (5), and since $\mathfrak{S}$ was to be maximal it follows that $s_i^* \in S_i$.

To attack (b) note that the n-tuple $(\mathbf{t}; s_i^*)$, where $\mathbf{t} \in \mathfrak{S}$, will be a limit point of the set of n-tuples of the form $(\mathbf{t}; s_i)$ where $s_i \in S_i$, since s_i^* is a limit point of S_i. But this set is a set of eq. pts. and hence any point in its closure is an eq. pt., since the set of all eq. pts. is closed. Therefore $(\mathbf{t}; s_i^*)$ is an eq. pt. and hence $s_i^* \in S_i$ from the same argument as for s_i^*.

Values:

Let $\mathfrak{S}$ be the set of equilibrium points of a game. We define

$$v_i^+ = \max_{\mathfrak{s} \in \mathfrak{S}} [p_i(\mathfrak{s})], \qquad v_i^- = \min_{\mathfrak{s} \in \mathfrak{S}} [p_i(\mathfrak{s})].$$

If $v_i^+ = v_i^-$ we write $v_i = v_i^+ = v_i^-$. v_i^+ is the *upper value* to player i of the game; v_i^- the *lower value*; and v_i the *value*, if it exists.

Values will obviously have to exist if there is but one equilibrium point.

One can define *associated values* for a sub-solution by restricting $\mathfrak{S}$ to the eq. pts. in the sub-solution and then using the same defining equations as above.

A two-person zero-sum game is always solvable in the sense defined above. The sets of equilibrium strategies S_1 and S_2 are simply the sets of "good" strategies. Such a game is not generally strongly solvable; strong solutions exist only when there is a "saddle point" in *pure* strategies.

Simple Examples

These are intended to illustrate the concepts defined in the paper and display special phenomena which occur in these games.

The first player has the roman letter strategies and the payoff to the left, etc.

Ex. 1	5	$a\alpha$	-3	Solution $\left(\frac{9}{16}a + \frac{7}{16}b, \frac{7}{17}\alpha + \frac{10}{17}\beta\right)$
	-4	$a\beta$	4	
	-5	$b\alpha$	5	$v_1 = \frac{-5}{17}, v_2 = +\frac{1}{2}$
	3	$b\beta$	-4	
Ex. 2	1	$a\alpha$	1	Strong Solution (b, β)
	-10	$a\beta$	10	
	10	$b\alpha$	-10	$v_1 = v_2 = -1$
	-1	$b\beta$	-1	
Ex. 3	1	$a\alpha$	1	Unsolvable; equilibrium points (a, α), (b, β),
	-10	$a\beta$	-10	and $\left(\frac{a}{2} + \frac{b}{2}, \frac{\alpha}{2} + \frac{\beta}{2}\right)$. The strategies in the last case
	-10	$b\alpha$	-10	
	1	$b\beta$	1	have maxi-min and mini-max properties.
Ex. 4	1	$a\alpha$	1	Strong Solution: all pairs of mixed strategies.
	0	$a\beta$	1	
	1	$b\alpha$	0	$v_1^+ = v_2^+ = 1, v_1^- = v_2^- = 0.$
	0	$b\beta$	0	

Ex. 5	1	$a\alpha$	2	Unsolvable; eq. pts. (a, α), (b, β) and $\left(\frac{1}{4}a + \frac{3}{4}b, \frac{3}{8}\alpha + \frac{5}{8}\beta\right)$. However, empirical tests show a tendency toward (a, α).
	-1	$a\beta$	-4	
	-4	$b\alpha$	-1	
	2	$b\beta$	1	
Ex. 6	1	$a\alpha$	1	Eq. pts.: (a, α) and (b, β), with (b, β) an example of instability.
	0	$a\beta$	0	
	0	$b\alpha$	0	
	0	$b\beta$	0	

Geometrical Form of Solutions

In the two-person zero-sum case it has been shown that the set of "good" strategies of a player is a convex polyhedral subset of his strategy space. We shall obtain the same result for a player's set of equilibrium strategies in any solvable game.

THEOREM 5. *The sets $S_1, S_2, \cdots, S_n$ of equilibrium strategies in a solvable game are polyhedral convex subsets of the respective mixed strategy spaces.*

PROOF. An n-tuple $\mathbf{s}$ will be an equilibrium point if and only if for every i

$$p_i(\mathbf{s}) = \max_{\alpha} p_{i\alpha}(\mathbf{s}) \tag{6}$$

which is condition (3). An equivalent condition is for every i and α

$$p_i(\mathbf{s}) - p_{i\alpha}(\mathbf{s}) \geqq 0. \tag{7}$$

Let us now consider the form of the set S_j of equilibrium strategies, s_j, of player j. Let $\mathbf{t}$ be any equilibrium point, then $(\mathbf{t}; s_j)$ will be an equilibrium point if and only if $s_j \in S_j$, from Theorem 2. We now apply conditions (2) to $(\mathbf{t}; s_j)$, obtaining

$$s_j \in S_j \longleftrightarrow \text{for all } i, \alpha \qquad p_i(\mathbf{t}; s_j) - p_i\alpha(\mathbf{t}; s_j) \geqq 0. \tag{8}$$

Since p_i is n-linear and $\mathbf{t}$ is constant these are a set of linear inequalities of the form $F_{i\alpha}(s_j) \geqq 0$. Each such inequality is either satisfied for all s_j or for those lying on and to one side of some hyperplane passing through the strategy simplex. Therefore, the complete set [which is finite] of conditions will all be satisfied simultaneously on some convex polyhedral subset of player j's strategy simplex. [Intersection of half-spaces.]

As a corollary we may conclude that S_j is the convex closure of a finite set of mixed strategies [vertices].

Dominance and Contradiction Methods

We say that s_i' dominates s_i if $p_i(\mathbf{t}; s_i') > p_i(\mathbf{t}; s_i)$ for every $\mathbf{t}$.

This amounts to saying that s_i' gives player i a higher payoff than s_i no matter what the strategies of the other players are. To see whether a strategy s_i' dominates s_i it suffices to consider only pure strategies for the other players because of the n-linearity of p_i.

It is obvious from the definitions that *no equilibrium point can involve a dominated strategy s_i.*

The domination of one mixed strategy by another will always entail other dominations. For suppose s_i' dominates s_i and t_i uses all of the pure strategies which have a higher coefficient in s_i than in s_i'. Then for a small enough ρ

$$t_i' = t_i + \rho(s_i' - s_i)$$

is a mixed strategy; and t_i dominates t_i' by linearity.

One can prove a few properties of the set of undominated strategies. It is simply connected and is formed by the union of some collection of faces of the strategy simplex.

The information obtained by discovering dominances for one player may be of relevance to the others, insofar as the elimination of classes of mixed strategies as possible components of an equilibrium point is concerned. For the t's whose components are all undominated are all that need be considered and thus eliminating some of the strategies of one player may make possible the elimination of a new class of strategies for another player.

Another procedure which may be used in locating equilibrium points is the contradiction-type analysis. Here one assumes that an equilibrium point exists having component strategies lying within certain regions of the strategy spaces and proceeds to deduce further conditions which must be satisfied if the hypothesis is true. This sort of reasoning may be carried through several stages to eventually obtain a contradiction indicating that there is no equilibrium point satisfying the initial hypothesis.

A Three-Man Poker Game

As an example of the application of our theory to a more or less realistic case we include the simplified poker game given below. The rules are as follows:

(a) The deck is large, with equally many *high* and *low* cards, and a hand consists of one card.

(b) Two chips are used to ante, open, or call.

(c) The players play in rotation and the game ends after all have passed or after one player has opened and the others have had a chance to call.

(d) If no one bets the antes are retrieved.

(e) Otherwise the pot is divided equally among the highest hands which have bet.

We find it more satisfactory to treat the game in terms of quantities we call "behavior parameters" than in the normal form of *Theory of Games and Economic Behavior*. In the normal form representation two mixed strategies of a player may be equivalent in the sense that each makes the individual choose each available course of action in each particular situation requiring action on his part with the same frequency. That is, they represent the same behavior pattern on the part of the individual.

Behavior parameters give the probabilities of taking each of the various possible actions in each of the various possible situations which may arise. Thus they describe behavior patterns.

In terms of behavior parameters the strategies of the players may be repre-

sented as follows, assuming that since there is no point in passing with a *high* card at one's last opportunity to bet that this will not be done. The greek letters are the probabilities of the various acts.

	First Moves	Second Moves
I	α Open on *high* β Open on *low*	κ Call III on *low* λ Call II on *low* μ Call II and III on *low*
II	γ Call I on *low* δ Open on *high* ε Open on *low*	ν Call III on *low* ξ Call III and I on *low*
III	ζ Call I and II on *low* η Open on *low* θ Call I on *low* ι Call II on *low*	Player III never gets a second move

We locate all possible equilibrium points by first showing that most of the greek parameters must vanish. By dominance mainly with a little contradiction-type analysis β is eliminated and with it go γ, ζ, and θ by dominance. Then contradictions eliminate μ, ξ, ι, λ, κ, and ν in that order. This leaves us with α, δ, ε, and η. Contradiction analysis shows that none of these can be zero or one and thus we obtain a system of simultaneous algebraic equations. The equations happen to have but one solution with the variables in the range (0, 1). We get

$$\alpha = \frac{21 - \sqrt{321}}{10}, \qquad \eta = \frac{5\alpha + 1}{4}, \qquad \delta = \frac{5 - 2\alpha}{5 + \alpha}, \qquad \varepsilon = \frac{4\alpha - 1}{\alpha + 5}.$$

These yield $\alpha = .308$, $\eta = .635$, $\delta = .826$, and $\varepsilon = .044$. Since there is only one equilibrium point the game has values; these are

$$v_1 = -.147 = -\frac{(1 + 17\alpha)}{8(5 + \alpha)}, \qquad v_2 = -.096 = -\frac{1 - 2\alpha}{4},$$

and

$$v_3 = .243. = \frac{79}{40}\left(\frac{1 - \alpha}{5 + \alpha}\right).$$

A more complete investigation of this poker game is published in Annals of Mathematics Study No. 24, *Contributions to the Theory of Games*. There the solution is studied as the ratio of ante to bet varies, and the potentialities of coalitions are investigated.

Applications

The study of n-person games for which the accepted ethics of fair play imply non-cooperative playing is, of course, an obvious direction in which to apply this

theory. And poker is the most obvious target. The analysis of a more realistic poker game than our very simple model should be quite an interesting affair.

The complexity of the mathematical work needed for a complete investigation increases rather rapidly, however, with increasing complexity of the game; so that analysis of a game much more complex than the example given here might only be feasible using approximate computational methods.

A less obvious type of application is to the study of cooperative games. By a cooperative game we mean a situation involving a set of players, pure strategies, and payoffs as usual; but with the assumption that the players can and will collaborate as they do in the von Neumann and Morgenstern theory. This means the players may communicate and form coalitions which will be enforced by an umpire. It is unnecessarily restrictive, however, to assume any transferability or even comparability of the payoffs [which should be in utility units] to different players. Any desired transferability can be put into the game itself instead of assuming it possible in the extra-game collaboration.

The writer has developed a "dynamical" approach to the study of cooperative games based upon reduction to non-cooperative form. One proceeds by constructing a model of the pre-play negotiation so that the steps of negotiation become moves in a larger non-cooperative game [which will have an infinity of pure strategies] describing the total situation.

This larger game is then treated in terms of the theory of this paper [extended to infinite games] and if values are obtained they are taken as the values of the cooperative game. Thus the problem of analyzing a cooperative game becomes the problem of obtaining a suitable, and convincing, non-cooperative model for the negotiation.

The writer has, by such a treatment, obtained values for all finite two person cooperative games, and some special n-person games.

Acknowledgements

Drs. Tucker, Gale, and Kuhn gave valuable criticism and suggestions for improving the exposition of the material in this paper. David Gale suggested the investigation of symmetric games. The solution of the Poker model was a joint project undertaken by Lloyd S. Shapley and the author. Finally, the author was sustained financially by the Atomic Energy Commission in the period 1949–50 during which this work was done.

Bibliography

(1) von Neumann, Morgenstern, Theory of Games and Economic Behavior, Princeton University Press, 1944.

(2) J. F. Nash, Jr., *Equilibrium Points in N-Person Games*, Proc. Nat. Acad. Sci. U. S. A. 36 (1950) 48–49.

(3) J. F. Nash, L. S. Shapley, A Simple Three-Person Poker Game, Annals of Mathematics Study No. 24, Princeton University Press, 1950.

(4) John Nash, *Two Person Cooperative Games*, to appear in Econometrica.

(5) H. W. Kuhn, *Extensive Games*, Proc. Nat. Acad. Sci. U. S. A., 36 (1950) 570–576.

Artificial Intelligence and Language Processing

David Waltz
Editor

A Theory of the Learnable

L. G. VALIANT

ABSTRACT: Humans appear to be able to learn new concepts without needing to be programmed explicitly in any conventional sense. In this paper we regard learning as the phenomenon of knowledge acquisition in the absence of explicit programming. We give a precise methodology for studying this phenomenon from a computational viewpoint. It consists of choosing an appropriate information gathering mechanism, the learning protocol, and exploring the class of concepts that can be learned using it in a reasonable (polynomial) number of steps. Although inherent algorithmic complexity appears to set serious limits to the range of concepts that can be learned, we show that there are some important nontrivial classes of propositional concepts that can be learned in a realistic sense.

1. INTRODUCTION

Computability theory became possible once precise models became available for modeling the commonplace phenomenon of mechanical calculation. The theory that evolved has been used to explain human experience and to suggest how artificial computing devices should be built. It is also worth studying for its own sake.

The commonplace phenomenon of learning surely merits similar attention. The problem is to discover good models that are interesting to study for their own sake and that promise to be relevant both to explaining human experience and to building devices that can learn. The models should also shed light on the limits of what can be learned, just as computability does on what can be computed.

In this paper we shall say that a program for performing a task has been acquired by *learning* if it has been acquired by any means other than explicit programming. Among human skills some clearly appear to have a genetically preprogrammed element, whereas some others consist of executing an explicit sequence of instructions that has been memorized. There remains a large area of skill acquisition where no such explicit programming is identifiable. It is this area that we describe here as learning. The recognition of familiar objects, such as tables, provides such examples. These skills often have the additional property that, although we have learned them, we find it difficult to articulate what algorithm we are really using. In these cases it would be especially significant if machines could be made to acquire them by learning.

This paper is concerned with precise computational models of the learning phenomenon. We shall restrict ourselves to skills that consist of recognizing whether a *concept* (or predicate) is true or not for given data. We shall say that a concept Q has been learned if a program for recognizing it has been deduced (i.e., by some method other than the acquisition from the outside of the explicit program).

The main contribution of this paper is that it shows that it is possible to design *learning machines* that have all three of the following properties:

1. The machines can provably learn whole classes of concepts. Furthermore, these classes can be characterized.
2. The classes of concepts are appropriate and nontrivial for general-purpose knowledge.
3. The computational process by which the machines deduce the desired programs requires a feasible (i.e., polynomial) number of steps.

A learning machine consists of a *learning protocol* together with a *deduction procedure*. The former specifies the manner in which information is obtained from the outside. The latter is the mechanism by which a correct recognition algorithm for the concept to be learned is deduced. At the broadest level, the suggested *methodology* for studying learning is the following: Define a plausible learning protocol, and investigate the class of con-

This research was supported in part by National Science Foundation grant MCS-83-02385. A preliminary version of this paper appeared in the proceedings of the 16th ACM Symposium on Theory of Computing, Washington, D.C., 1984, 436–445.

© 1984 ACM 0001-0782/84/1100-1134 75¢

cepts for which recognition programs can be deduced in polynomial time using the protocol.

The specific protocols considered in this paper allow for two kinds of information supply. First the learner has access to a supply of typical data that positively exemplify the concept. More precisely, it is assumed that these positive examples have a probabilistic distribution determined arbitrarily by nature. A call of a routine EXAMPLES produces one such positive example. The relative probability of producing different examples is determined by the distribution. The second source of information that may be available is a routine ORACLE. In its most basic version, when presented with data, it will tell the learner whether or not the data positively exemplify the concept.

The major remaining design choice that has to be made is that of knowledge representation. Since our declared aim is to represent general knowledge, it seems almost unavoidable that we use some kind of logic rather than, for example, formal grammars or geometrical constructs. In this paper we shall represent concepts as Boolean functions of a set of propositional variables. The recognition algorithms that we attempt to deduce will be therefore Boolean circuits or expressions.

The adequacy of the propositional calculus for representing knowledge in practical learning systems is clearly a crucial and controversial question. Few would argue that this much power is not necessary. The question is whether it is enough. There are several arguments suggesting that such a system would, at least, be a very good start. First, when one examines the most famous examples of systems that embody preprogrammed knowledge, namely, expert systems such as DENDRAL and MYCIN, essentially no logical notation beyond the propositional calculus is used. Surely it would be overambitious to try to base learning systems on representations that are more powerful than those that have been successfully managed in programmed systems. Second, the results in this paper can be negatively interpreted to suggest that the class of learnable concepts even within the propositional calculus is severely circumscribed. This suggests that the search for extensions to the propositional calculus that have substantially larger learnable classes may be a difficult one.

The positive conclusions of this paper are that there are specific classes of concepts that are learnable in polynomial time using learning protocols of the kind described. These classes can all be characterized by defining the class of programs that recognize them. In each case the programs are special kinds of Boolean expressions. The three classes are (1) conjunctive normal form expressions with a bounded number of literals in each clause, (2) monotone disjunctive normal form expressions, and (3) arbitrary expressions in which each variable occurs just once. In the first of these, no calls of the oracle are necessary. In the last, no access to typical examples is necessary but the oracles need to be more powerful than the one described above.

The deduction procedure will in each case output an expression that with high likelihood closely approximates the expression to be learned. Such an approximate expression never says yes when it should not but may say no on a small fraction of the probability space of positive examples. This fraction can be made arbitrarily small by increasing the runtime of the deduction procedure. Perhaps the main technical discovery contained in the paper is that with this probabilistic notion of learning highly convergent learning is possible for whole classes of Boolean functions. This appears to distinguish this approach from more traditional ones where learning is seen as a process of "inducing" some general rule from information that is insufficient for a reliable deduction to be made. The power of this probabilistic viewpoint is illustrated, for example, by the fact that an arbitrary set of polynomially many positive examples cannot be relied on to determine expressions consisting even of only a single monomial in any reliable way. A more detailed description of this methodological problem can be found in [9].

There is another aspect of our formulation that is worth emphasizing. In a learning system that is about to learn a new concept, there may be an enormous number of propositional variables available. These may be primitive inputs, the values of preprogrammed concepts, or the values of concepts that have been learned previously. We want the complexity of learning the new concept to be related only to the number of variables that may be set in natural examples of it, and not on the cardinality of the universe of available variables. Hence, the questions asked of ORACLE and the values given by EXAMPLES will not be truth assignments to all the variables. In the archetypal case, they will specify an assignment to a subset of the variables that is still sufficient to guarantee the truth of the function.

Whether the classes of learnable Boolean concepts can be extended significantly beyond the three classes given is an interesting question. There is circumstantial evidence from cryptography, however, that the whole class of functions computable by polynomial size circuits is not learnable. Consider a cryptographic scheme that encodes messages by evaluating the function E_k where k specifies the key. Suppose that this scheme is immune to chosen plaintext attack in the sense that, even if the values of E_k are known for polynomially many dynamically chosen inputs, it is computationally infeasible to deduce an algorithm for E_k or for an approximation to it. This is equivalent to saying, however, that E_k is not learnable. The conjectured existence of good cryptographic functions that are easy to compute therefore implies that some easy-to-compute functions are not learnable.

If the class of learnable concepts is as severely limited as suggested by our results, then it would follow that the only way of teaching more complicated concepts is to build them up from such simple ones. Thus, a good teacher would have to identify, name, and sequence these intermediate concepts in the manner of a programmer. The results of *learnability theory* would then indicate the maximum granularity of the single

concepts that can be acquired without programming.

In summary, this paper attempts to explore the limits of what is learnable as allowed by algorithmic complexity. The results are distinguishable from the diverse body of previous work on learning because they attempt to reconcile the three properties ((1)–(3)) mentioned earlier. Closest in rigor to our approach is the inductive inference literature (see Angluin and Smith [1] for a survey) that deals with inducing such things as recursive functions or formal grammars (but not Boolean functions) from examples. There is a large body of work on pattern recognition and classification, using statistical and other tools (e.g., [4]), but the question of general knowledge representation is not addressed there directly. Learning, in various less formal senses, has been studied widely as a branch of artificial intelligence. A survey and bibliography can be found in [2, 7]. In their terminology the subject of this paper is concept learning.

In previous work on learning, much emphasis is placed on the diversity of methods that humans appear to be using. These include learning by example, learning by analogy, and learning by being told. This paper emphasizes the polarity between learning by being programmed and learning in the absence of any programming element. Many of the conventionally accepted modes of learning can be regarded as being hybrids of these two extremes.

2. A LEARNING PROTOCOL FOR BOOLEAN FUNCTIONS

We consider t Boolean variables $p_1, \ldots, p_t$, each of which can take the value 1 or 0 to indicate whether the associated proposition is true or false. There is no assumption about the independence of the variables. Indeed, they may be functions of each other.

A *vector* is an assignment to each of the t variables of a value from $\{0, 1, *\}$. The symbol $*$ denotes that a variable is *undetermined*. A vector is *total* if every variable is *determined* (i.e., is assigned a value from $\{0, 1\}$). For example, the assignment $p_1 = p_3 = 1$, $p_4 = 0$, and $p_2 = *$ is a vector that is not total.

A Boolean function F is a mapping from the set of 2^t total vectors to $\{0, 1\}$. A Boolean function F becomes a *concept* F if its domain is extended to the set of all vectors as follows: For a vector v, $F(v) = 1$ if and only if $F(w) = 1$ for all total vectors w that agree with v on all variables for which v is determined. The purpose of this extension is that it permits us not to mention the variables on which F does not depend.

Given a concept F, we consider an *arbitrary* probability distribution D over the set of all vectors v such that $F(v) = 1$. In other words, for each v such that $F(v) = 1$, it is the case that $D(v) \geq 0$. Also $\sum D(v) = 1$ when summation is over this set of vectors. There are no other assumed restrictions on D, which is intended to describe the relative frequency with which the positive examples of F occur in nature.

What are reasonable learning protocols to consider? First we must avoid giving the teacher too much power, namely, the power to communicate a program instruction by instruction. For example, if a premeditated sequence of vectors with repetitions could be communicated, then this could be used to encode the description of the program even if just two such vectors were used for binary notation. Second we must avoid giving the teacher what is evidently too little power. In particular, the protocol must provide some typical examples of vectors for which F is true, for otherwise, if F is true for just one vector that is total, only an exponential search or more powerful oracles would be able to find it.

Such considerations led us to consider the following learning protocol as a reasonable one. It gives the learner access to the following two routines:

1. EXAMPLES: This has no input. It gives as output a vector v such that $F(v) = 1$. For each such v, the probability that v is output on any single call is $D(v)$.
2. ORACLE( ): On vector v, as input it outputs 1 or 0 according to whether $F(v) = 1$ or 0.

The first of these gives randomly chosen positive examples of the concept being learned. The second provides a test for whether a vector, which the deduction procedure considers to be critical, is an example of the concept or not. In a real system, the oracle may be a human expert, a database of past observations, some deduction system, or a combination of these.

Finally we observe that our particular choice of learning protocol was strongly influenced by the earlier decision to deal with concepts rather than raw Boolean functions. If we were to deal with the latter, then many other alternatives are equally natural. For example, given a vector v, instead of asking, as we do, whether all completions of it make the function true, we could ask whether there exist any completions that make it true. This suggests natural alternative semantics for EXAMPLES and ORACLE. In Section 7 we shall use such more elaborate oracles.

3. LEARNABILITY

We shall consider various classes of programs having $p_1, \ldots, p_t$ as inputs and show that in each case any program in the class can be deduced, with only small probability of error, in polynomial time using the protocol previously described. We assume that programs can take the values 1, 0, and undetermined.

More precisely we say that a class X of programs is *learnable* with respect to a given learning protocol if and only if there exists an algorithm A (the deduction procedure) invoking the protocol with the following properties:

1. The algorithm runs in time polynomial in an adjustable parameter h, in the various parameters that quantify the size of the program to be learned, and in the number of variables t.
2. For all programs $f \in X$ and all distributions D over vectors v on which f outputs 1, the algorithm will deduce with probability at least $(1 - h^{-1})$ a program $g \in X$ that never outputs one when it should not but outputs one almost always when it should. In

particular, (i) for all vectors v, $g(v) = 1$ implies $f(v) = 1$, and (ii) the sum of $D(v)$ over all v such that $f(v) = 1$, but $g(v) \neq 1$ is at most h^{-1}.

In our definition we have disallowed positive answers from g to be wrong, only because the particular classes X in this paper do not require it. This we call one-sided error learning. In other circumstances it may be efficacious to allow two-sided errors. In that more general case, we would need to put a probability distribution $\bar{D}$ on the set of vectors such that $f(v) \neq 1$. Condition (i) in (2) is then replaced by $\sum \bar{D}(v)$ over all v such that $f(v) \neq 1$, but $g(v) = 1$ is at most h^{-1}.

A second aspect of our definition is that the parameter h is used in two independent probabilistic bounds. This simplifies the statement of the results. It would be more precise, however, to work explicitly with two independent parameters h_1 and h_2.

Third we should note that programs that compute concepts should be distinguished from those that compute merely the Boolean function. This distinction has little consequence for the three classes of expressions that we consider, since in these cases programs for the concepts follow directly from the specification of the expressions. For the sake of generality, our definitions do allow the value of a program to be undefined, since a nontotal vector will not, in general, determine the value of an expression as 0 or 1.

It would be interesting to obtain negative results, other than the informal cryptographic evidence mentioned in the introduction, indicating that the class of unrestricted Boolean circuits is not learnable. A recent result [6] in this direction does establish this, under the hypothesis that integer factorization is intractable. Their result applies to the learning protocol consisting of EXAMPLES and ORACLE, when the latter is restricted to total vectors. It may also be interesting to go in the opposite direction and determine, for example, whether the hypothesis that P equals NP would imply that all interesting classes are learnable.

4. A COMBINATORIAL BOUND

The probabilistic analysis needed for our later results can be abstracted into a single lemma. The proof of the lemma is independent of the rest of the paper.

We define the function $L(h,s)$ for any real number h greater than one and any positive integer S as follows: Let $L(h,S)$ be the smallest integer such that in $L(h,S)$ independent Bernoulli trials each with probability at least h^{-1} of success, the probability of having fewer than S successes is less than h^{-1}.

The relevance of the function $L(h,S)$ to learning can be illustrated as follows. Consider an urn U containing a very large number of marbles possibly of many different types. Suppose we wish to "learn" what kinds of marbles U contains by taking a small random sample X. If all the marbles in U are of distinct types, then it is clearly impossible to deduce much information about them. Suppose, however, that we know that there are only S types of marbles and that we wish to choose X so that it contains representatives of all but 1 percent of the marbles in U. Whatever the respective frequencies of the S types may be, the definition of $L(h,S)$ implies that if $|X| = L(100,S)$ then with probability greater than 99 percent we will have succeeded. To see why this implication holds, suppose that after $L(100,S)$ successive choices marble types representing between them at least 1 percent of the urn remain unchosen. In that case we have made $L(100,S)$ Bernoulli trials each with probability at least 1 percent of success (i.e., of choosing a previously unchosen type) and have succeeded fewer than S times (since some type remains unchosen). Note that what constitutes a success here will depend on previous choices. The probability of success, however, will always be at least 1 percent, independent of previous choices, and this is sufficient for our argument.

The following simple upper bound holds for the whole range of values of S and h and shows that $L(h,S)$ is essentially linear both in h and in S.

PROPOSITION: *For all integers $S \geq 1$ and all real $h > 1$.*

$$L(h,S) \leq 2h(S + \log_e h).$$

Proof: We use the following three well-known inequalities of which the last is due to Chernoff (see [6] page 18).

1. For all $x > 0$, $(1 + x^{-1})^x < e$.
2. For all $x > 0$, $(1 - x^{-1})^x < e^{-1}$.
3. In m independent trials, each with probability at least p of success, the probability that there are at most k success, where $k < mp$, is at most

$$\left(\frac{m - mp}{m - k}\right)^{m-k}\left(\frac{mp}{k}\right)^{k}.$$

The first factor in the expression in (3) above can be rewritten as

$$\left(1 - \frac{mp - k}{m - k}\right)^{((m-k)/(mp-k))(mp-k)}.$$

Using (2) with $x = (m - k)/(mp - k)$, we can upper bound the product by

$$e^{-mp+k}(mp/k)^k.$$

Substituting $m = 2h(S + \log_e h)$, $p = h^{-1}$, and $k = S$ and using logarithms to the base e gives the bound

$$e^{-2S-2\log h+S}\cdot[2(1 + (\log h)/S)]^{(S/\log h)\log h}.$$

Rewriting this using (1) with $x = (\log_e h)/S$ gives

$$e^{-S-2\log h}\cdot 2^S\cdot e^{\log h} \leq (2/e)^S\cdot e^{-\log h} \leq h^{-1}.$$

As an illustration of an application, suppose that we have access to EXAMPLES, a source of natural positive examples of vectors for concept F. Suppose that we want to find an approximation to the subset P of variables that are determined in at least one natural example of F. Consider the following procedure: Pick the first $L(h, |P|)$ vectors provided by EXAMPLES, and let P' be the union of the sets of variables determined by these vectors. The definition then implies that with probability at least $1 - h^{-1}$ the set P' will have the following

property: If a vector is chosen with probability distribution D, then the probability that it determines a variable in $P - P'$ is less than h^{-1}. The argument used above for the urn model establishes this as follows: If P' does not have the claimed property, then the following has happened: $L(h, |P|)$ trials have been made (i.e., calls of EXAMPLES) each with probability greater than h^{-1} of success (i.e., finding a vector that determines a variable in $P - P'$, but there have been fewer than $|P|$ successes (i.e., discoveries of new additions to P'). The probability of such a happening is less than h^{-1} by the definition of L.

The above application shows that the set of variables that are determined in natural examples of F can be approximated by a procedure whose runtime is independent of t, the total number of variables. It follows that in the definition of learnability the dependence on t in part (1) can be relaxed to a dependence on the number of variables that are ever defined in positive examples.

5. BOUNDED CNF EXPRESSIONS

A *conjunctive normal form* (CNF) expression is any product $c_1c_2 \cdots c_r$ of clauses where each clause c_i is a sum $q_1 + q_2 + \cdots + q_{j_i}$ of literals. A *literal* q is either a variable p or the negation $\bar{p}$ of a variable. For example, $(p_1 + \bar{p}_2)(\bar{p}_1 + \bar{p}_2 + p_3)$ is a CNF expression. For a positive integer k, a *k-CNF expression* is a CNF expression where each clause is the sum of at most k literals.

For CNF expressions in general, we do not know of any polynomial time deduction procedure with respect to our learning protocol. The question of whether one exists is tantalizing because of its apparent simplicity.

In this section we shall prove that for any fixed integer k the class of k-CNF expressions is learnable. In fact it is learnable easily in the sense that calls of EXAMPLES suffice and no calls of ORACLE are required.

In this and subsequent sections, we shall use the relation $\Rightarrow$ on concepts as follows: $F \Rightarrow G$ means that for all vectors v whenever $F(v) = 1$ it is also the case that $G(v) = 1$. (N.B. This is equivalent to the relation $F \Rightarrow G$ when F, G are regarded as functions.) For brevity we shall often denote both expressions and even vectors by the concepts they represent. Thus the concept represented by vector w is true for exactly those vectors that agree with w on all variables on which w is determined. The concept represented by expression f is obtained by considering the Boolean function computed by f and extending its domain to become a concept. In this section we regard a clause c_i to be a special kind of expression. In Sections 7 and 8, we consider monomials m, simple products of literals, as special forms of expressions. In all cases statements of the form $f \Rightarrow g$, $v \Rightarrow c_i$, $v \Rightarrow m$, or $m \Rightarrow f$ can be understood by interpreting the two sides as concepts or functions in the obvious way.

THEOREM A: *For any positive integer k, the class of k-CNF expressions is learnable via an algorithm A that uses* $L = L(h, (2t)^{k+1})$ *calls of EXAMPLES and no calls of ORACLE, where t is the number of variables.*

Proof: The algorithm is initialized with formula g as the product of all possible clauses of up to k literals from $\{p_1, \bar{p}_1, p_2, \bar{p}_2, \ldots, p_t, \bar{p}_t\}$. Clearly the number of ways of choosing a clause of exactly k literals is at most $(2t)^k$, and hence the number of ways of choosing a clause with up to k literals is less than $2t + (2t)^2 + \cdots + (2t)^k < (2t)^{k+1}$. This bounds the initial number of clauses in g.

The algorithm then calls EXAMPLES L times to give positive examples of the concept represented by k-CNF formula f. For each vector v so obtained, it deletes all of the remaining clauses in g that do not contain a literal that is determined to be true in v (i.e., in the clauses deleted, each literal is either not determined or is negated in v). More precisely the following is repeated L times:

begin $v :=$ EXAMPLES

for each c_i in g delete c_i if $v \not\Rightarrow c_i$.

end

The claim is that with high probability the value of g after the Lth execution of this block will be the desired approximation to the formula f that is being learned.

Initially the set of vectors $\{v \mid v \Rightarrow g\}$ is clearly empty. We first claim that as the algorithm proceeds the set $\{v \mid v \Rightarrow g\}$ will always be a subset of $\{v \mid v \Rightarrow f\}$. To prove this it is clearly sufficient to prove the same statement when both sets are restricted to only total vectors. Let B be the product of all the clauses c containing up to k literals with the property that "$\forall v$ if $v \Rightarrow f$ then $v \Rightarrow c$." It is clear that in the course of the algorithm no clause in B can ever be deleted. Hence it is certainly true that $\{v \mid v \Rightarrow g\} \subset \{v \mid v \Rightarrow B\}$. To establish the claim, it remains only to prove that B computes the same Boolean function as f. (In fact B will be the maximal k-CNF formula equivalent to f.) It is easy to see that $f \Rightarrow B$, since by the definition of B, for every c in B it is the case that "for all v if $v \Rightarrow f$ then $v \Rightarrow c$." To verify the converse, that $B \Rightarrow f$, we first note that, by definition, f is a k-CNF formula. If some clause in f did not occur in B, then this clause c' would have the property that "$\exists v$ such that $v \Rightarrow f$ but $v \not\Rightarrow c'$." But this is impossible since if c' is a clause of f and if $v \not\Rightarrow c'$ then $v \not\Rightarrow f$. We conclude that every k-CNF representation of the function that f represents consists of a set of clauses that is a subset of the clauses in B. Hence $B \Rightarrow f$.

Let $X = \sum D(v)$ with summation over all (not necessarily total) vectors v such that $v \Rightarrow f$ but $v \not\Rightarrow g$. This quantity is defined for each intermediate value of g in the course of the algorithm and, as is easily verified, it is monotone decreasing with time. Now clauses will be removed from g whenever EXAMPLES outputs a vector v such that $v \not\Rightarrow g$. The probability of this happening at any moment is exactly the current value of X. Also, the process of running the algorithm to completion can have one of just two possible outcomes: (1) At some point X becomes less than h^{-1}, in which case the g found will approximate to f exactly as required by the definition of learnability. (2) The value of X never goes below h^{-1}, and hence g is not an acceptable approximation. The probability of the latter eventuality is, how-

ever, at most h^{-1} since it corresponds to the situation of performing $L(h, (2t)^{k+1})$ Bernoulli experiments (i.e., calls of EXAMPLES) each with probability greater than h^{-1} of success (i.e., finding a v such that $v \not\Rightarrow g$) and obtaining fewer than $(2t)^{k+1}$ successes (a success being manifested by the removal of at least one clause from g). □

In conclusion we observe that a CNF expression g immediately yields a program for computing the associated concept. Given a vector v, we substitute the determined truth values in g. The concept will be true for v if and only if all the clauses are made true by the substitution.

6. DNF EXPRESSIONS

A *disjunctive normal form* (DNF) expression is any sum $m_1 + m_2 + \cdots + m_r$ of monomials where each monomial m_i is a product of literals. For example, $p_1\bar{p}_2 + p_1p_3\bar{p}_4$ is a DNF expression. Such expressions appear particularly easy for humans to comprehend. Hence we expect that any practical learning system would have to allow for them.

An expression is *monotone* if no variable is negated in it. We shall show that for monotone DNF expressions there exists a simple deduction procedure that uses both EXAMPLES and ORACLE. For unrestricted DNF a similar result can be proved with respect to a different size measure. In the unrestricted case, there is the additional difficulty that we can guarantee to deduce a program only for the function and not for the concept. This difficulty does not arise in the monotone case where given an expression we can always compute the value of the associated concept for a vector v by making the substitution and asking whether the resulting expression is identically true.

A monomial m is a *prime implicant* of a function F (or of an expression representing the function) if $m \Rightarrow F$ and if $m' \not\Rightarrow F$ for any m' obtained by deleting one literal from m. A DNF expression is *prime* if it consists of the sum of prime implicants none of which is redundant in the sense of being implied by the sum of the others. There is always a unique prime DNF expression in the monotone case, but not in general. We therefore define the *degree* of a DNF expression to be the largest number of monomials that a prime DNF expression equivalent to it can have. The unique prime DNF expression in the monotone case is simply the sum of all the prime implicants.

THEOREM B: *The class of monotone DNF expressions is learnable via an algorithm B that uses $L = L(h,d)$ calls of EXAMPLES and dt calls of ORACLE, where d is the degree of the DNF expression f to be learned and t the number of variables.*

Proof: The algorithm is initialized with formula g identically equal to the constant zero. The algorithm then calls EXAMPLES L times to produce positive examples of f. Each time a vector v is produced such that $v \not\Rightarrow g$, a new monomial m is added to g. The monomial m is the product of those literals determined in v that are essential to make $v \not\Rightarrow f$. More precisely, the loop that is executed L times is the following:

```
begin v := EXAMPLES
      if v ⇏ g then
      begin for i := 1 to t do
            if p_i is determined in v then
            begin set ṽ equal to v but with p_i := *;
                  if ORACLE(ṽ) = 1 then v := ṽ
            end
            set m equal to the product of all literals q
              such that v ⇒ q;
            g := g + m
      end
end
```

The test $v \not\Rightarrow g$ amounts to asking whether none of the monomials of g are made true by the values determined to be true in v. Every time EXAMPLES produces a value v such that $v \not\Rightarrow g$, the inner loop of the algorithm will find a prime implicant m to add to g. Each such m is different from any previously added (since the contrary would have implied that $v \Rightarrow g$). It follows that such a v will be found at most d times, and hence ORACLE will be called at most dt times.

Let $X = \sum D(w)$ with summation over all (not necessarily total) vectors w such that $w \Rightarrow f$ but $w \not\Rightarrow g$. This quantity is defined for each intermediate value of g in the course of the algorithm, is initially unity, and decreases monotonically with time. Now a monomial will be added to g each time EXAMPLES outputs a vector v such that $v \not\Rightarrow g$. The probability of this occurring at any call of EXAMPLES is exactly X. The process of running the algorithm to completion can have just two possible outcomes: (1) At some time X has become less than h^{-1}, in which case the final expression g found will approximate to f as required by the definition of learnability; and (2) the value of X never goes below h^{-1}, and hence g is not an acceptable approximation. The probability of this second eventuality is, however, at most h^{-1}, since it corresponds to the situation of performing $L(h,d)$ Bernoulli experiments (i.e., calls of EXAMPLES) each with probability greater than h^{-1} of success (i.e., of finding a v such that $v \Rightarrow f$ but $v \not\Rightarrow g$) and obtaining fewer than d successes (each manifested by the addition of a new monomial). □

For unrestricted DNF expressions, several problems arise. The main source of difficulty is the fact that the problem of determining whether a nontotal vector implies the function specified by a DNF formula is *NP*-hard. This is simply because the problem of determining whether the nowhere determined vector implies the function is the tautology question of Cook [3]. An immediate consequence of this is that it is unreasonable to assume that the program being learned computes concepts rather than functions. Another consequence is that in any algorithm akin to Algorithm B the test $v \not\Rightarrow g$ may not be feasible if v is not total. Algorithm B is sufficient, however, to establish the following:

THEOREM B′: *Suppose the notion of learnability is restricted to distributions D such that $D(v) = 0$ whenever v is not total. Then the class of DNF expressions is learnable, in*

the sense that programs for the functions (but not necessarily the concepts) can be deduced in L(h,d) calls of EXAMPLES and dt calls of ORACLE where d is the degree of the DNF expression to be learned.

The reader should note that here there is the additional philosophical difficulty that availability of the expression does not in itself imply a polynomial time algorithm for ORACLE. On the other hand, the theorem does say that if an agent has some, maybe ad hoc, black box for ORACLE whose workings are unknown to him or her, it can be used to teach someone else a DNF expression that approximates the function.

Finally, it may be worth emphasizing that the monotone case is nonproblematic and the deduction procedure for it very simpleminded. It may be interesting to pursue more sophisticated deduction procedures for it if contexts can be found in which they can be proved advantageous. The question as to whether monotone DNF expressions can be learned from EXAMPLES alone is open. A positive answer would be especially significant. A partial result in this direction can be obtained by dualizing Theorem A: For any k the class of DNF expressions having a bound k on the length of each disjunct can be learned in polynomial time from negative examples alone (i.e., from a source of vectors v such that $\bar{D}(v) \neq 0$).

7. μ-EXPRESSIONS

We have already seen a class of expressions, namely the k-CNF expressions, that can be learned from positive examples alone. Here we shall consider the other extreme, a class that can be learned using oracles alone.

Deducing expressions on which there are less severe structural restrictions than DNF or CNF appears much more difficult. The aim of this section is to give a paradigmatic example of how far one can go in that direction if one is willing to pay the price of oracles that are more sophisticated than the one previously described.

A general *expression* over variable set $\{p_1, \ldots, p_t\}$ is defined recursively as follows:

1. For each i $(1 \leq i \leq t)$ "p_i" and "$\bar{p}_i$" are expressions;
2. if $f_1, \ldots, f_r$ are expressions, then "$(f_1 + f_2 + \cdots + f_r)$" is an expression (called a plus expression);
3. if $f_1, \ldots, f_r$ are expressions, then "$(f_1 \times f_2 \times \cdots \times f_r)$" is an expression (called a times expression).

A *μ-expression* is an expression in which each variable appears at most once. We can assume that a μ-expression is monotone, since we can always relabel negated variables with new names that denote their negation. In the recursive definition of any μ-expression, there are clearly at most $2t - 1$ intermediate expressions of which at most t are of type (1) and at most $t - 1$ of types (2) or (3). Without loss of generality, we shall assume that in the definition of a μ-expression rules (2) and (3) alternate. We shall regard two μ-expressions as identical if their definitions can be made formally the same by reordering sums and products and relabeling as necessary.

For learning μ-expressions, we shall employ more powerful oracles. The boundary between reasonable and unreasonable oracles does not appear sharp. We make no claims about the reasonableness of these new oracles except that they may serve as vehicles for understanding learnability.

The definitions refer to the Boolean function F (not regarded as a concept here). The oracle of Section 2 will be renamed N-ORACLE, since it is one of *necessity*: N-ORACLE$(v) = 1$ if and only if for all total vectors w such that $w \Rightarrow v$ it is the case that $F(w) = 1$. The dual of this would be a *possibility oracle*: P-ORACLE$(v) = 1$ if and only if there exists a total vector w such that $w \Rightarrow v$ and $F(w) = 1$. For brevity we shall also define the *prime implicant* oracle: $PI(v) = 1$ if and only if N-ORACLE$(v) = 1$, but N-ORACLE$(w) = 0$ for any w obtained from v by making one variable determined in v undetermined.

Finally we define two further oracles, ones of *relevant possibility RP* and of *relevant accompaniment RA*. For convenience we define the first by how it behaves on vectors represented as monomials: $RP(m) = 1$ iff from some monomial $m'\ mm'$ is a prime implicant of f. For sets V, W of variables, we define $RA(V, W) = 1$ iff every prime implicant of f that contains a variable from V also contains a variable from W.

THEOREM C: *The class of μ-expressions in learnable via a deduction procedure C that uses $0(t^3)$ calls of N-ORACLE RP and RA altogether, where t is the number of variables (and no calls of EXAMPLES). The procedure always deduces exactly the correct expression.*

Proof: Let λ be the monomial that is the product of no literals. The algorithm will first compute $RP(p_j)$ for each p_j to determine which of the variables occur in the prime implicants of the function that the expression f to be learned represents. Let $g_1, \ldots, g_r$ be distinct single variable expressions, one for each p_j for which $RP(p_j) = 1$. Note that these are exactly the variables that occur in f, since f is monotone.

With each g_i we associate two monomials m_i and $\hat{m}_i$. The former will be the single variable p_j that g_i represents. The latter is defined as any monomial $\hat{m}$ having no variable in common with m_i such that $m_i\hat{m}$ is a prime implicant of f. The algorithm will construct each such $\hat{m}_i$ as the final value of m in the following procedure: Set $m = \lambda$; while $PI(mm_i) = 0$ find a p_k not in mm_i such that $RP(p_k mm_i) = 1$ and set $m := p_k m$. Hence in at most t^2 calls of PI (i.e., t^3 calls of N-ORACLE) and t^2 calls of RP values for every $\hat{m}_i$ will be found.

Once initialized the algorithm proceeds by alternately executing a *plus-phase* and a *times-phase*. At the start of each phase, we have a set of expressions $g_1, \ldots, g_r$ where each g_i is associated with two monomials m_i and $\hat{m}_i$ (having no variables in common) where m_i is a prime implicant of g_i and $m_i\hat{m}_i$ is a prime implicant of f. We distinguish the gs as *plus* or *times expressions* according to whether the outermost construction rule was addition or multiplication. A plus-phase will first compute an equivalence relation S on the subset of $\{g_i\}$ that are times expressions. For each equivalence class G such that no $g_i \in G$ already occurs as a summand in a

sum, we construct a new expression that is the sum of the members of G and call this sum g_k where k is a previously unused index. If some members of G already occur in a sum, say g_j (N.B. they are never distributed in more than one sum), then we modify the sum expression g_j to equal the sum of every expression in G. A times-phase is exactly analogous except that it computes a different equivalence relation T, now on plus expressions, and will form new, or extend old, times expressions. In the above context, single variable expressions will be regarded as both times and plus expressions. Also, it is immaterial which of the two kinds of phase is used to start the algorithm.

The intention of the algorithm is best expressed by the claim to follow, which will be verified later. Suppose that the expressions occurring in the definition of f are $f_1, \ldots, f_q$ (where $q \leq 2t - 1$). We shall say that $g \leq f_i$ iff the set of prime implicants of g is a subset of the set of prime implicants of $\tilde{f}_i$ where (1) if f_i is a plus expression then $\tilde{f}_i = f_i$ and (2) if f_i is a times expression then $\tilde{f}_i$ is the product of some subset of the multiplicands of f_i.

Claim 1: After every phase of the algorithm for every g_i that has been constructed, there is exactly one expression f_i such that (1) $g_i \leq f_i$ and (2) f_i is of the same kind (i.e., plus or times) as g_i.

The procedure builds up the rooted tree of the expression as rooted subtrees starting from the leaves. One evident difficulty is that there is no a priori knowledge available about the shape of the tree. Hence in the grafting process a subtree may become attached to another at just about any node.

Whenever a plus or times expression g_i is created or enlarged, its associated m_i and $\hat{m}_i$ are updated as follows. If g_i is a sum, then we let $m_i = m_j$ and $\hat{m}_i = \hat{m}_j$ for any g_j that is a summand in g_i. If g_i is a product, then m_i will be the product of all the m_js that correspond to multiplicands in g_i. Finally $\hat{m}_i$ will be generated as the final value of m in the following procedure: Set $m := \lambda$; while $PI(mm_i) = 0$ find a p_k not in mm_i such that $RP(p_k mm_i) = 1$ and set $m := p_k m$. Since such an $\hat{m}_i$ has to be found at most t times in the overall algorithm, the total cost is at most t^2 calls of PI (i.e., t^3 calls of N-ORACLE) and t^2 calls of RP.

In order to complete the description of the algorithm, it remains only to define the equivalence relations S and T.

Definition: $g_i S g_j$ if and only if

1. $PI(m_i\hat{m}_j) = PI(m_j\hat{m}_i) = 1$ and
2. m_i, $\hat{m}_j$ contain disjoint sets of variables, as do m_j, $\hat{m}_i$.

For defining T we shall denote by V_i the set of variables that occur in the expression g_i.

Definition: $g_i T g_j$ if and only if $RA(V_i, V_j) = RA(V_j, V_i) = 1$.

First we shall verify that S and T are indeed equivalence relations under the assumption that Claim 1 holds at the beginning of each phase. Clearly S and T are defined to be both reflexive and symmetric. To verify that T is transitive, suppose that $g_i T g_j$ and $g_j T g_k$. The former implies that every prime implicant of f containing some variable from g_i also contains some variable from g_j. The latter implies that every prime implicant of f containing some variable from g_j also contains a variable from g_k. Hence $g_i T g_k$ follows. In order to verify the transitivity of S, we shall make a more general observation about any subexpression of f.

Claim 2: If m_i, m_j are prime implicants of distinct times subexpressions f_i, f_j of f and if for some m, with no variable in common with either m_i or m_j, both $m_i m$ and $m_j m$ are prime implicants of f, then f_i, f_j must occur as summands in the same plus expression of f.

Proof: Most easily verified by representing expression f as a directed graph (e.g., [8]) with edges labeled by the Boolean variables. The sets of labels along directed paths between a distinguished source node and a distinguished sink node correspond exactly to prime implicants in the case of μ expressions where all edges will have different labels.

Now to verify that S is transitive, suppose that $g_i S g_j$ and $g_j S g_k$ where, by Claim 1, $g_i \leq f_i$, $g_j \leq f_j$, and $g_k \leq f_k$ for appropriate times expressions f_i, f_j, f_k. Then it follows that $m_i\hat{m}_j$, $m_j\hat{m}_j$, and $m_k\hat{m}_j$ are all prime implicants of f where m_i, m_j, m_k have no variable in common with $\hat{m}_j$. It follows from Claim 2 that f_i, f_j, and f_k are addends in the same sum expression. Hence $g_i S g_k$ follows.

Claim 3: Suppose that after some phase of the algorithm Claim 1 holds and times expressions g_i, g_j have been formed where $g_i \leq f_i$, $g_j \leq f_j$, and f_i, f_j are times expressions. Then g_i and g_j will be placed in the same sum expression at the next plus-phase if and only if $f_i = \tilde{f}_i$, $f_j = \tilde{f}_j$, and f_i, f_j are addends in the same sum expression in f.

Proof: ($\Rightarrow$) If $g_i \leq f_i = \tilde{f}_i$, $g_j \leq f_j = \tilde{f}_j$, and f_i, f_j are in the same sum, then m_i, m_j will be prime implicants of f_i, f_j. Also $m_i\hat{m}_j$ and $m_j\hat{m}_i$ will be prime implicants of f, and the variables in m_i will be disjoint from those in $\hat{m}_j$ as will be those in m_j from those in $\hat{m}_i$. Hence $g_i S g_j$ will hold and g_i,g_j will be in the same plus expression after the next plus-phase.

($\Leftarrow$) Suppose that $g_i \leq f_i$, $g_j \leq f_j$, and $g_i S g_j$ holds. Then m_i, m_j will be prime implicants of g_i, g_j, respectively, $m_i\hat{m}_j$, $m_j\hat{m}_j$ will both be prime implicants of f, and the variables in $\hat{m}_j$ will be disjoint from those of both m_i and m_j. It follows from Claim 2 that $f_i = \tilde{f}_i$ and $f_j = \tilde{f}_j$ must occur as summands in the same plus expression of f. [N.B. Here we are using the fact that Claim 2 remains true if we allow a "times subexpression" to be the product of any subset of the multiplicands in a times expression in f.]

Claim 4: Suppose that after some phase of the algorithm, Claim 1 holds and some plus expressions $g_i \leq f_i$ and $g_j \leq f_j$ (f_i, f_j plus expressions) have been found. (1) If $g_i = f_i$, $g_j = f_j$, and f_i, f_j are multiplicands in the same times expression in f, then g_i, g_j will be placed in the same times expression after the next times phase. (2) If

g_i, g_j are placed in the same times expression at any subsequent phase, then f_i, f_j are in the same product in f.

Proof: (1) If the conditions of (1) above hold, then $g_i T g_j$ will be discovered at the next times-phase and the claim follows. (2) If f_i, f_j are not in the same product in f, then f contains some prime implicant containing variables from one of g_i, g_j and not from the other. Hence $g_i T g_j$ will never hold.

Proof of Claim 1: By induction on the number of phases on Claims 1, 3, and 4 simultaneously.

We define the *depth* of a formula f_i to be the maximum number of alternations of sum and product required in its definition.

Claim 5: If f_i is an expression of depth k in f, then after k phases a g_i identical to f_i will have been constructed by the algorithm.

Proof: By induction on Claims 3 and 4.

To conclude the proof of the theorem, it remains to analyze the runtime. This is dominated by the cost of computing m_i and $\hat{m}_i$, for which we have already accounted, plus the cost of computing the equivalence relations S and T. We first note that fewer than $2t$ expression names g_i are used in the course of the algorithm. When an expression is grafted into another at a point deep in the latter, then the semantics of all the subexpressions above will change. By Claims 3 and 4 such grafting can occur when a g_i is added to a sum expression but not when added to a times expression. Hence the values m_i and $\hat{m}_i$ do not need to be changed for any expression when such grafting occurs. It follows that for computing S only $2t$ values of m_i, $\hat{m}_i$ need to be considered. Hence $0(t^2)$ calls of PI or $0(t^3)$ calls of N-ORACLE suffice overall. For computing T grafting may cause a ripple effect. On each occasion the value of V_j may have to be updated for up to t such sets, and hence t^2 calls of RA will suffice. Hence $0(t^3)$ calls of RA in the overall algorithm will be enough. □

8. REMARKS

In this paper we have considered learning as the process of deducing a program for performing a task, from information that does not provide an explicit description of such a program. We have given precise meaning to this notion of learning and have shown that in some restricted but nontrivial contexts it is computationally feasible.

Consider a world containing robots and elephants. Suppose that one of the robots has discovered a recognition algorithm for elephants that can be meaningfully expressed in k-conjunctive normal form. Our Theorem A implies that this robot can communicate its algorithm to the rest of the robot population by simply exclaiming "elephant" whenever one appears.

An important aspect of our approach, if cast in its greatest generality, is that we require the recognition algorithms of the teacher and learner to agree on an overwhelming fraction of only the natural inputs. Their behavior on unnatural inputs is irrelevant, and hence descriptions of all possible worlds are not necessary. If followed to its conclusion, this idea has considerable philosophical implications: A learnable concept is nothing more than a short program that distinguishes some natural inputs from some others. If such a concept is passed on among a population in a distributed manner, substantial variations in meaning may arise. More importantly, what consensus there is will only be meaningful for natural inputs. The behavior of an individual's program for unnatural inputs has no relevance. Hence thought experiments and logical arguments involving unnatural hypothetical situations may be meaningless activities.

The second important aspect of the formulation is that the notion of oracles makes it possible to discuss a whole range of teacher–learner interactions beyond the mere identification of examples. This is significant in the context of artificial intelligence where humans may be willing to go to great lengths to convey their skills to machines but are frustrated by their inability to articulate the algorithms they themselves use in the practice of the skills. We expect that some explicit programming does become essential for transmitting skills that are beyond certain limits of difficulty. The identification of these limits is a major goal of the line of work proposed in this paper.

REFERENCES

1. Angluin, D., Smith, C.H. Inductive inference: Theory and methods. *Comput. Surv. 15*, 3 (Sept. 1983), 237–269.
2. Barr, A., and Feigenbaum, E.A. *The Handbook of Artificial Intelligence.* Vol. 2. William Kaufmann, Los Altos, Calif., 1982.
3. Cook, S.A. The complexity of theorem proving procedures. In *Proceedings of 3rd Annual ACM Symposium on Theory of Computing* (Shaker Heights, Ohio, May 3–5). ACM, New York, 1971, 151–158.
4. Duda, R.O., and Hart, P.F. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.
5. Erdös, P., and Spencer, J. *Probabilistic Methods in Combinatorics.* Academic Press, New York, 1974.
6. Goldreich, O., Goldwasser, S., and Micali, S. How to construct random functions. In *Proceedings of 25th IEEE Symposium on Foundations of Computer Science* (Singer Island, Fla., Oct. 24–26). IEEE, New York, 1984.
7. Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. *Machine Learning: An Artificial Intelligence Approach.* Tioga Publishing Co., Palo Alto, Calif., 1983.
8. Skyum, S., and Valiant, L.G. A complexity theory based on Boolean algebra. In *Proceedings of 22nd IEEE Symposium on Foundations of Computer Science* (Nashville, Tenn., Oct. 28–30). IEEE, New York, 1981, 244–253.
9. Valiant, L.G. Deductive learning. *Philosophical Transactions of the Royal Society of London* (1984). To be published.

CR Categories and Subject Descriptors: F.1.1 [**Computation by Abstract Devices**]: Models of Computation; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; I.2.6 [**Artificial Intelligence**]: Learning—*concept learning*

General Terms: Algorithms

Additional Key Words and Phrases: inductive inference, probabilistic models of learning, propositional expressions

Received 1/84; revised 6/84; accepted 7/84

Author's Present Address: L.G. Valiant, Center for Research in Computing Technology, Aiken Computation Laboratory, Harvard University, Cambridge, MA 02138.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

Nature **261** 459–67 (1976)

Simple mathematical models with very complicated dynamics

Robert M. May*

First-order difference equations arise in many contexts in the biological, economic and social sciences. Such equations, even though simple and deterministic, can exhibit a surprising array of dynamical behaviour, from stable points, to a bifurcating hierarchy of stable cycles, to apparently random fluctuations. There are consequently many fascinating problems, some concerned with delicate mathematical aspects of the fine structure of the trajectories, and some concerned with the practical implications and applications. This is an interpretive review of them.

THERE are many situations, in many disciplines, which can be described, at least to a crude first approximation, by a simple first-order difference equation. Studies of the dynamical properties of such models usually consist of finding constant equilibrium solutions, and then conducting a linearised analysis to determine their stability with respect to small disturbances: explicitly nonlinear dynamical features are usually not considered.

Recent studies have, however, shown that the very simplest nonlinear difference equations can possess an extraordinarily rich spectrum of dynamical behaviour, from stable points, through cascades of stable cycles, to a regime in which the behaviour (although fully deterministic) is in many respects "chaotic", or indistinguishable from the sample function of a random process.

This review article has several aims.

First, although the main features of these nonlinear phenomena have been discovered and independently rediscovered by several people, I know of no source where all the main results are collected together. I have therefore tried to give such a synoptic account. This is done in a brief and descriptive way, and includes some new material: the detailed mathematical proofs are to be found in the technical literature, to which signposts are given.

Second, I indicate some of the interesting mathematical questions which do not seem to be fully resolved. Some of these problems are of a practical kind, to do with providing a probabilistic description for trajectories which seem random, even though their underlying structure is deterministic. Other problems are of intrinsic mathematical interest, and treat such things as the pathology of the bifurcation structure, or the truly random behaviour, that can arise when the nonlinear function $F(X)$ of equation (1) is not analytical. One aim here is to stimulate research on these questions, particularly on the empirical questions which relate to processing data.

Third, consideration is given to some fields where these notions may find practical application. Such applications range from the abstractly metaphorical (where, for example, the transition from a stable point to "chaos" serves as a metaphor for the onset of turbulence in a fluid), to models for the dynamic behaviour of biological populations (where one can seek to use field or laboratory data to estimate the values of the parameters in the difference equation).

*King's College Research Centre, Cambridge CB2 1ST; on leave from Biology Department, Princeton University, Princeton 08540.

Fourth, there is a very brief review of the literature pertaining to the way this spectrum of behaviour—stable points, stable cycles, chaos—can arise in second or higher order difference equations (that is, two or more dimensions; two or more interacting species), where the onset of chaos usually requires less severe nonlinearities. Differential equations are also surveyed in this light; it seems that a three-dimensional system of first-order ordinary differential equations is required for the manifestation of chaotic behaviour.

The review ends with an evangelical plea for the introduction of these difference equations into elementary mathematics courses, so that students' intuition may be enriched by seeing the wild things that simple nonlinear equations can do.

First-order difference equations

One of the simplest systems an ecologist can study is a seasonally breeding population in which generations do not overlap[1–4]. Many natural populations, particularly among temperate zone insects (including many economically important crop and orchard pests), are of this kind. In this situation, the observational data will usually consist of information about the maximum, or the average, or the total population in each generation. The theoretician seeks to understand how the magnitude of the population in generation $t+1$, X_{t+1}, is related to the magnitude of the population in the preceding generation t, X_t: such a relationship may be expressed in the general form

$$X_{t+1} = F(X_t) \tag{1}$$

The function $F(X)$ will usually be what a biologist calls "density dependent", and a mathematician calls nonlinear; equation (1) is then a first-order, nonlinear difference equation.

Although I shall henceforth adopt the habit of referring to the variable X as "the population", there are countless situations outside population biology where the basic equation (1) applies. There are other examples in biology, as, for example in genetics[5,6] (where the equation describes the change in gene frequency in time) or in epidemiology[7] (with X the fraction of the population infected at time t). Examples in economics include models for the relationship between commodity quantity and price[8], for the theory of business cycles[9], and for the temporal sequences generated by various other economic quantities[10]. The general equation (1) also is germane to the social sciences[11], where it arises, for example, in theories of

learning (where X may be the number of bits of information that can be remembered after an interval t), or in the propagation of rumours in variously structured societies (where X is the number of people to have heard the rumour after time t). The imaginative reader will be able to invent other contexts for equation (1).

In many of these contexts, and for biological populations in particular, there is a tendency for the variable X to increase from one generation to the next when it is small, and for it to decrease when it is large. That is, the nonlinear function $F(X)$ often has the following properties: $F(0)=0$; $F(X)$ increases monotonically as X increases through the range $0<X<A$ (with $F(X)$ attaining its maximum value at $X=A$); and $F(X)$ decreases monotonically as X increases beyond $X=A$. Moreover, $F(X)$ will usually contain one or more parameters which "tune" the severity of this nonlinear behaviour; parameters which tune the steepness of the hump in the $F(X)$ curve. These parameters will typically have some biological or economic or sociological significance.

A specific example is afforded by the equation[1,4,12–23]

$$N_{t+1} = N_t(a-bN_t) \tag{2}$$

This is sometimes called the "logistic" difference equation. In the limit $b=0$, it describes a population growing purely exponentially (for $a>1$); for $b\neq 0$, the quadratic nonlinearity produces a growth curve with a hump, the steepness of which is tuned by the parameter a. By writing $X=bN/a$, the equation may be brought into canonical form[1,4,12–23]

$$X_{t+1} = aX_t(1-X_t) \tag{3}$$

In this form, which is illustrated in Fig. 1, it is arguably the simplest nonlinear difference equation. I shall use equation (3) for most of the numerical examples and illustrations in this article. Although attractive to mathematicians by virtue of its extreme simplicity, in practical applications equation (3) has the disadvantage that it requires X to remain on the interval $0<X<1$; if X ever exceeds unity, subsequent iterations diverge towards $-\infty$ (which means the population becomes extinct). Furthermore, $F(X)$ in equation (3) attains a maximum value of $a/4$ (at $X=\frac{1}{2}$); the equation therefore possesses non-trivial dynamical behaviour only if $a<4$. On the other hand, all trajectories are attracted to $X=0$ if $a<1$. Thus for non-trivial

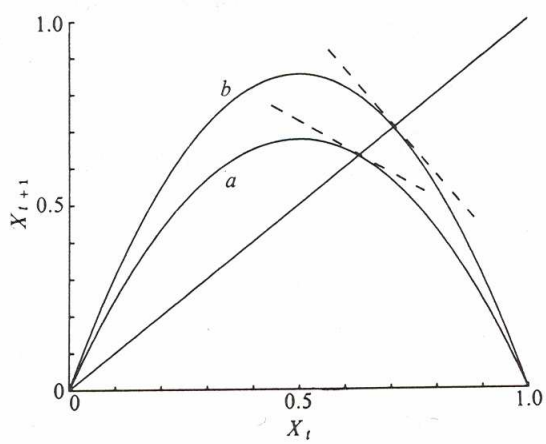


Fig. 1 A typical form for the relationship between X_{t+1} and X_t described by equation (1). The curves are for equation (3), with $a = 2.707$ (a); and $a = 3.414$ (b). The dashed lines indicate the slope at the "fixed points" where $F(X)$ intersects the 45° line: for the case a this slope is less steep than −45° and the fixed point is stable; for b the slope is steeper than −45°, and the point is unstable.

dynamical behaviour we require $1<a<4$; failing this, the population becomes extinct.

Another example, with a more secure provenance in the biological literature[1,23–27], is the equation

$$X_{t+1} = X_t \exp[r(1-X_t)] \tag{4}$$

This again describes a population with a propensity to simple exponential growth at low densities, and a tendency to decrease at high densities. The steepness of this nonlinear behaviour is tuned by the parameter r. The model is plausible for a single species population which is regulated by an epidemic disease at high density[28]. The function $F(X)$ of equation (4) is slightly more complicated than that of equation (3), but has the compensating advantage that local stability implies global stability[1] for all $X>0$.

The forms (3) and (4) by no means exhaust the list of single-humped functions $F(X)$ for equation (1) which can be culled from the ecological literature. A fairly full such catalogue is given, complete with references, by May and Oster[1]. Other similar mathematical functions are given by Metropolis *et al.*[16]. Yet other forms for $F(X)$ are discussed under the heading of "mathematical curiosities" below.

Dynamic properties of equation (1)

Possible constant, equilibrium values (or "fixed points") of X in equation (1) may be found algebraically by putting $X_{t+1}=X_t=X^*$, and solving the resulting equation

$$X^* = F(X^*) \tag{5}$$

An equivalent graphical method is to find the points where the curve $F(X)$ that maps X_t into X_{t+1} intersects the 45° line, $X_{t+1}=X_t$, which corresponds to the ideal nirvana of zero population growth; see Fig. 1. For the single-hump curves discussed above, and exemplified by equations (3) and (4), there are two such points: the trivial solution $X=0$, and a non-trivial solution X^* (which for equation (3) is $X^*=1-[1/a]$).

The next question concerns the stability of the equilibrium point X^*. This can be seen[24,25,19–21,1,4] to depend on the slope of the $F(X)$ curve at X^*. This slope, which is illustrated by the dashed lines in Fig. 1, can be designated

$$\lambda^{(1)}(X^*) = [\mathrm{d}F/\mathrm{d}X]_{X=X^*} \tag{6}$$

So long as this slope lies between 45° and −45° (that is, $\lambda^{(1)}$ between +1 and −1), making an acute angle with the 45° ZPG line, the equilibrium point X^* will be at least locally stable, attracting all trajectories in its neighbourhood. In equation (3), for example, this slope is $\lambda^{(1)}=2-a$: the equilibrium point is therefore stable, and attracts all trajectories originating in the interval $0<X<1$, if and only if $1<a<3$.

As the relevant parameters are tuned so that the curve $F(X)$ becomes more and more steeply humped, this stability-determining slope at X^* may eventually steepen beyond −45° (that is, $\lambda^{(1)}<-1$), whereupon the equilibrium point X^* is no longer stable.

What happens next? What happens, for example, for $a>3$ in equation (3)?

To answer this question, it is helpful to look at the map which relates the populations at successive intervals 2 generations apart; that is, to look at the function which relates X_{t+2} to X_t. This second iterate of equation (1) can be written

$$X_{t+2} = F[F(X_t)] \tag{7}$$

or, introducing an obvious piece of notation,

$$X_{t+2} = F^{(2)}(X_t) \tag{8}$$

The map so derived from equation (3) is illustrated in Figs 2 and 3.

Population values which recur every second generation (that

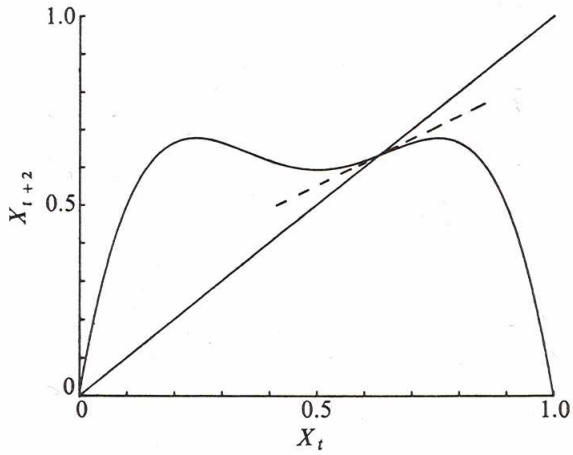


Fig. 2 The map relating X_{t+2} to X_t, obtained by two iterations of equation (3). This figure is for the case (*a*) of Fig. 1, $a = 2.707$: the basic fixed point is stable, and it is the only point at which $F^{(2)}(X)$ intersects the 45° line (where its slope, shown by the dashed line, is less steep than 45°).

is, fixed points with period 2) may now be written as X^*_2, and found either algebraically from

$$X^*_2 = F^{(2)}(X^*_2) \tag{9}$$

or graphically from the intersection between the map $F^{(2)}(X)$ and the 45° line, as shown in Figs 2 and 3. Clearly the equilibrium point X^* of equation (5) is a solution of equation (9); the basic fixed point of period 1 is a degenerate case of a period 2 solution. We now make a simple, but crucial, observation[1]: the slope of the curve $F^{(2)}(X)$ at the point X^*, defined as $\lambda^{(2)}(X^*)$ and illustrated by the dashed lines in Figs 2 and 3, is the square of the corresponding slope of $F(X)$

$$\lambda^{(2)}(X^*) = [\lambda^{(1)}(X^*)]^2 \tag{10}$$

This fact can now be used to make plain what happens when the fixed point X^* becomes unstable. If the slope of $F(X)$ is less than −45° (that is, $|\lambda^{(1)}| < 1$), as illustrated by curve *a* in Fig. 1, then X^* is stable. Also, from equation (10), this implies $0 < \lambda^{(2)} < 1$ corresponding to the slope of $F^{(2)}$ at X^* lying between 0° and 45°, as shown in Fig. 2. As long as the fixed point X^* is stable, it provides the only non-trivial solution to equation (9). On the other hand, when $\lambda^{(1)}$ steepens beyond −45° (that is, $|\lambda^{(1)}| > 1$), as illustrated by curve *b* in Fig 1, X^* becomes unstable. At the same time, from equation (10) this implies $\lambda^{(2)} > 1$, corresponding to the slope of $F^{(2)}$ at X^* steepening beyond 45°, as shown in Fig. 3. As this happens, the curve $F^{(2)}(X)$ must develop a "loop", and two new fixed points of period 2 appear, as illustrated in Fig. 3.

In short, as the nonlinear function $F(X)$ in equation (1) becomes more steeply humped, the basic fixed point X^* may become unstable. At exactly the stage when this occurs, there are born two new and initially stable fixed points of period 2, between which the system alternates in a stable cycle of period 2. The sort of graphical analysis indicated by Figs 1, 2 and 3, along with the equation (10), is all that is needed to establish this generic result[1,4].

As before, the stability of this period 2 cycle depends on the slope of the curve $F^{(2)}(X)$ at the 2 points. (This slope is easily shown to be the same at both points[1,20], and more generally to be the same at all k points on a period k cycle.) Furthermore, as is clear by imagining the intermediate stages between Figs 2 and 3, this stability-determining slope has the value $\lambda = +1$ at the birth of the 2-point cycle, and then decreases through zero towards $\lambda = -1$ as the hump in $F(X)$ continues to steepen. Beyond this point the period 2 points will in turn become unstable, and bifurcate to give an initially stable cycle of period 4. This in turn gives way to a cycle of period 8, and thence to a hierarchy of bifurcating stable cycles of periods 16, 32, 64, . . ., 2^n. In each case, the way in which a stable cycle of period k becomes unstable, simultaneously bifurcating to produce a new and initially stable cycle of period $2k$, is basically similar to the process just adumbrated for $k=1$. A more full and rigorous account of the material covered so far is in ref. 1.

This "very beautiful bifurcation phenomenon"[22] is depicted in Fig. 4, for the example equation (3). It cannot be too strongly emphasised that the process is generic to most functions $F(X)$ with a hump of tunable steepness. Metropolis *et al.*[16] refer to this hierarchy of cycles of periods 2^n as the harmonics of the fixed point X^*.

Although this process produces an infinite sequence of cycles with periods 2^n ($n \to \infty$), the "window" of parameter values wherein any one cycle is stable progressively diminishes, so that the entire process is a convergent one, being bounded above by some critical parameter value. (This is true for most, but not all, functions $F(X)$: see equation (17) below.) This critical parameter value is a point of accumulation of period 2^n cycles. For equation (3) it is denoted a_c: $a_c = 3.5700\ldots$

Beyond this point of accumulation (for example, for $a > a_c$ in equation (3)) there are an infinite number of fixed points with different periodicities, and an infinite number of different periodic cycles. There are also an uncountable number of initial points X_0 which give totally aperiodic (although bounded) trajectories; no matter how long the time series generated by $F(X)$ is run out, the pattern never repeats. These facts may be established by a variety of methods[1,4,20,22,29]. Such a situation, where an infinite number of different orbits can occur, has been christened "chaotic" by Li and Yorke[20].

As the parameter increases beyond the critical value, at first all these cycles have even periods, with X_t alternating up and down between values above, and values below, the fixed point X^*. Although these cycles may in fact be very complicated (having a non-degenerate period of, say, 5,726 points before repeating), they will seem to the casual observer to be rather like a somewhat "noisy" cycle of period 2. As the parameter value continues to increase, there comes a stage (at $a = 3.6786\ldots$ for equation (3)) at which the first odd period cycle appears. At first these odd cycles have very long periods, but as the parameter value continues to increase cycles with smaller and smaller odd periods are picked up, until at last the three-point

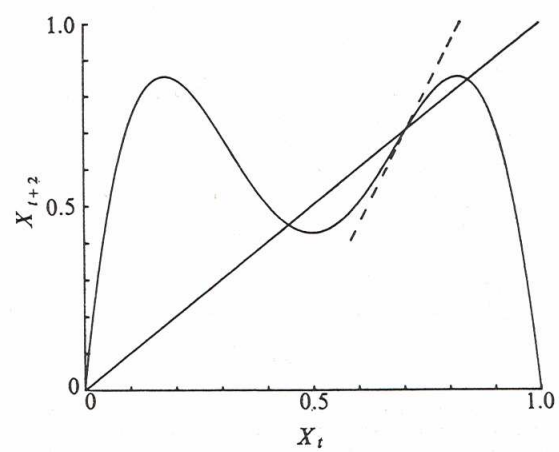


Fig. 3 As for Fig. 2, except that here $a = 3.414$, as in Fig. 1*b*. The basic fixed point is now unstable: the slope of $F^{(2)}(X)$ at this point steepens beyond 45°, leading to the appearance of two new solutions of period 2.

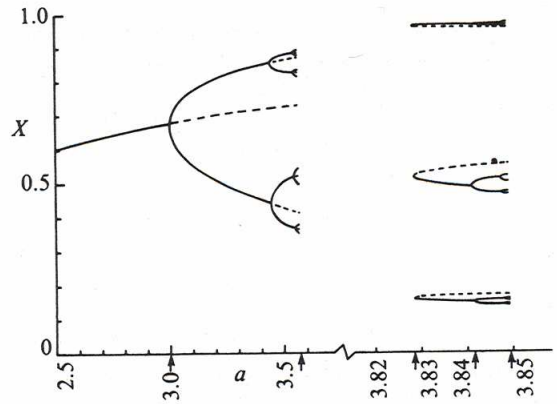


Fig. 4 This figure illustrates some of the stable (———) and unstable (— — — —) fixed points of various periods that can arise by bifurcation processes in equation (1) in general, and equation (3) in particular. To the left, the basic stable fixed point becomes unstable and gives rise by a succession of pitchfork bifurcations to stable harmonics of period 2^n; none of these cycles is stable beyond $a = 3.5700$. To the right, the two period 3 cycles appear by tangent bifurcation: one is initially unstable; the other is initially stable, but becomes unstable and gives way to stable harmonics of period 3×2^n, which have a point of accumulation at $a = 3.8495$. Note the change in scale on the a axis, needed to put both examples on the same figure. There are infinitely many other such windows, based on cycles of higher periods.

cycle appears (at a=3.8284 . . for equation (3)). Beyond this point, there are cycles with every integer period, as well as an uncountable number of asymptotically aperiodic trajectories: Li and Yorke[20] entitle their original proof of this result "Period Three Implies Chaos".

The term "chaos" evokes an image of dynamical trajectories which are indistinguishable from some stochastic process. Numerical simulations[12,15,21,23,25] of the dynamics of equation (3), (4) and other similar equations tend to confirm this impression. But, for smooth and "sensible" functions $F(X)$ such as in equations (3) and (4), the underlying mathematical fact is that for any specified parameter value there is one unique cycle that is stable, and that attracts essentially all initial points[22,29] (see ref. 4, appendix A, for a simple and lucid exposition). That is, there is one cycle that "owns" almost all initial points; the remaining infinite number of other cycles, along with the asymptotically aperiodic trajectories, own a set of points which, although uncountable, have measure zero.

As is made clear by Tables 3 and 4 below, any one particular stable cycle is likely to occupy an extraordinarily narrow window of parameter values. This fact, coupled with the long time it is likely to take for transients associated with the initial conditions to damp out, means that in practice the unique cycle is unlikely to be unmasked, and that a stochastic description of the dynamics is likely to be appropriate, in spite of the underlying deterministic structure. This point is pursued further under the heading "practical applications", below.

The main messages of this section are summarised in Table 1, which sets out the various domains of dynamical behaviour of the equations (3) and (4) as functions of the parameters, a and r respectively, that determine the severity of the nonlinear response. These properties can be understood qualitatively in a graphical way, and are generic to any well behaved $F(X)$ in equation (1).

We now proceed to a more detailed discussion of the mathematical structure of the chaotic regime for analytical functions, and then to the practical problems alluded to above and to a consideration of the behavioural peculiarities exhibited by non-analytical functions (such as those in the two right hand columns of Table 1).

Fine structure of the chaotic regime

We have seen how the original fixed point X^* bifurcates to give harmonics of period 2^n. But how do new cycles of period k arise?

The general process is illustrated in Fig. 5, which shows how period 3 cycles originate. By an obvious extension of the notation introduced in equation (8), populations three generations apart are related by

$$X_{t+3} = F^{(3)}(X_t) \tag{11}$$

If the hump in $F(X)$ is sufficiently steep, the threefold iteration will produce a function $F^{(3)}(X)$ with 4 humps, as shown in Fig. 5 for the $F(X)$ of equation (3). At first (for $a<3.8284$ . . in equation 3) the 45° line intersects this curve only at the single point X^* (and at X=0), as shown by the solid curve in Fig. 5. As the hump in $F(X)$ steepens, the hills and valleys in $F^{(3)}(X)$ become more pronounced, until simultaneously the first two valleys sink and the final hill rises to touch the 45° line, and then to intercept it at 6 new points, as shown by the dashed curve in Fig. 5. These 6 points divide into two distinct three-point cycles. As can be made plausible by imagining the intermediate stages in Fig. 5, it can be shown that the stability-determining slope of $F^{(3)}(X)$ at three of these points has a common value, which is $\lambda^{(3)}=+1$ at their birth, and thereafter steepens beyond $+1$: this period 3 cycle is never stable. The slope of $F^{(3)}(X)$ at the other three points begins at $\lambda^{(3)}=+1$, and then decreases towards zero, resulting in a stable cycle of period 3. As $F(X)$ continues to steepen, the slope $\lambda^{(3)}$ for this intially stable three-point cycle decreases beyond -1; the cycle becomes unstable, and gives rise by the bifurcation process discussed in the previous section to stable cycles of period 6, 12, 24, . . ., 3×2^n. This birth of a stable and unstable pair of period 3 cycles, and the subsequent harmonics which arise as the initially stable cycle becomes unstable, are illustrated to the right of Fig. 4.

Table 1 Summary of the way various "single-hump" functions $F(X)$, from equation (1), behave in the chaotic region, distinguishing the dynamical properties which are generic from those which are not

The function $F(X)$ of equation (1)	$aX(1-X)$	$X\exp[r(1-X)]$	aX; if $X<\frac{1}{2}$ $a(1-X)$; if $X>\frac{1}{2}$	λX; if $X<1$ λX^{1-b}; if $X>1$
Tunable parameter	a	r	a	b
Fixed point becomes unstable	3.0000	2.0000	1.0000*	2.0000
"Chaotic" region begins [point of accumulation of cycles of period 2^n]	3.5700	2.6924	1.0000	2.0000
First odd-period cycle appears	3.6786	2.8332	1.4142	2.6180
Cycle with period 3 appears [and therefore every integer period present]	3.8284	3.1024	1.6180	3.0000
"Chaotic" region ends	4.0000†	∞‡	2.000†	∞‡
Are there stable cycles in the chaotic region?	Yes	Yes	No	No

* Below this a value, $X = 0$ is stable.
† All solutions are attracted to $-\infty$ for a values beyond this.
‡ In practice, as r or b becomes large enough, X will eventually be carried so low as to be effectively zero, thus producing extinction in models of biological populations.

Table 2 Catalogue of the number of periodic points, and of the various cycles (with periods $k = 1$ up to 12), arising from equation (1) with a single-humped function $F(X)$

k	1	2	3	4	5	6	7	8	9	10	11	12
Possible total number of points with period k	2	4	8	16	32	64	128	256	512	1,024	2,048	4,096
Possible total number of points with non-degenerate period k	2	2	6	12	30	54	126	240	504	990	2,046	4,020
Total number of cycles of period k, including those which are degenerate and/or harmonics and/or never locally stable	2	3	4	6	8	14	20	36	60	108	188	352
Total number of non-degenerate cycles (including harmonics and unstable cycles)	2	1	2	3	6	9	18	30	56	99	186	335
Total number of non-degenerate, stable cycles (including harmonics)	1	1	1	2	3	5	9	16	28	51	93	170
Total number of non-degenerate, stable cycles whose basic period is k (that is, excluding harmonics)	1	–	1	1	3	4	9	14	28	48	93	165

There are, therefore, two basic kinds of bifurcation processes[1,4] for first order difference equations. Truly new cycles of period k arise in pairs (one stable, one unstable) as the hills and valleys of higher iterates of $F(X)$ move, respectively, up and down to intercept the 45° line, as typified by Fig. 5. Such cycles are born at the moment when the hills and valleys become tangent to the 45° line, and the initial slope of the curve $F^{(k)}$ at the points is thus $\lambda^{(k)}=+1$: this type of bifurcation may be called[1,4] a tangent bifurcation or a $\lambda=+1$ bifurcation. Conversely, an originally stable cycle of period k may become unstable as $F(X)$ steepens. This happens when the slope of $F^{(k)}$ at these period k points steepens beyond $\lambda^{(k)}=-1$, whereupon a new and initially stable cycle of period $2k$ is born in the way typified by Figs 2 and 3. This type of bifurcation may be called a pitchfork bifurcation (borrowing an image from the left hand side of Fig. 4) or a $\lambda=-1$ bifurcation[1,4].

Putting all this together, we conclude that as the parameters in $F(X)$ are varied the fundamental, stable dynamical units are cycles of basic period k, which arise by tangent bifurcation, along with their associated cascade of harmonics of periods $k2^n$, which arise by pitchfork bifurcation. On this basis, the constant equilibrium solution X^* and the subsequent hierarchy of stable cycles of periods 2^n is merely a special case, albeit a conspicuously important one (namely $k=1$), of a general phenomenon. In addition, remember[1,4,22,29] that for sensible, analytical functions (such as, for example, those in equations (3) and (4)) there is a unique stable cycle for each value of the parameter in $F(X)$. The entire range of parameter values ($1<a<4$ in equation (3), $0<r$ in equation (4)) may thus be regarded as made up of infinitely many windows of parameter values—some large, some unimaginably small—each corresponding to a single one of these basic dynamical units. Tables 3 and 4, below, illustrate this notion. These windows are divided from each other by points (the points of accumulation of the harmonics of period $k2^n$) at which the system is truly chaotic, with no attractive cycle: although there are infinitely many such special parameter values, they have measure zero on the interval of all values.

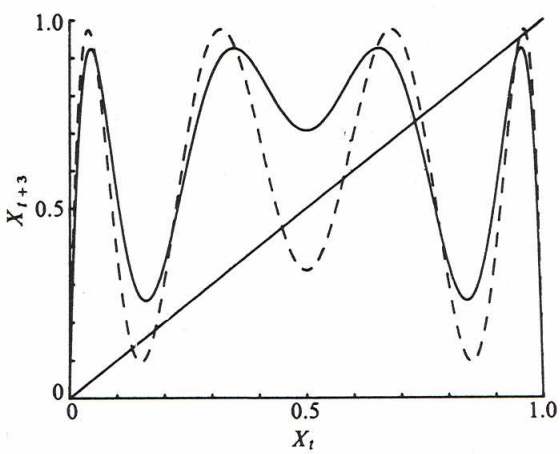


Fig. 5 The relationship between X_{t+3} and X_t, obtained by three iterations of equation (3). The solid curve is for $a = 3.7$, and only intersects the 45° line once. As a increases, the hills and valleys become more pronounced. The dashed curve is for $a = 3.9$, and six new period 3 points have appeared (arranged as two cycles, each of period 3).

How are these various cycles arranged along the interval of relevant parameter values? This question has to my knowledge been answered independently by at least 6 groups of people, who have seen the problem in the context of combinatorial theory[16,30], numerical analysis[13,14], population biology[1], and dynamical systems theory[22,31] (broadly defined).

A simple-minded approach (which has the advantage of requiring little technical apparatus, and the disadvantage of being rather clumsy) consists of first answering the question, how many period k points can there be? That is, how many distinct solutions can there be to the equation

$$X^*_k = F^{(k)}(X^*_k)? \tag{12}$$

If the function $F(X)$ is sufficiently steeply humped, as it will be once the parameter values are sufficiently large, each successive iteration doubles the number of humps, so that $F^{(k)}(X)$ has 2^{k-1} humps. For large enough parameter values, all these hills and valleys will intersect the 45° line, producing 2^k fixed points of period k. These are listed for $k \leqslant 12$ in the top row of Table 2. Such a list includes degenerate points of period k, whose period is a submultiple of k; in particular, the two period 1 points ($X=0$ and X^*) are degenerate solutions of equation (12) for all k. By working from left to right across Table 2, these degenerate points can be subtracted out, to leave the total number of non-degenerate points of basic period k, as listed in the second row of Table 2. More sophisticated ways of arriving at this result are given elsewhere[13,14,16,22,30,31].

For example, there eventually are $2^6=64$ points with period 6. These include the two points of period 1, the period 2 "harmonic" cycle, and the stable and unstable pair of triplets of points with period 3, for a total of 10 points whose basic period is a submultiple of 6; this leaves 54 points whose basic period is 6.

The 2^k period k points are arranged into various cycles of period k, or submultiples thereof, which appear in succession by either tangent or pitchfork bifurcation as the parameters in $F(X)$ are varied. The third row in Table 2 catalogues the total number of distinct cycles of period k which so appear. In the fourth row[14], the degenerate cycles are subtracted out, to give the total number of non-degenerate cycles of period k: these numbers must equal those of the second row divided by k. This fourth row includes the (stable) harmonics which arise by pitchfork bifurcation, and the pairs of stable–unstable cycles arising by tangent bifurcation. By subtracting out the cycles which are unstable from birth, the total number of possible stable cycles is given in row five; these figures can also be obtained by less pedestrian methods[13,16,30]. Finally we may subtract out the stable cycles which arise by pitchfork bifurcation, as harmonics of some simpler cycle, to arrive at the final

Table 3 A catalogue of the stable cycles (with basic periods up to 6) for the equation $X_{t+1} = aX_t(1-X_t)$

Period of basic cycle	*a* value at which: Basic cycle first appears	Basic cycle becomes unstable	Subsequent cascade of "harmonics" with period $k2^n$ all become unstable	Width of the range of *a* values over which the basic cycle, or one of its harmonics, is attractive
1	1.0000	3.0000	3.5700	2.5700
3	3.8284	3.8415	3.8495	0.0211
4	3.9601	3.9608	3.9612	0.0011
5(*a*)	3.7382	3.7411	3.7430	0.0048
5(*b*)	3.9056	3.9061	3.9065	0.0009
5(*c*)	3.99026	3.99030	3.99032	0.00006
6(*a*)	3.6265	3.6304	3.6327	0.0062
6(*b*)	3.937516	3.937596	3.937649	0.000133
6(*c*)	3.977760	3.977784	3.977800	0.000040
6(*d*)	3.997583	3.997585	3.997586	0.000003

row in Table 2, which lists the number of stable cycles whose basic period is k.

Returning to the example of period 6, we have already noted the five degenerate cycles whose periods are submultiples of 6. The remaining 54 points are parcelled out into one cycle of period 6 which arises as the harmonic of the only stable three-point cycle, and four distinct pairs of period 6 cycles (that is, four initially stable ones and four unstable ones) which arise by successive tangent bifurcations. Thus, reading from the foot of the column for period 6 in Table 2, we get the numbers 4, 5, 9, 14.

Using various labelling tricks, or techniques from combinatorial theory, it is also possible to give a generic list of the order in which the various cycles appear[1,13,16,22]. For example, the basic stable cycles of periods 3, 5, 6 (of which there are respectively 1, 3, 4) must appear in the order 6, 5, 3, 5, 6, 6, 5, 6: compare Tables 3 and 4. Metropolis *et al.*[16] give the explicit such generic list for all cycles of period $k \leqslant 11$.

As a corollary it follows that, given the most recent cycle to appear, it is possible (at least in principle) to catalogue all the cycles which have appeared up to this point. An especially elegant way of doing this is given by Smale and Williams[22], who show, for example, that when the stable cycle of period 3 first originates, the total number of other points with periods k, N_k, which have appeared by this stage satisfy the Fibonacci series, $N_k=2, 4, 5, 8, 12, 19, 30, 48, 77, 124, 200, 323$ for $k=1, 2, \ldots, 12$: this is to be contrasted with the total number of points of period k which will eventually appear (the top row of Table 2) as $F(X)$ continues to steepen.

Such catalogues of the total number of fixed points, and of their order of appearance, are relatively easy to construct. For any particular function $F(X)$, the numerical task of finding the windows of parameter values wherein any one cycle or its harmonics is stable is, in contrast, relatively tedious and inelegant. Before giving such results, two critical parameter values of special significance should be mentioned.

Hoppensteadt and Hyman[21] have given a simple graphical method for locating the parameter value in the chaotic regime at which the first odd period cycle appears. Their analytic recipe is as follows. Let α be the parameter which tunes the steepness of $F(X)$ (for example, $\alpha=a$ for equation (3), $\alpha=r$ for equation (4)), $X^*(\alpha)$ be the fixed point of period 1 (the non-trivial solution of equation (5)), and $X_{max}(\alpha)$ the maximum value attainable from iterations of equation (1) (that is, the value of $F(X)$ at its hump or stationary point). The first odd period cycle appears for that value of α which satisfies[21,31]

$$X^*(\alpha) = F^{(2)}(X_{max}(\alpha)) \tag{13}$$

As mentioned above, another critical value is that where the period 3 cycle first appears. This parameter value may be found numerically from the solutions of the third iterate of equation (1): for equation (3) it is[14] $a=1+\sqrt{8}$.

Myrberg[13] (for all $k \leqslant 10$) and Metropolis *et al.*[16]. (for all $k \leqslant 7$) have given numerical information about the stable cycles in equation (3). They do not give the windows of parameter values, but only the single value at which a given cycle is maximally stable; that is, the value of a for which the stability-determining slope of $F^{(k)}(X)$ is zero, $\lambda^{(k)}=0$. Since the slope of the k-times iterated map $F^{(k)}$ at any point on a period k cycle is simply equal to the product of the slopes of $F(X)$ at each of the points X^*_k on this cycle[1,8,20], the requirement $\lambda^{(k)}=0$ implies that $X=A$ (the stationary point of $F(X)$, where $\lambda^{(1)}=0$) is one of the periodic points in question, which considerably simplifies the numerical calculations.

For each basic cycle of period k (as catalogued in the last row of Table 2), it is more interesting to know the parameter values at which: (1) the cycle first appears (by tangent bifurcation); (2) the basic cycle becomes unstable (giving rise by successive pitchfork bifurcations to a cascade of harmonics of periods $k2^n$); (3) all the harmonics become unstable (the point of accumulation of the period $k2^n$ cycles). Tables 3 and 4 extend the work of May and Oster[1], to give this numerical information for equations (3) and (4), respectively. (The points of accumulation are not ground out mindlessly, but are calculated by a rapidly convergent iterative procedure, see ref. 1, appendix A.) Some of these results have also been obtained by Gumowski and Mira[32].

Practical problems

Referring to the paradigmatic example of equation (3), we can now see that the parameter interval $1<a<4$ is made up of a one-dimensional mosaic of infinitely many windows of a-values, in each of which a unique cycle of period k, or one of its harmonics, attracts essentially all initial points. Of these windows, that for $1<a<3.5700\ldots$ corresponding to $k=1$ and its harmonics is by far the widest and most conspicuous. Beyond the first point of accumulation, it can be seen from Table 3 that these windows are narrow, even for cycles of quite low periods, and the windows rapidly become very tiny as k increases.

As a result, there develops a dichotomy between the underlying mathematical behaviour (which is exactly determinable) and the "commonsense" conclusions that one would draw from numerical simulations. If the parameter a is held constant at one value in the chaotic region, and equation (3) iterated for an arbitrarily large number of generations, a density plot of the observed values of X_t on the interval 0 to 1 will settle into k equal spikes (more precisely, delta functions) corresponding to the k points on the stable cycle appropriate to this a-value. But for most a-values this cycle will have a fairly large period, and moreover it will typically take many thousands of generations before the transients associated with the initial conditions are damped out: thus the density plot produced by numerical simulations usually looks like a sample of points taken from some continuous distribution.

An especially interesting set of numerical computations are due to Hoppensteadt (personal communication) who has combined many iterations to produce a density plot of X_t for each one of a sequence of a-values, gradually increasing from 3.5700 . . to 4. These results are displayed as a movie. As can be expected from Table 3, some of the more conspicuous cycles

do show up as sets of delta functions: the 3-cycle and its first few harmonics; the first 5-cycle; the first 6-cycle. But for most values of a the density plot looks like the sample function of a random process. This is particularly true in the neighbourhood of the a-value where the first odd cycle appears ($a=3.6786\ldots$), and again in the neighbourhood of $a=4$: this is not surprising, because each of these locations is a point of accumulation of points of accumulation. Despite the underlying discontinuous changes in the periodicities of the stable cycles, the observed density pattern tends to vary smoothly. For example, as a increases toward the value at which the 3-cycle appears, the density plot tends to concentrate around three points, and it smoothly diffuses away from these three points after the 3-cycle and all its harmonics become unstable.

I think the most interesting mathematical problem lies in designing a way to construct some approximate and "effectively continuous" density spectrum, despite the fact that the exact density function is determinable and is always a set of delta functions. Perhaps such techniques have already been developed in ergodic theory[33] (which lies at the foundations of statistical mechanics), as for example in the use of "coarse-grained observers". I do not know.

Such an effectively stochastic description of the dynamical properties of equation (4) for large r has been provided[28], albeit by tactical tricks peculiar to that equation rather than by any general method. As r increases beyond about 3, the trajectories generated by this equation are, to an increasingly good approximation, almost periodic with period $(1/r)\exp(r-1)$.

The opinion I am airing in this section is that although the exquisite fine structure of the chaotic regime is mathematically fascinating, it is irrelevant for most practical purposes. What seems called for is some effectively stochastic description of the deterministic dynamics. Whereas the various statements about the different cycles and their order of appearance can be made in generic fashion, such stochastic description of the actual dynamics will be quite different for different $F(X)$: witness the difference between the behaviour of equation (4), which for large r is almost periodic "outbreaks" spaced many generations apart, versus the behaviour of equation (3), which for $a\rightarrow 4$ is not very different from a series of Bernoulli coin flips.

Mathematical curiosities

As discussed above, the essential reason for the existence of a succession of stable cycles throughout the "chaotic" regime is that as each new pair of cycles is born by tangent bifurcation (see Fig. 5), one of them is at first stable, by virtue of the way the smoothly rounded hills and valleys intercept the 45° line. For analytical functions $F(X)$, the only parameter values for which the density plot or "invariant measure" is continuous and truly ergodic are at the points of accumulation of harmonics, which divide one stable cycle from the next. Such exceptional parameter values have found applications, for example, in the use of equation (3) with $a=4$ as a random number generator[34,35]: it has a continuous density function proportional to $[X(1-X)]^{-\frac{1}{2}}$ in the interval $0<X<1$.

Non-analytical functions $F(X)$ in which the hump is in fact a spike provide an interesting special case. Here we may imagine spikey hills and valleys moving to intercept the 45° line in Fig. 5, and it may be that both the cycles born by tangent bifurcation are unstable from the outset (one having $\lambda^{(k)}>1$, the other $\lambda^{(k)}<-1$), for all $k>1$. There are then no stable cycles in the chaotic regime, which is therefore literally chaotic with a continuous and truly ergodic density distribution function.

One simple example is provided by

$$\begin{aligned} X_{t+1} &= aX_t;\ \text{if } X_t<\tfrac{1}{2} \\ X_{t+1} &= a(1-X_t);\ \text{if } X_t>\tfrac{1}{2} \end{aligned} \tag{14}$$

defined on the interval $0<X<1$. For $0<a<1$, all trajectories are attracted to $X=0$; for $1<a<2$, there are infinitely many periodic orbits, along with an uncountable number of aperiodic trajectories, none of which are locally stable. The first odd period cycle appears at $a=\sqrt{2}$, and all integer periods are represented beyond $a=(1+\sqrt{5})/2$. Kac[36] has given a careful discussion of the case $a=2$. Another example, this time with an extensive biological pedigree[1-3], is the equation

$$\begin{aligned} X_{t+1} &= \lambda X_t;\ \text{if } X_t<1 \\ X_{t+1} &= \lambda X_t^{1-b};\ \text{if } X_t>1 \end{aligned} \tag{15}$$

If $\lambda>1$ this possesses a globally stable equilibrium point for $b<2$. For $b>2$ there is again true chaos, with no stable cycles: the first odd cycle appears at $b=(3+\sqrt{5})/2$, and all integer periods are present beyond $b=3$. The dynamical properties of equations (14) and (15) are summarised to the right of Table 2.

The absence of analyticity is a necessary, but not a sufficient, condition for truly random behaviour[31]. Consider, for example,

$$\begin{aligned} X_{t+1} &= (a/2)X_t;\ \text{if } X_t<\tfrac{1}{2} \\ X_{t+1} &= aX_t(1-X_t)\ ;\ \text{if } X_t>\tfrac{1}{2} \end{aligned} \tag{16}$$

This is the parabola of equation (3) and Fig. 1, but with the left hand half of $F(X)$ flattened into a straight line. This equation does possess windows of a values, each with its own stable cycle, as described generically above. The stability-determining slopes $\lambda^{(k)}$ vary, however, discontinuously with the parameter a, and the widths of the simpler stable regions are narrower than for equation (3): the fixed point becomes unstable at $a=3$; the point of accumulation of the subsequent harmonics is at $a=3.27\ldots$; the first odd cycle appears at $a=3.44\ldots$; the 3-point cycle at $a=3.67\ldots$ (compare the first column in Table 1).

These eccentricities of behaviour manifested by non-analytical functions may be of interest for exploring formal questions in ergodic theory. I think, however, that they have no relevance to models in the biological and social sciences, where functions such as $F(X)$ should be analytical. This view is elaborated elsewhere[37].

As a final curiosity, consider the equation

$$X_{t+1} = \lambda X_t[1+X_t]^{-\beta} \tag{17}$$

Table 4 Catalogue of the stable cycles (with basic periods up to 6) for the equation $X_{t+1} = X_t\exp[r(1-X_t)]$

Period of basic cycle	r value at which: Basic cycle first appears	Basic cycle becomes unstable	Subsequent cascade of "harmonics" with period $k2^n$ *all* become unstable	Width of the range of r values over with the basic cycle, or one of its harmonics, is attractive
1	0.0000	2.0000	2.6924	2.6924
3	3.1024	3.1596	3.1957	0.0933
4	3.5855	3.6043	3.6153	0.0298
5(*a*)	2.9161	2.9222	2.9256	0.0095
5(*b*)	3.3632	3.3664	3.3682	0.0050
5(*c*)	3.9206	3.9295	3.9347	0.0141
6(*a*)	2.7714	2.7761	2.7789	0.0075
6(*b*)	3.4558	3.4563	3.4567	0.0009
6(*c*)	3.7736	3.7745	3.7750	0.0014
6(*d*)	4.1797	4.1848	4.1880	0.0083

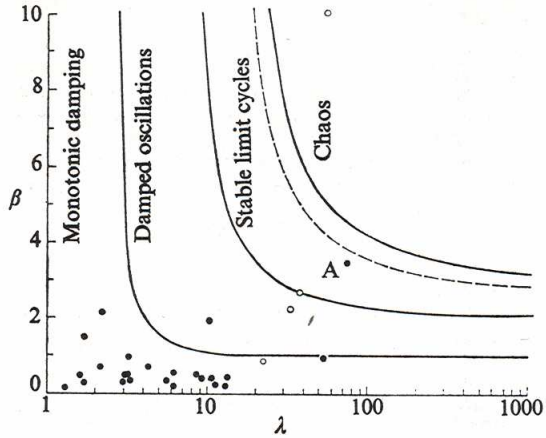


Fig. 6 The solid lines demarcate the stability domains for the density dependence parameter, β, and the population growth rate, λ, in equation (17); the dashed line shows where 2-point cycles give way to higher cycles of period 2^n. The solid circles come from analyses of life table data on field populations, and the open circles from laboratory populations (from ref. 3, after ref. 39).

This has been used to fit a considerable amount of data on insect populations[38,39]. Its stability behaviour, as a function of the two parameters λ and β, is illustrated in Fig. 6. Notice that for $\lambda < 7.39\ldots$ there is a globally stable equilibrium point for all β; for $7.39\ldots < \lambda < 12.50\ldots$ this fixed point becomes unstable for sufficiently large β, bifurcating to a stable 2-point cycle which is the solution for all larger β; as λ increases through the range $12.50\ldots < \lambda < 14.77\ldots$ various other harmonics of period 2^n appear in turn. The hierarchy of bifurcating cycles of period 2^n is thus truncated, and the point of accumulation and subsequent regime of chaos is not achieved (even for arbitrarily large β) until $\lambda > 14.77\ldots$

Applications

The fact that the simple and deterministic equation (1) can possess dynamical trajectories which look like some sort of random noise has disturbing practical implications. It means, for example, that apparently erratic fluctuations in the census data for an animal population need not necessarily betoken either the vagaries of an unpredictable environment or sampling errors: they may simply derive from a rigidly deterministic population growth relationship such as equation (1). This point is discussed more fully and carefully elsewhere[1].

Alternatively, it may be observed that in the chaotic regime arbitrarily close initial conditions can lead to trajectories which, after a sufficiently long time, diverge widely. This means that, even if we have a simple model in which all the parameters are determined exactly, long term prediction is nevertheless impossible. In a meteorological context, Lorenz[15] has called this general phenomenon the "butterfly effect": even if the atmosphere could be described by a deterministic model in which all parameters were known, the fluttering of a butterfly's wings could alter the initial conditions, and thus (in the chaotic regime) alter the long term prediction.

Fluid turbulence provides a classic example where, as a parameter (the Reynolds number) is tuned in a set of deterministic equations (the Navier–Stokes equations), the motion can undergo an abrupt transition from some stable configuration (for example, laminar flow) into an apparently stochastic, chaotic regime. Various models, based on the Navier–Stokes differential equations, have been proposed as mathematical metaphors for this process[15,40,41]. In a recent review of the theory of turbulence, Martin[42] has observed that the one-dimensional difference equation (1) may be useful in this context. Compared with the earlier models[15,40,41], it has the disadvantage of being even more abstractly metaphorical, and the advantage of having a spectrum of dynamical behaviour which is more richly complicated yet more amenable to analytical investigation.

A more down-to-earth application is possible in the use of equation (1) to fit data[1,2,3,38,39,43] on biological populations with discrete, non-overlapping generations, as is the case for many temperate zone arthropods. Figure 6 shows the parameter values λ and β that are estimated[39] for 24 natural populations and 4 laboratory populations when equation (17) is fitted to the available data. The figure also shows the theoretical stability domains: a stable point; its stable harmonics (stable cycles of period 2^n); chaos. The natural populations tend to have stable equilibrium point behaviour. The laboratory populations tend to show oscillatory or chaotic behaviour; their behaviour may be exaggeratedly nonlinear because of the absence, in a laboratory setting, of many natural mortality factors. It is perhaps suggestive that the most oscillatory natural population (labelled *A* in Fig. 6) is the Colorado potato beetle, whose present relationship with its host plant lacks an evolutionary pedigree. These remarks are only tentative, and must be treated with caution for several reasons. Two of the main caveats are that there are technical difficulties in selecting and reducing the data, and that there are no single species populations in the natural world: to obtain a one-dimensional difference equation by replacing a population's interactions with its biological and physical environment by passive parameters (such as λ and β) may do great violence to the reality.

Some of the many other areas where these ideas have found applications were alluded to in the second section, above[5–11]. One aim of this review article is to provoke applications in yet other fields.

Related phenomena in higher dimensions

Pairs of coupled, first-order difference equations (equivalent to a single second-order equation) have been investigated in several contexts[4,44–46], particularly in the study of temperate zone arthropod prey–predator systems[2–4,23,47]. In these two-dimensional systems, the complications in the dynamical behaviour are further compounded by such facts as: (1) even for analytical functions, there can be truly chaotic behaviour (as for equations (14) and (15)), corresponding to so-called "strange attractors"; and (2) two or more different stable states (for example, a stable point and a stable cycle of period 3) can occur together for the same parameter values[4]. In addition, the manifestation of these phenomena usually requires less severe nonlinearities (less steeply humped $F(X)$) than for the one-dimensional case.

Similar systems of first-order ordinary differential equations, or two coupled first-order differential equations, have much simpler dynamical behaviour, made up of stable and unstable points and limit cycles[48]. This is basically because in continuous two-dimensional systems the inside and outside of closed curves can be distinguished; dynamic trajectories cannot cross each other. The situation becomes qualitatively more complicated, and in many ways analogous to first-order difference equations, when one moves to systems of three or more coupled, first-order ordinary differential equations (that is, three-dimensional systems of ordinary differential equations). Scanlon (personal communication) has argued that chaotic behaviour and "strange attractors", that is solutions which are neither points nor periodic orbits[48], are typical of such systems. Some well studied examples arise in models for reaction–diffusion systems in chemistry and biology[49], and in the models of Lorenz[15] (three dimensions) and Ruelle and Takens[40] (four dimensions) referred to above. The analysis of these systems is, by virtue of their higher dimensionality, much less transparent than for equation (1).

An explicit and rather surprising example of a system which

has recently been studied from this viewpoint is the ordinary differential equations used in ecology to describe competing species. For one or two species these systems are very tame: dynamic trajectories will converge on some stable equilibrium point (which may represent coexistence, or one or both species becoming extinct). As Smale[50] has recently shown, however, for 3 or more species these general equations can, in a certain reasonable and well-defined sense, be compatible with any dynamical behaviour. Smale's[50] discussion is generic and abstract: a specific study of the very peculiar dynamics which can be exhibited by the familiar Lotka-Volterra equations once there are 3 competitors is given by May and Leonard[51].

Conclusion

In spite of the practical problems which remain to be solved, the ideas developed in this review have obvious applications in many areas.

The most important applications, however, may be pedagogical.

The elegant body of mathematical theory pertaining to linear systems (Fourier analysis, orthogonal functions, and so on), and its successful application to many fundamentally linear problems in the physical sciences, tends to dominate even moderately advanced University courses in mathematics and theoretical physics. The mathematical intuition so developed ill equips the student to confront the bizarre behaviour exhibited by the simplest of discrete nonlinear systems, such as equation (3). Yet such nonlinear systems are surely the rule, not the exception, outside the physical sciences.

I would therefore urge that people be introduced to, say, equation (3) early in their mathematical education. This equation can be studied phenomenologically by iterating it on a calculator, or even by hand. Its study does not involve as much conceptual sophistication as does elementary calculus. Such study would greatly enrich the student's intuition about nonlinear systems.

Not only in research, but also in the everyday world of politics and economics, we would all be better off if more people realised that simple nonlinear systems do not necessarily possess simple dynamical properties.

I have received much help from F. C. Hoppensteadt, H. E. Huppert, A. I. Mees, C. J. Preston, S. Smale, J. A. Yorke, and particularly from G. F. Oster. This work was supported in part by the NSF.

1 May, R. M., and Oster, G. F., *Am. Nat.*, **110** (in the press).
2 Varley, G. C., Gradwell, G. R., and Hassell, M. P., *Insect Population Ecology* (Blackwell, Oxford, 1973).
3 May, R. M. (ed.), *Theoretical Ecology: Principles and Applications* (Blackwell, Oxford, 1976).
4 Guckenheimer, J., Oster, G. F., and Ipaktchi, A., *Theor. Pop. Biol.* (in the press).
5 Oster, G. F., Ipaktchi, A., and Rocklin, I., *Theor. Pop. Biol.* (in the press).
6 Asmussen, M. A., and Feldman, M. W., *J. theor. Biol.* (in the press).
7 Hoppensteadt, F. C., *Mathematical Theories of Populations: Demographics, Genetics and Epidemics* (SIAM, Philadelphia, 1975).
8 Samuelson, P. A., *Foundations of Economic Analysis* (Harvard University Press, Cambridge, Massachusetts, 1947).
9 Goodwin, R. E., *Econometrica*, **19**, 1–17 (1951).
10 Baumol, W. J., *Economic Dynamics*, 3rd ed. (Macmillan, New York, 1970).
11 See, for example, Kemeny, J., and Snell, J. L., *Mathematical Models in the Social Sciences* (MIT Press, Cambridge, Massachusetts, 1972).
12 Chaundy, T. W., and Phillips, E., *Q. Jl Math. Oxford*, **7**, 74–80 (1936).
13 Myrberg, P. J., *Ann. Akad. Sc. Fennicae, A*, **I**, No. 336/3 (1963).
14 Myrberg, P. J., *Ann. Akad. Sc. Fennicae, A*, **I**, No. 259 (1958).
15 Lorenz, E. N., *J. Atmos. Sci.*, **20**, 130–141 (1963); *Tellus*, **16**, 1–11 (1964).
16 Metropolis, N., Stein, M. L., and Stein, P. R., *J. Combinatorial Theory*, **15(A)**, 25–44 (1973).
17 Maynard Smith, J., *Mathematical Ideas in Biology* (Cambridge University Press, Cambridge, 1968).
18 Krebs, C. J., *Ecology* (Harper and Row, New York, 1972).
19 May, R. M., *Am. Nat.*, **107**, 46–57 (1972).
20 Li, T-Y., and Yorke, J. A., *Am. Math. Monthly*, **82**, 985–992 (1975).
21 Hoppensteadt, F. C., and Hyman, J. M. (Courant Institute, New York University: preprint, 1975).
22 Smale, S., and Williams, R. (Department of Mathematics, Berkeley: preprint, 1976).
23 May, R. M., *Science*, **186**, 645–647 (1974).
24 Moran, P. A. P., *Biometrics*, **6**, 250–258 (1950).
25 Ricker, W. E., *J. Fish. Res. Bd. Can.*, **11**, 559–623 (1954).
26 Cook, L. M., *Nature*, **207**, 316 (1965).
27 Macfadyen, A., *Animal Ecology: Aims and Methods* (Pitman, London, 1963).
28 May, R. M., *J. theor. Biol.*, **51**, 511–524 (1975).
29 Guckenheimer, J., *Proc. AMS Symposia in Pure Math., XIV*, 95–124 (1970).
30 Gilbert, E. N., and Riordan, J., *Illinois J. Math.*, **5**, 657–667 (1961).
31 Preston, C. J. (King's College, Cambridge: preprint, 1976).
32 Gumowski, I., and Mira, C., *C. r. hebd. Séanc. Acad. Sci., Paris*, **281a**, 45–48 (1975); **282a**, 219–222 (1976).
33 Layzer, D., *Sci. Am.*, **233**(6), 56–69 (1975).
34 Ulam, S. M., *Proc. Int. Congr. Math.1950, Cambridge, Mass.; Vol. II*, pp. 264–273 (AMS, Providence R.I., 1950).
35 Ulam, S. M., and von Neumann, J., *Bull. Am. math. Soc.* (abstr.), **53**, 1120 (1947).
36 Kac, M., *Ann. Math.*, **47**, 33–49 (1946).
37 May, R. M., *Science*, **181**, 1074 (1973).
38 Hassell, M. P., *J. Anim. Ecol.*, **44**, 283–296 (1974).
39 Hassell, M. P., Lawton, J. H., and May, R. M., *J. Anim. Ecol.* (in the press).
40 Ruelle, D., and Takens, F., *Comm. math. Phys.*, **20**, 167–192 (1971).
41 Landau, L. D., and Lifshitz, E. M., *Fluid Mechanics* (Pergamon, London, 1959).
42 Martin, P. C., *Proc. Int. Conf. on Statistical Physics, 1975, Budapest* (Hungarian Acad. Sci., Budapest, in the press).
43 Southwood, T. R. E., in *Insects, Science and Society* (edit. by Pimentel, D.), 151–199 (Academic, New York, 1975).
44 Metropolis, N., Stein, M. L., and Stein, P. R., *Numer. Math.*, **10**, 1–19 (1967).
45 Gumowski, I., and Mira, C., *Automatica*, **5**, 303–317 (1969).
46 Stein, P. R., and Ulam, S. M., *Rospravy Mat.*, **39**, 1–66 (1964).
47 Beddington, J. R., Free, C. A., and Lawton, J. H., *Nature*, **255**, 58–60 (1975).
48 Hirsch, M. W., and Smale, S., *Differential Equations, Dynamical Systems and Linear Algebra* (Academic, New York, 1974).
49 Kolata, G. B., *Science*, **189**, 984–985 (1975).
50 Smale, S. (Department of Mathematics, Berkeley: preprint, 1976).
51 May, R. M., and Leonard, W. J., *SIAM J. Appl. Math.*, **29**, 243–253 (1975).