On the calculation of orthogonal vectors

By M. J. D. Powell*

Given an orthonormal basis, $d_1, d_2, \ldots, d_n$ of Euclidean n-space, and given some vector d_0 which is not orthogonal to d_n, this paper shows how to calculate, in $O(n^2)$ computer operations, a new orthonormal basis, $d_1^*, d_2^*, \ldots, d_n^*$, having the property that d_k^* is a linear combination of the k vectors $d_0, d_1, \ldots, d_{k-1}$. The method is useful because it reduces the amount of computer time that is needed by Rosenbrock's (1960) minimisation procedure. We show that any errors do not grow if the method is applied many times.

(Received May 1968)

Downloaded from https://academic.oup.com/comjnl/article/11/3/302/416473 by guest on 19 April 2024

1. Introduction

In an iteration of Rosenbrock's (1960) minimisation procedure we start with n orthonormal directions, $d_1, d_2 \ldots, d_n$, in Euclidean n-space, and we search along them in a cyclic way, until a non-zero step has been made along each one, and so we obtain a total displacement

$$d_0 = \alpha_1 d_1 + \alpha_2 d_2 + \ldots + \alpha_n d_n, \tag{1}$$

having the property that the multipliers $\alpha_i (i = 1, 2, \ldots, n)$ are all non-zero. For the next iteration a new set of orthonormal directions, $d_1^*, d_2^*, \ldots, d_n^*$, is obtained, and it is chosen so that, for $k = 1, 2, \ldots, n$, d_k^* is a linear combination of $d_0, d_1, \ldots, d_{k-1}$. In the original paper the new directions are calculated by the Gram–Schmidt orthonormalisation process (see Davis, 1962, for instance), which requires $O(n^3)$ multiplications, and the same method is followed in the recent I.C.I. Monograph (Box, Davies and Swann, 1968). However, it often happens that the rest of the computation of an iteration is $O(n^2)$, so it is useful that this note shows how the new directions can also be obtained in $O(n^2)$ computer operations.

The minimisation method due to Davies, Swann and Campey (Swann, 1964) is similar to Rosenbrock's procedure, but an important difference is that some of the multipliers α_i in equation (1) may be zero. To account for this case we depart from the above definition of d_k^* if it happens that $\alpha_k = \alpha_{k+1} = \ldots = \alpha_n = 0$, and instead we let $d_k^* = d_k$. Thus our device is also relevant to the D.S.C. algorithm.

A third application of the algorithm of this paper is to a method for solving systems of non-linear algebraic equations (Powell, 1968).

The algorithm is described in Section 2, and in Section 3 we prove two stability theorems to show that errors do not grow if the method is applied many times.

2. The algorithm

There are two versions of the algorithm because sometimes, for instance in Rosenbrock's method, the direction d_0 is specified by the values of the multipliers $\alpha_1, \alpha_2 \ldots, \alpha_n$ of equation (1). If the multipliers are given we use Algorithm A; otherwise we commence Algorithm B by calculating the numbers $\alpha_i (i = 1, 2, \ldots, n)$ from the scalar products

$$\alpha_i = (d_i, d_0), \; i = 1, 2, \ldots, n. \tag{2}$$

Next, in both algorithms, we inspect the sequence of multipliers $\alpha_1, \alpha_2, \ldots, \alpha_n$, and we let α_k be the last non-zero multiplier of the sequence. Usually $k = n$, and this is always the case when the algorithms are applied to Rosenbrock's method. However, if $k < n$, we now define

$$d_i^* = d_i, \; i = k+1, k+2, \ldots, n. \tag{3}$$

Next we set the quantities

$$\left.\begin{aligned} t &= k \\ s &= \alpha_k^2 \\ \sigma &= \alpha_k d_k \end{aligned}\right\} \tag{4}$$

in order to start an iterative process, which calculates the new directions $d_2^*, d_3^*, \ldots d_k^*$. In fact the index t is used to count the iterations, and we finish iterating if $t = 1$. If $t > 1$ we calculate d_t^* from the formula

$$d_t^* = (s d_{t-1} - \alpha_{t-1}\sigma)/[s(s + \alpha_{t-1}^2)]^{1/2}. \tag{5}$$

Before starting the next iteration we decrease t by one, and we add the quantities α_t^2 and $\alpha_t d_t$ to s and to σ, where now the subscripts have the new value of t.

Finally in Algorithm A we set

$$d_1^* = \sigma/\sqrt{s}, \tag{6}$$

and in Algorithm B we set

$$d_1^* = d_0/||d_0||_2. \tag{7}$$

Note that the two algorithms give identical results if the directions $d_1, d_2, \ldots, d_n$ are orthonormal, and if we use exact arithmetic.

To understand the operations of the above procedure we write out the vectors obtained by Algorithm A as shown in (8)

* *Mathematics Branch, Theoretical Physics Division, Atomic Energy Research Establishment, Harwell, Berkshire*

$$\left.\begin{aligned}
d_1^* &= \frac{\alpha_1 d_1 + \alpha_2 d_2 + \ldots + a_k d_k}{(\alpha_1^2 + \alpha_2^2 + \ldots + \alpha_k^2)^{1/2}} \\
d_2^* &= \frac{(\alpha_2^2 + \alpha_3^2 + \ldots + \alpha_k^2) d_1 - \alpha_1(\alpha_2 d_2 + \alpha_3 d_3 + \ldots + \alpha_k d_k)}{[(\alpha_2^2 + \alpha_3^2 + \ldots + \alpha_k^2)(\alpha_1^2 + \alpha_2^2 + \ldots + \alpha_k^2)]^{1/2}} \\
d_3^* &= \frac{(\alpha_3^2 + \alpha_4^2 + \ldots + \alpha_k^2) d_2 - \alpha_2(\alpha_3 d_3 + \alpha_4 d_4 + \ldots + \alpha_k d_k)}{[(\alpha_3^2 + \alpha_4^2 + \ldots + \alpha_k^2)(\alpha_2^2 + \alpha_3^2 + \ldots + \alpha_k^2)]^{1/2}} \\
&\;\;\vdots \\
d_k^* &= \frac{\alpha_k^2 d_{k-1} - \alpha_{k-1}(\alpha_k d_k)}{[\alpha_k^2(\alpha_{k-1}^2 + \alpha_k^2)]^{1/2}} \\
d_i^* &= d_i,\, i = k+1, k+2, \ldots, n
\end{aligned}\right\} . \quad (8)$$

It is straightforward to verify that this new set of vectors is orthonormal, and satisfies the condition that, for $t \leqslant k$, d_t^* is a linear combination of $d_0, d_1, \ldots, d_{t-1}$.

3. Stability

Whenever the algorithms are applied there are some errors due to the limited precision of the computer that is used for the calculation. We consider the errors in this section, and to help the analysis we introduce some matrix notation. We let D and D^* be the matrices

$$\left.\begin{aligned} D &= (d_1, d_2, \ldots, d_n) \\ D^* &= (d_1^*, d_2^*, \ldots, d_n^*) \end{aligned}\right\}, \quad (9)$$

so, for example, d_1 is the first column of D.

Using this notation we write equations (8) as

$$D^* = D\Omega, \quad (10)$$

where the elements of the matrix Ω are functions of the parameters $\alpha_1, \alpha_2, \ldots, \alpha_n$. It is important to notice that the parameters occur in such a way that (i) Ω is an orthogonal matrix, and (ii) there is no cancellation in the formulae defining its elements. Therefore there will be no large errors if the method of this paper is applied only a few times.

However, the details of Rosenbrock's method show that the algorithms may be used iteratively, in such a way that the directions $d_1^*, d_2^*, \ldots, d_n^*$ replace the directions $d_1, d_2, \ldots, d_n$ on every iteration. Therefore the question of stability is important, and we must consider the possibility of the growth of errors. The point is that, due to previous calculation, the directions $d_1, d_2, \ldots, d_n$ will not be exactly orthonormal, and the deviations will cause the vectors $d_1^*, d_2^*, \ldots, d_n^*$ to depart from orthonormality as well. If the resultant errors in $d_1^*, d_2^*, \ldots, d_n^*$ tend to be larger than those in $d_1, d_2, \ldots, d_n$, we have an unstable process, and it can happen that a sequence of iterations causes negligible errors to grow to an unacceptable size.

Fortunately the algorithms of this paper are not unstable, and to prove this fact we measure the deviation from orthonormality of the vectors $d_1, d_2, \ldots, d_n$ by the number

$$\Delta(D) = \sum_{i=1}^{n} \sum_{j=1}^{n} [(d_i, d_j) - \delta_{ij}]^2, \quad (11)$$

where δ_{ij} is the Kronecker delta,

$$\delta_{ij} = \begin{cases} 0, i \neq j \\ 1, i = j . \end{cases} \quad (12)$$

Two theorems show that, even if $d_1, d_2, \ldots, d_n$ satisfy no orthonormality conditions, the value of $\Delta(D^*)$ will not exceed the value of $\Delta(D)$ if either Algorithm A or Algorithm B is applied. The result depends on the assumption that there is no error in the calculation of the new directions, which is tolerable because we are considering the effect of previous errors.

THEOREM 1

Given any n directions $d_1, d_2, \ldots, d_n$, and any n multipliers $\alpha_1, \alpha_2, \ldots, \alpha_n$, then the equation

$$\Delta(D^*) = \Delta(D) \quad (13)$$

holds, where the matrices D and D^* are defined by equations (8) and (9).

PROOF

The definition (11) implies the equation

$$\begin{aligned} \Delta(D) &= \text{tr}\{(D^T D - I)^T (D^T D - I)\} \\ &= \text{tr}(D^T D D^T D) - 2tr(D^T D) + tr(I), \quad (14) \end{aligned}$$

where the notation tr (.) stands for the trace of the matrix inside the brackets. Therefore, because D^* is related to D by equation (10), and because Ω is an orthogonal matrix, we obtain the identity

$$\Delta(D^*) = tr(\Omega^T D^T D D^T D \Omega) - 2tr(\Omega^T D^T D \Omega) + tr(I). \quad (15)$$

Now the definition of a trace and the orthogonality of Ω imply that expressions (14) and (15) are identical, so the theorem is proved.

An immediate corollary of Theorem 1 is that equation (13) holds for Algorithm A. But we cannot

Downloaded from https://academic.oup.com/comjnl/article/11/3/302/416473 by guest on 19 April 2024

deduce the same result for Algorithm B unless expressions (6) and (7) are identical. Instead we have the more favourable theorem:

THEOREM 2

The matrix D^*, defined by equation (9) and Algorithm B of Section 2, is such that, for any set of directions $d_1, d_2, \ldots, d_n$, we have the inequality

$$\Delta(D^*) \leqslant \Delta(D). \tag{16}$$

PROOF

For Algorithm B the multipliers $\alpha_1, \alpha_2, \ldots, \alpha_n$ are defined by equation (2), d_1^* is defined by equation (7), and for $t \geqslant 2$ the definition of d_t^* is given by equation (8). It follows that the vector d_1^* is normalised, and is orthogonal to the other calculated vectors, $d_2^*, d_3^*, \ldots, d_n^*$. Therefore from the definition (11) we obtain the equation

$$\Delta(D^*) = \sum_{i=2}^{n} \sum_{j=2}^{n} [(d_i^*, d_j^*) - \delta_{ij}]^2. \tag{17}$$

Let Ω be the orthogonal matrix that is consistent with equations (8) and (10), and let

$$\tilde{D} = D\Omega. \tag{18}$$

Because the last $(n - 1)$ columns of $\tilde{D}$ are identical to the last $(n - 1)$ columns of D^*, equation (17) provides the inequality

$$\Delta(\tilde{D}) \geqslant \Delta(D^*). \tag{19}$$

Now by the argument of Theorem 1 we obtain the identity

$$\Delta(\tilde{D}) = \Delta(D), \tag{20}$$

so Theorem 2 is an immediate consequence of inequality (19).

Both the algorithms and the stability theorems have been checked by numerical examples, and they confirm that the method of this paper is satisfactory.

References

BOX, M. J., DAVIES, D., and SWANN, W. H. (1968). *Optimization Techniques*, I.C.I. Monograph No. 5, to be published by Oliver and Boyd Ltd.

DAVIS, P. J. (1962). Orthonormalizing codes in numerical analysis (from *Survey of numerical analysis*, edited by J. Todd, McGraw-Hill (New York), p. 347).

POWELL, M. J. D. (1968). A Fortran subroutine for solving systems of non-linear algebraic equations, (to be published as an A.E.R.E. Report).

ROSENBROCK, H. H. (1960). An automatic method for finding the greatest or the least value of a function, *Computer Journal*, Vol. 3, p. 175.

SWANN, W. H. (1964). *Report on the development of a new direct search method of optimization*, I.C.I. Ltd., Central Instrument Laboratory, Research Note No. 64/3.

Book Review

Asymptotic Methods in the Theory of Linear Differential Equations, by S. F. FESHCHENKO, N. I. SHKIL' and L. D. NIKOLENKO, 1967; 270 pp. (Barking, Essex: *Elsevier Publishing Co. Ltd.*, 140s.)

This book was originally written in Russian and translated very efficiently by Scripta Technica, Inc. It contains an advanced treatment to a rather specialised topic and is only suitable for mathematical graduates. It describes some methods for obtaining approximate solutions to linear differential equations in which the coefficients vary slowly in time. If t denotes the normal independent variable and ϵ is some small positive parameter, then a slow time variable $\tau = \epsilon t$ is introduced, and equations are considered whose coefficients are functions of τ. This occurs, for example, in a single equation in which the coefficient of the highest derivative is small, or in Sturm–Liouville equations in which large values of the eigenvalue are being considered. The basis of the methods is to find an approximate solution in the form of an asymptotic series in ϵ combined with an oscillatory term.

After a brief historical introduction, in Chapter 1 is given a method for solving a single second-order ordinary equation in cases of both resonance and non-resonance. This is generalised in Chapter 2 to simultaneous second-order equations. In Chapter 3 it is shown how to decompose a set of simultaneous first-order equations $\dot{x} = Ax + b$ into a number of sets of mutually independent equations, and the complications of solving these when A has some multiple characteristic values with non-linear elementary divisors is given in Chapter 4. The method is generalised in Chapter 5 to the solution of equations in Banach space. Finally, in Chapter 6 the solution of a particular hyperbolic partial differential equation is described in which the method of separation of variables is used to produce a system of ordinary differential equations which are solved by the method given in Chapter 5.

The book is well printed and only a small number of misprints were noticed.

V. E. PRICE (London)

Downloaded from https://academic.oup.com/comjnl/article/11/3/302/416473 by guest on 19 April 2024