JOURNAL OF COMPUTATIONAL PHYSICS 111, 410-413 (1994)

NOTE

"Diagonal Shadow"—A Quasi-Newton Iteration in Spectral Domain

1. INTRODUCTION

The main idea of the approach proposed below is to perform a rotation in the functional space to such an orthonormal basis that the Fréchet derivative of a given operator, whose inversion is necessary for Newton's method, becomes nearly diagonal. This derivative is then approximated by a diagonal (multiplication) operator, so that its inversion becomes trivial. The suitable basis can be chosen among the classical orthonormal sets, so that rotation to it is easily performed (e.g., by fast Fourier transform). The technique may be useful for some linear and nonlinear operator equations, including "black-box" operators.

As an example of the latter, consider a free-boundary problem in fluid dynamics which can be solved by the following iterative process [1, 2]:

(1) for a given shape of the free boundary compute the flow field by solving the Navier–Stokes equations;

(2) knowing the flow field, calculate the normal stress at the free boundary;

(3) if this normal stress is not continuous across the free boundary, adjust the boundary shape using the local normal stress difference as a "driving force";

(4) go to 1 and repeat until the normal stress difference is zero.

Step 3 in the above algorithm is usually the "bottleneck": the relaxation parameter τ, which multiplies the normal stress difference to give the shape increment, must normally be very small in order to achieve convergence. Let us describe the effect of the shape of the free boundary ($f(x)$ in parametric form) on the resulting distribution of the normal stress difference as action of some abstract operator (denoted by A). The result of this action (the normal stress difference) is actually determined by the overall flow field, governed by the Navier–Stokes equations and boundary conditions at all boundaries. Step 3 can be viewed as solving the operator equation $A[f(x)]=0$ by simple iteration

$$f^{n+1}=f^n+\tau A[f^n]. \tag{1}$$

Clearly, A is so complicated that it should be viewed as a "black-box" operator for all practical purposes. Nevertheless, we ask: can (1) be replaced by a more efficient procedure, so that the above operator equation could be solved, i.e., the whole algorithm described above could converge, after a smaller number of iterations? The task seems impossible at first; we shall see, however, that significant progress *can* be made, with the resulting acceleration of convergence of the whole algorithm by a factor of 10 or more.

An approach that is applicable to (some) black-box operators will also work with (some) conventional operators, and for the clarity of exposition it is convenient to introduce the idea of the method using a second-order differential operator, defined on a unit interval.

2. DESCRIPTION OF THE METHOD

Consider a two-point boundary-value problem (extension to higher dimensions will be obvious),

$$\begin{gathered} A[f(x)]=g(x), \qquad 0\leqslant x\leqslant 1, \\ f(0)=f(1)=0, \end{gathered} \tag{2}$$

where A is a second-order differential operator (linear or nonlinear). Let us consider such A that the problem (2), discretized on a grid $x_j=jh$, $h=1/N$, $j=0, 1, 2, \ldots, N$, can be solved by the simple iteration,

$$f^{n+1}=f^n+\tau\{A[f^n]-g\}, \qquad \tau>0. \tag{3}$$

Stability requirements will usually result in the restriction $\tau\leqslant O(h^2)$, and a very large number of iterations, $O(N^2)$, will be necessary for convergence. The high-frequency components of the error will decrease fast, but the low-frequency, smooth components will be decreasing very slowly.

In principle, Eq. (2) can also be solved by Newton's method

$$f^{n+1}=f^n-A_{f^n}^{-1}[A[f^n]-g], \tag{4}$$

where the linear operator A_{f^n} is the Fréchet derivative of $A[f]$ evaluated at $f=f^n$. Of course, if A is linear, $A_{f^n}=A$, Newton's method is equivalent to the direct inversion

0021-9991/94 $6.00
Copyright © 1994 by Academic Press, Inc.
All rights of reproduction in any form reserved.

$f = A^{-1}[g]$, and only one "iteration" is required. Newton's method, if it converges, requires few iterations, but the inversion of the operator A_{f^n} may be extremely time consuming.

If one could find an operator B which is sufficiently close to A_{f^n} but can be inverted easily, one could use (for linear and nonlinear A) the following quasi-Newton iteration

$$f^{n+1} = f^n - B^{-1}[A[f^n] - g]. \tag{5}$$

The operators which are easiest to invert are the diagonal ones, but A_{f^n}, being a second-order differential operator, is very far from being diagonal. A rotation in the functional space to the basis consisting of eigenfunctions of A_{f^n} would make A_{f^n} diagonal, and its inversion would be trivial. In fact, for a linear A and thus $A_{f^n} = A$, this sequence of operations would comprise the classical solution technique of eigenfunction expansions. However, only in rare cases are the eigenfunctions of A_{f^n} known. Although in many other cases they can be computed, such computation would normally require more effort than a solution of (2) by other means.

The main idea of the present approach is to perform a rotation in the functional space to an orthonormal basis $\{u_k\}$, which does not necessarily consist of the eigenfunctions of A_{f^n}, but (i) is sufficiently close, so that A_{f^n} is nearly diagonal in $\{u_k\}$, and (ii) the basis functions u_k are known and the rotation can be easily performed. Then we shall take for B^* (where * means "in the basis $\{u_k\}$") a *diagonal* operator whose values coincide with, or approximate (at least crudely), the diagonal values of $A^*_{f^n}$. That is, B^* is the "diagonal shadow" of $A^*_{f^n}$. Such B^* is easily invertible, and so (5) becomes an efficient method of iterative solution.

The basis $\{u_k\}$ can be chosen among the classical orthonormal sets of functions; simultaneously, the appropriate form of the scalar product (i.e., the density function) must be chosen. One of the simplest choices would be the orthonormal set

$$u_k = \sqrt{2} \sin k\pi \frac{j}{N}, \qquad j = 0, 1, 2, \ldots, N, \tag{6}$$

in which case the rotation to $\{u_k\}$ is performed by the discrete Fourier transform. If more than one alternative appears, it seems reasonable to choose the basis that would make A_{f^n} more nearly diagonal, i.e., the one in which the scalar products $(u_l, A_{f^n}[u_m])$, $l \neq m$, are smaller (by absolute value). Or simply to perform a few iterations in each likely candidate, and then choose the one in which convergence is the fastest.

Obviously, the proposed approach is a heuristic one. Some support for the notion that the eigenfunctions of different differential operators of the same order, with the same domain and boundary conditions, are "close" to each other can be found in the classical studies of the behavior of eigenfunctions [3], but the ultimate judgment should be based on the performance of the resulting numerical technique. Note that it is most important to approximate correctly the high-frequency eigenfunctions of A_{f^n}, since it is their behavior that causes instability of the simple iteration (3) unless τ is very small. These high-frequency eigenfunctions are well approximated by trigonometric functions for many different operators [3].

The diagonal values of B^*, denoted λ_k, can be found as diagonal values of A_{f^n} in $\{u_k\}$. If A is linear and thus $A_{f^n} = A$, we have $\lambda_k = (u_k, A[u_k])$. In general,

$$\lambda_k = (1/\varepsilon)(u_k, A[f^n + \varepsilon u_k] - A[f^n]), \tag{7}$$

where ε is a small number.

To save on computing of λ_k, one may compute a couple of values (say, for $k = 1$ and $k = N - 1$), and then use the knowledge of classical eigenvalue distributions such as the one for $A = d^2/dx^2$,

$$\lambda_k = \lambda_k^F \equiv -4N^2 \sin^2(k\pi/2N), \tag{8}$$

to produce an "empirical correlation" for λ_k as a function of k. Or even simply set $\lambda_k = \lambda_1 k^\alpha$ and then adjust λ_1 and α for the fastest convergence.

Having constructed the operator B^*, we can solve (2) by the following quasi-Newton iteration in the basis $\{u_k\}$ (here $\mathscr{R}$ denotes rotation to the basis $\{u_k\}$):

$$f^{n+1} = f^n - \mathscr{R}^{-1}\{B^{*-1}[\mathscr{R}\{A[f^n] - g\}]\}, \tag{9}$$

or, in detail,

(1) for a given nth approximation to the solution $f^n(x)$ compute the residual $A[f^n] - g \equiv r^n(x)$;

(2) compute its transform $p_k^n = (u_k, r^n)$;

(3) compute $q_k^n = p_k^n/\lambda_k$;

(4) transform back to the original basis and subtract from f^n to obtain f^{n+1}, i.e., $f^{n+1} = f^n - \sum_k q_k^n u_k$;

(5) go to step 1 and repeat until convergence.

Obviously, the technique is exceedingly simple. If, e.g., $\{u_k\}$ is given by (6), the required transforms are Fourier transforms and can be performed by FFT; a variety of chips are available which can perform FFT at very high speeds. The required programming work is minimal.

Each $-\lambda_k^{-1}$ can be viewed as the relaxation parameter τ_k of the simple iteration (3) for the kth "mode," and then the "diagonal-shadow" technique appears as an application of the simple iteration (3) to each mode separately, with different relaxation parameters. (There will be, of course, some mixing and generation of the modes unless A is a linear operator and $\{u_k\}$ is exactly its eigenfunction basis.) The essence of the technique is the ability to use a nearly optimal

relaxation parameter for each mode, and thus to reduce all components of the error equally rapidly. This simple meaning of λ_k, plus a natural requirement for λ_k to be a smooth function of k, and also the knowledge of this function for classical operators, afford an opportunity to adjust λ_k for optimal convergence, using numerical experiment and other available information. Note that the sign of τ (and thus also of λ_k) will often be obvious from the physics of the problem.

In particular, one can adjust λ_k *during* the iterations using the "secant algorithm." Since λ_k can be viewed as an approximation to the derivative of A with respect to the kth mode, one can find an improved value for λ_k^n, beginning with $n=2$, as

$$\lambda_k^n = \frac{p_k^n - p_k^{n-1}}{-q_k^{n-1}} = \lambda_k^{n-1} - \frac{p_k^n}{q_k^{n-1}} = \lambda_k^{n-1}\left(1 - \frac{p_k^n}{p_k^{n-1}}\right), \quad (10)$$

and then use this value to find q_k^n in the step 3 of (9).

Obviously, some value λ_k^1 is required to start the iterations; this value can be taken from the classical distributions such as (8), or inferred by other means (the influence of this initial value must be minimal anyhow). It would be prudent to use (10) only while the absolute value of the residual is not too small, and to keep λ_k constant thereafter. Otherwise, spurious values of λ_k may be obtained due to round-off errors; also, the Fréchet derivative A_{f^n} should be changing very little after f^n has approached the solution closely.

The above secant algorithm looks promising, but it should be kept in mind that the modes are not really independent. In extreme cases, their generation and mixing may result in completely unsuitable values for λ_k being computed from (10). The secant algorithm should thus be used with caution, the values of λ_k^n should be monitored and, perhaps, restricted to remain within some bounds chosen a priori, etc. It is likely to be most useful for the low-frequency modes, since the high-frequency modes are less dependent on the structure of A, and it should be possible to estimate λ_k for these modes by other means.

3. NUMERICAL EXAMPLES

The examples listed in Table I all correspond to problem (2); three-point finite-difference formulae, accurate to $O(h^2)$, were used to evaluate $A[f^n]$. In all these examples $N=32$, u_k are given by (6), and the rotation is performed by the discrete Fourier transform. The initial guess for all the examples was sufficiently bad, viz.,

$$f^1\left(\frac{j}{N}\right) = \begin{cases} 0 & \text{for } j=0, \quad j=N, \\ 1 & \text{for } 1 \leqslant j \leqslant 23, \quad 25 \leqslant j \leqslant N-1, \\ -1 & \text{for } j=24. \end{cases}$$

TABLE I

$A[f(x)]$	$g(x)$	λ_k (or method of solution if other than "diagonal shadow")	Number of iterations necessary to approach the exact solution within 10^{-4}
$\frac{d^2f}{dx^2}$	0	simple iteration (3) with $\tau = 1/2N^2$	1958
		$\lambda_k^F \equiv -4N^2 \sin^2(k\pi/2N)$	1
$\frac{d^2f}{dx^2} + \frac{1}{x}\frac{df}{dx} - \frac{f}{x^2}$	0	λ_k^F	21
		$(u_k, A[u_k])$	14
		$\lambda_k^F\left(1 + \frac{\beta - 1}{k}\right)$; $\beta \equiv \frac{(u_1, A[u_1])}{\lambda_1^F} = 1.45$	14
		$\lambda_k^F(1 + 1/k)$	11
$\frac{d^2f}{dx^2} + 5\left(\frac{1}{x}\frac{df}{dx} - \frac{f}{x^2}\right)$	0	λ_k^F	Did not converge
		$(u_k, A[u_k])$	Did not converge
		$2\lambda_k^F$	49
		$2\lambda_k^F(1 + 1/k)$	20
$\frac{d^2f}{dx^2} + 6f^2$	$A[x(1-x)]$	λ_k^F	10
	$A[4x(1-x)]$	λ_k^F	Did not converge
		$\lambda_k = \lambda_k^F$ for $k \geqslant 2$; λ_1 is adjusted by the secant algorithm (10) until $\lvert A[f^n] - g \rvert < 10^{-2}$ and then kept constant	10

The first two examples in Table I, with $A = d^2/dx^2$, are not really examples of the diagonal shadow. The first of these is an illustration of the slow convergence of the simple iteration (3), even with a parameter τ close to optimal, while the second, although utilizing the algorithm (9), is essentially a solution by an eigenfunction expansion, since $\{u_k\}$ is exactly the eigenbasis of A in this case.

All the other examples do utilize the diagonal-shadow approach. They show, in particular, that the theoretically appealing choice $\lambda_k = (u_k, A[u_k])$ is not always the best even for linear operators, and that the much simpler (and less expensive) "semi-empirical" formulae for λ_k, sometimes in conjunction with a judicious use of the secant algorithm (10), may yield better results. They also show that the diagonal shadow is capable of reducing the number of iterations, as compared to the simple iterative algorithm (3), by two orders of magnitude.

The discretization of the problem on a grid and the use of finite-differences are not, of course, essential for the present approach. One needs some way to obtain $A[f^n] - g$ in order to perform the iteration (9), but this way can be different in different situations, and not even necessarily numerical. It is conceivable that an analytical solution, in the form of an expansion in $\{u_k\}$, could be obtained along similar lines, using a symbolic manipulation software to perform the iteration.

4. APPLICATION TO A BLACK-BOX OPERATOR

Perhaps, the main advantage of the present approach is its applicability to black-box operators. A black-box operator A may represent some physical system or device, possibly of unknown structure, and the task may be to find such an input function f that the output is equal to g. In this case $A[f^n]$ would be obtained by measurement.

Another kind of a black-box operator may occur in large-scale computational problems, such as the one discussed in the Introduction. The initial approach to this problem [2] was to perform Step 3 as the simple iteration (1). This procedure did lead to convergence, but only when τ was sufficiently small, of order 10^{-3}, and a very large number of global iterations, of order 10^4, was required.

Then the diagonal-shadow approach was attempted. To obtain at least some information about the operator A, the behavior of the normal stress difference $A[f^n]$ was observed, using graphical output, during the iteration (1) with τ which was *not* sufficiently small, so the computation was unstable. This observation revealed that, as iterations progressed, the high-frequency modes were growing much faster than the low-frequency ones, which suggested that using different relaxation parameters for different modes might speed up the convergence. Hence the diagonal shadow was applied, with the discrete Fourier transform as $\mathscr{R}$, and $\lambda_k = \lambda_1 k^\alpha$, where the optimal values of α and λ_1 were to be determined by numerical experiment. It turned out that choosing $\alpha = 1$ leads to roughly uniform relaxation of all modes and allows one to use τ_1 (i.e., $-\lambda_1^{-1}$) which is about 20 times greater than the maximum stable τ in the simple iteration procedure. As a result, the number of global iterations necessary for convergence (and thus also the total CPU time, since inserting the two Fourier transforms in Step 3 adds very little) was reduced by an order of magnitude. The seemingly impossible problem posed in Introduction has been solved.

No claim is made here that the proposed approach can be applied to all possible operators. Nevertheless, the simplicity of the approach, and its potential to produce extremely powerful results when it does work, appear to justify further experimentation and study.

REFERENCES

1. G. Ryskin and L. G. Leal, *J. Comput. Phys.* **50**, 71 (1983).
2. G. Ryskin and L. G. Leal, *J. Fluid Mech.* **148**, 1 (1984).
3. R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Vol. 1 (Wiley, New York, 1953).

Received March 1990; revised November 2, 1993

GREGORY RYSKIN

Department of Chemical Engineering
Northwestern University
Evanston, Illinois 60208