INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 2, 43–60 (1982)

ON HIGHER ORDER MULTIGRID METHODS WITH APPLICATION TO A GEOTHERMAL RESERVOIR MODEL

JOHN GARY

Mathematics Department, Colorado School of Mines, Golden, CO 80401, U.S.A.

SUMMARY

This paper considers the multigrid iterative method applied to the solution of finite difference approximations to a linear second-order self-adjoint elliptic equation. It represents an extension of work by Dinar and Brandt. We compare two methods to obtain fourth-order convergence. The first is local error extrapolation developed by Brandt, the second is iterative improvement developed by Lindberg. This work considers non-separable problems, but only on a rectangular domain with Dirichlet boundary conditions. We consider test cases with non-smooth (i.e. discontinuous second derivatives) as well as smooth solutions. We also apply the multigrid method to an elliptic equation with non-separable coefficients which occurs in a geothermal model. In this case an analysis of the error fails to show any advantage in a fourth-order difference scheme over a second-order scheme. However, we do demonstrate that the multigrid iteration performs well on this problem. Also, this example shows that the multigrid iteration can be combined with iterative improvement to create an efficient fourth-order method for a non-separable elliptic equation which is coupled with a marching equation. Other work has found an advantage in this fourth-order scheme for a similar geothermal model.

KEY WORDS Multigrid Iterative Improvement Geothermal Model Alternating-direction-implicit Tau-extrapolation Higher-order-accuracy

INTRODUCTION

This paper is concerned with the solution of the following self-adjoint elliptic problem

$$\frac{\partial}{\partial x}\left(\sigma\frac{\partial U}{\partial x}\right)+\frac{\partial}{\partial y}\left(\sigma\frac{\partial U}{\partial y}\right)=r \tag{1}$$

where $\sigma=\sigma(x, y)$ and $r=r(x, y)$. Our results are restricted to a rectangular domain with Dirichlet boundary conditions (i.e. $U=g(x, y)$ on the boundary). We will assume the reader is familiar with the multigrid iteration for this problem as described by Brandt.[1] Our primary interest here is the solution of an elliptic equation coupled with a marching (time dependent) problem. Examples are the Navier–Stokes equations[2] or the following porous flow problem.[3]

$$\begin{aligned}\frac{\partial\theta}{\partial t}&=\frac{1}{\mathrm{Ra}}\left[\frac{\partial^2\theta}{\partial x^2}+\frac{\partial^2\theta}{\partial y^2}\right]-\frac{\partial\psi}{\partial x}-\frac{\partial\theta}{\partial x}\frac{\partial\psi}{\partial y}+\frac{\partial\theta}{\partial y}\frac{\partial\psi}{\partial x}\\ \frac{\partial}{\partial x}\left(\sigma\frac{\partial\psi}{\partial x}\right)&+\frac{\partial}{\partial y}\left(\sigma\frac{\partial\psi}{\partial y}\right)=-\frac{\partial\theta}{\partial x}\end{aligned} \tag{2}$$

Here $\theta(x, y, t)$ determines the temperature, $\psi(x, y, t)$ is the stream function and $\sigma=\sigma(\theta)$ is a known function of θ. Certain solutions of this system require considerable accuracy. For

0271–2091/82/010043–18$01.80

Received 29 January 1981
Revised 15 June 1981

these cases a scheme of second-order accuracy is not adequate, a fourth-order scheme gives much better results.[4] Therefore this paper is devoted to a comparison of two methods which yield fourth-order accuracy for the elliptic equation (1).

The first method is described by Brandt and Dinar.[5] They call this 'local truncation extrapolation' or 'τ-extrapolation'. It has the appearance of Richardson extrapolation, however, they state that it is more general since it does not require a global expansion of the solution error in terms of the mesh spacing h.[5] The following is a brief description of τ-extrapolation. One of the most critical steps in the multigrid algorithm is the projection of the residue on the grid M^{k+1} (mesh spacing h_{k+1}) to form the right side of the difference equation on the M^k grid ($h_k = 2h_{k+1}$). The equation on the M^k grid is

$$L^k u^k = f^k = L^k I_{k+1}^k u^{k+1} + I_{k+1}^k (f^{k+1} - L^{k+1} u^{k+1}) \tag{3}$$

Here L^k is the finite difference approximation for the elliptic operator (1) which is defined in the next section. The operator I_{k+1}^k is the projection from the fine to the coarse mesh. Unless otherwise noted, we used the simple projection which assigns the fine mesh value to the corresponding coarse mesh point (note that each coarse mesh point is also a fine mesh point). Brandt shows that the use of 'residual weighting' in this projection can improve the convergence rate somewhat. If we denote the solution of the differential equation by U, then the discretization error (i.e. truncation error) on mesh level k is defined by (here r^k denotes the right side on mesh M^k)

$$L^k U - r^k \equiv \tau^k$$

Brandt and Dinar[5] indicate that an estimate for this error is given by (4.3) τ_{k+1}^k where

$$\tau_{k+1}^k \equiv I_{k+1}^k (r^{k+1} - L^{k+1} u^{k+1}) - (r^k - L^k I_{k+1}^k u^{k+1})$$

Note that $I_{k+1}^k r^{k+1} = r^k$, since we do not use weighted residuals, and therefore

$$\tau_{k+1}^k = L^k I_{k+1}^k u^{k+1} - I_{k+1}^k L^{k+1} u^{k+1}$$

If we have an estimate of the discretization error, say τ_e^k, then a solution of higher order accuracy can be obtained by solving the following equation

$$L^k u^k - r^k = \tau_e^k$$

This is the basis for the method of deferred corrections[6] and also iterative improvement.[5] If $k = M$ is the index for the finest mesh, then the right side of the relaxation equation for the next finest mesh is

$$f^{M-1} = I_M^{M-1}(r^M) + L^{M-1} I_M^{M-1} u^M - I_M^{M-1} L^M u^M$$

If instead we use

$$I_M^{M-1}(r^M) + \tfrac{4}{3}(L^{M-1} I_M^{M-1} u^M - I_M^{M-1} L^M u^M) = r^{M-1} + \tfrac{4}{3}\tau_M^{M-1}$$

then on the mesh level $l = M-1$ the relaxation equation becomes

$$L^{M-1} u - r^{M-1} = \tau_e^{M-1}$$

and we have a correction which should yield fourth-order accuracy on the $M-1$ mesh level.

As an alternative we will also describe results obtained from iterative improvement as developed by Lindberg.[7] This requires first the solution for the second-order equation

$$L^M u = r^M$$

Then a fourth-order operator L_4^M is used to obtain a correction, but only on the finest mesh, from the following equation

$$L^M v^M + L_4^M u^M = 2r^M$$

Lindberg shows that the solution v^M, again obtained with the second-order operator L^M, has fourth-order accuracy. This method requires two solutions using the second-order operator.

The numerical experiments use a rectangular mesh (x_i, y_j) where $1 \le i \le m2^{M-1}$, $1 \le j \le n2^{M-1}$ with equal spacing in each direction. Usually $1 \le m, n \le 3$. Here M is the index of the finest mesh level. The finite difference operator L^M of second order is defined by

$$L^M u = (\sigma_{i+\frac{1}{2},j}(u_{i+1,j} - u_{ij}) - \sigma_{i-\frac{1}{2},j}(u_{ij} - u_{i-1,j}))/\Delta x^2$$
$$+ (\sigma_{i,j+\frac{1}{2}}(u_{i,j+1} - u_{ij}) - \sigma_{i,j-\frac{1}{2}}(u_{i,j} - u_{i,j-1}))/\Delta y^2$$

In all our experiments Dirichlet boundary conditions were used. In one test case we used a discontinuous function $\sigma(x, y)$. However, the discontinuity was not severe, and therefore the techniques developed by Dendy *et al.*[10] for the multigrid method with discontinuous $\sigma(x, y)$ were not needed.

THE MULTIGRID ALGORITHM

In this section we will describe the variants of the multigrid method which we compared. All but the second are taken from Dinar's thesis.[8] We used the Full Approximation Scheme (FAS), thus we solved for the full solution u^k on each grid rather than solving for a correction to the solution on the finer grids. The interpolation operator I_{k-1}^k was the linear operator except for the interpolation to the finest grid I_{M-1}^M which will be discussed below. The operator I_k^{k-1} simply assigned the fine mesh value to the course mesh point, that is residual weighting was not used.

In the first two schemes a cycling algorithm was used, that is the iteration started on the finest mesh level. In the remaining schemes the full multigrid scheme was used which solved the problem on mesh level $k = 2$, then solved on each finer level to $k = M$. Cubic (four-point) interpolation was used to obtain an initial guess on level k from the final approximation on level $k-1$. The details of this algorithm are given in several papers by Brandt and also in Dinar's thesis.[8] We proceed to a description of the eleven schemes, each of which corresponds to a column in Table I.

1. This is a fixed cycling algorithm starting on the finest mesh. On the downward pass two relaxations are done on each mesh level below the finest level except on level one where four are used. On the pass back up only one relaxation is performed. On the highest (or finest) level only one relaxation is done per cycle. For a case with four mesh levels the number of relaxations on each mesh level through one complete cycle is the following

$$\{2*3, 2*2, 4*1, 1*2, 1*3, 1*4\}$$

Here $2*3$ denotes two relaxations on the third level. In addition, at the beginning of the first cycle two relaxation sweeps are used at level 4. At the end of the cycle, after the relaxation on the finest level, a check of the residue is made. If this residue is less than a given input tolerance the iteration is terminated. The residue is computed using a weighted maximum norm. The weights are

$$W(x, y) = \text{Min}\,(x - x_a, x_b - x) * \text{Min}\,(y - y_a, y_b - y)$$

where x_a, x_b and y_a, y_b are the edge co-ordinates of the rectangular domain. This weight function is normalized to a maximum value of one on the rectangle. The relaxation sweeps

Table I. Error in multigrid solution. The four numbers are (top to bottom), relative error in L_2 norm, relative error in maximum norm, relative error in maximum norm on $M-1$ mesh level, and CPU time (sec) on DEC KL1091

Test case \ Scheme	1 Second-order limiting solution	2 Iterative improve limiting solution	3 Double cycle. 6-point interp.	4 Single cycle. 6-point interp.	5 Double cycle. 4-point interp.	6 Double cycle. Linear interp.	7 Single cycle. Linear interp.	8 Iterative improve. Double cycle.	9 Iterative improve. Single cycle.	10 As col. 8 except second cycling algorithm	11 As col. 9 except second cycling algorithm
$\sigma = e^{-x-y}$	9·63E–4	4·61E–6	3·35E–5	1·54E–4	4·41E–4	1·00E–4	1·68E–4	4·53E–6	1·64E–5	4·73E–6	1·66E–5
$u = \sin \pi(x+y)$	2·22E–3	9·69E–6	1·30E–4	5·94E–4	1·21E–2	1·98E–4	4·49E–4	1·04E–5	5·88E–5	8·96E–6	5·89E–5
$x_b = y_b = 1;\ h = 1/16$	2·22E–3	9·69E–6	4·38E–5	4·50E–4	4·38E–5	1·78E–4	4·13E–4	1·03E–5	5·88E–4	8·96E–6	5·89E–5
	0·62	1·28	0·41	0·24	0·35	0·38	0·21	0·87	0·53	1·00	0·56
$\sigma = 1$	6·86E–3	4·11E–5	1·28E–3	2·14E–3	6·07E–3	1·18E–3	1·50E–3	4·61E–5	1·23E–4	4·14E–5	1·29E–4
$u = \sin 3(x+y)$	1·38E–2	1·55E–4	5·86E–3	1·15E–2	2·30E–2	2·19E–3	3·15E–3	1·39E–4	3·74E–4	1·32E–4	3·84E–4
$x_b = 3,\ y_b = 2,\ h = 1/8$	1·38E–2	7·75E–5	7·08E–4	6·95E–3	7·08E–4	2·03E–3	3·10E–3	8·72E–5	3·74E–4	7·90E–5	3·84E–4
	0·78	1·71	0·61	0·38	0·52	0·58	0·32	1·24	0·75	1·42	0·78
$\sigma = 1·0$	3·32E–4	5·83E–6	1·26E–3	1·28E–3	9·48E–4	1·71E–4	2·53E–4	2·92E–5	2·25E–5	3·01E–5	2·22E–5
$u = e^{-3(x^2+y^2)}$	7·56E–4	1·11E–5	1·37E–2	1·38E–2	5·05E–3	1·25E–3	1·28E–3	1·26E–4	7·88E–5	1·28E–4	7·79E–5
$x_b = y_b = 2;\ h = 1/8$	7·56E–4	1·09E–5	2·93E–4	2·56E–3	2·93E–4	7·49E–4	1·06E–3	1·26E–4	6·56E–5	1·28E–4	6·42E–5
	0·56	1·11	0·40	0·22	0·35	0·37	0·22	0·80	0·48	0·92	0·51

first along rows from left to right (inner loop), then from the bottom to the top of the mesh. The τ-extrapolation is not used in this first scheme so that this scheme is accurate to second order. The results of running this scheme are shown in Table I for three test cases. For each case and scheme four numbers are given. The top number is the relative error in the L_2 norm (Euclidean norm) with a unity weighting (i.e. no weighting, $w(x, y) \equiv 1$). By error we mean the difference between the mesh values and the solutions of the differential equation. Therefore this error does not approach zero as the number of iterations increases, but will go to zero with the mesh spacing. The second number is the relative error in the maximum norm. The third number is the maximum norm of the error measured on every other grid point, that is measured on mesh level $M-1$. This third number is of significance for some of the later schemes. Thus these three errors are

$$\|u^M - U\|_2/\|U\|_2 \qquad \|u^M - U\|_\infty/\|U\|_\infty \qquad \|u^{M-1} - U\|_\infty/\|U\|_\infty$$

The fourth number is the computing time on the DEC KL1091 at the Colorado School of Mines. These runs were all made using single precision arithmetic.

2. The second scheme uses iterative improvement to obtain fourth-order accuracy. This requires two solutions of the second-order equation each of which is obtained using the same multigrid iteration as in the first scheme. For both the first and second scheme the residue tolerance is taken small enough so that it has no significant effect on the error. The fourth-order approximation of the differential equation required by the iterative improvement uses six-point approximations of the derivatives at mesh points adjacent to the boundary and five-point approximations elsewhere. The self-adjoint form is not used for the fourth-order operator, thus the x-term is a difference approximation of

$$\sigma u_{xx} + \sigma_x u_x$$

3. The third multigrid scheme is one described by Dinar.[8] Unlike the cycling algorithm used in the first two schemes this is an FMG algorithm which starts on the second mesh level ($k = 2$). It is a double cycle which uses τ-extrapolation. On this level a zero initial guess is used and the solution is obtained by a dynamic cycling algorithm similar to 'cycle C'.[1] To obtain the solution on the third level, the second-level solution is interpolated to the third level using cubic interpolation. This provides an initial guess on the third level. Then a fixed 'double cycle' iteration is used to obtain an approximate solution on the third level. The process continues in this manner to higher grid levels. Each cycle moves down through the grid levels using two relaxation sweeps per level (except four sweeps on the lowest level). Then the iteration moves back up to the level immediately below the finest level, down again, then back up to the finest level. On this last move up to the finest mesh only one relaxation sweep is used per level. Each of these cycles uses approximately 4·2 work units, that is 4·2 relaxation sweeps on the finest mesh not counting the work required to interpolate between grid levels. Note that there is no final relaxation on the finest mesh.

The number of relaxations at each level for a grid with four levels is illustrated by the following two cycles on the fourth level

$$2*4, 2*3, 2*2, 4*1, 2*2, 2*3, 2*2, 4*1, 1*2, 1*3$$

$$\underline{2*4}, 2*3, 2*2, 4*1, 2*2, 2*3, 2*2, 4*1, 1*2, 1*3$$

Also, τ-extrapolation is used for $k > 2$. The same cycle is repeated twice except that the τ-extrapolation correction is done after the underlined relaxation on the second cycle. This

yields a fourth-order solution on the level below the finest level. Also the final relaxation on the fine mesh is not performed. Instead a high order interpolation is used to transfer the solution from mesh level 3 to level 4 after the final relaxation on level 3. Note that this interpolation is done only on the second cycle, the interpolation from level 3 at the end of the first cycle to level 4 to start the second cycle is linear. Also note that this high order interpolation is only used to obtain the solution on the finest mesh on the last FMG step. If there are 5 mesh levels, then this interpolation is used only to obtain the final solution on the fifth level, it is not used when FMG obtains a solution on the fourth level, instead linear interpolation is used to pass from the third to fourth level. Then a four-point cubic interpolation is used to obtain the initial guess for the FMG solution on the fifth level. For this third scheme the interpolation uses the nearest six coarse mesh points to interpolate to a fine mesh point. Thus the interpolation has sixth-order accuracy (i.e. $O(h^6)$).

4. This scheme is the same as the third except only a single cycle is used, that is the relaxation sweeps through the mesh levels are defined as follows

$$\underline{2*4}, 2*3, 2*2, 4*1, 2*2, 2*3, 2*2, 4*1, 1*2, 1*3$$

The τ-extrapolation correction is made after the first pair of relaxation sweeps on the fine mesh.

5. This scheme is the same as the double cycle of scheme three except that a four-point, fourth-order accurate interpolation is used to obtain the final values on the fine mesh.

6. This scheme is the same as the double cycle of scheme three except that linear interpolation followed by a single relaxation is used to obtain the final values on the fine mesh. The relaxation sweeps through the mesh levels as follows

$$2*4, 2*3, 2*2, 4*1, 2*2, 2*3, 2*2, 4*1, 1*2, 1*3$$
$$\underline{2*4}, 2*3, 2*2, 4*1, 2*2, 2*3, 2*2, 4*1, 1*2, 1*3, 1*4$$

The τ-extrapolation is used.

7. This is the same as scheme six except that only a single cycle is used. The relaxation sweep is

$$\underline{2*4}, 2*3, 2*2, 4*1, 2*2, 2*3, 2*2, 4*1, 1*2, 1*3, 1*4$$

The τ-extrapolation is used.

8. This scheme uses the double cycle sweep without τ-extrapolation in a solution by iterative improvement. The FMG algorithm is used with the same fixed cycle as in scheme six (illustrated for $k=4$) for levels three and higher ('cycle C' is used on level two). This FMG algorithm must be used twice since iterative improvement requires two solutions of the second-order accurate scheme.

9. This scheme is the same as scheme eight except the single-cycle sweep of scheme seven is used for each solution required by iterative improvement.

10. This is similar to scheme eight except that a different fixed cycle suggested by Brandt was used. Note that iterative improvement without τ-extrapolation is used here. The same FMG algorithm is used starting with the second level $k=2$. At each higher level ($k>2$) a double cycle is used. As before, cubic interpolation from the next lower level is used to initialize the double cycle. The cycle is given below for $k=5$.

$$2*5, 2*4, 2*3, 2*2, 4*1, 3*2, 1*4, 2*2, 3*3, 2*2, 4*1,$$
$$2*2, 2*3, 3*4, 2*3, 2*2, 4*1, 2*2, 2*3, 2*4, 1*5$$

This iteration moves from the finest level down to level one, then moves back up in steps. In moving up, when a level is reached for the first time, the iteration then drops to level one before starting back up. Approximately 5·8 work units are required per cycle. The cycle can also be described by the following pseudo program.

```
LB = K
For KK = 2 upto K
   For L = LB downto 1
      perform relaxation(s) on level L;
      If L > 1 then
         Project to level L-1
      Endfor;
   For L = 2 upto KK
      Inject from level L-1 to L;
      Perform relaxation(s) on level L
      Endfor;
LB = KK
Endfor
```

11. This is the same as scheme ten except that a single cycle is used in place of the double cycle. Iterative improvement without τ-extrapolation is again used.

RESULTS FOR MODEL PROBLEMS

The results for these nine schemes are shown in Table I. The rows represent three test cases. The first is a non-separable problem with diffusion coefficient $\sigma = \exp(-x - y)$. The next two are taken from Dinar's thesis.[8] We have used essentially the same multigrid schemes, except that we have extended his results to use six-point interpolation and we have used iterative improvement. Dinar's result for the maximum norm on level M for test case 2, scheme 6 is 2·2E–3 which is the same as our value 2·19E–3. Dinar's double cycle is the following

$$2*4, 2*3, 2*2, 4*1, 1*2, 1*3,$$
$$\underline{2*4}, 2*3, 2*2, 4*1, 2*2, 2*3, 2*2, 4*1, 1*2, 1*3, 1*4$$

This is slightly different than ours, however the results do not differ in a significant way. Dinar gives the maximum norm error on the level two mesh using his double cycle with τ-extrapolation. His value for our case 2 is 7·0E–4 which compares with our value 7·1E–4. For $h = 1/16$ his value is 4·4E–5 and ours is 4·70E–5.

The second cycling algorithm did not have much effect on the accuracy of iterative improvement as is seen in a comparison of columns 8 and 9 with columns 10 and 11 in Table I. There was also little difference in the results obtained from the two cycling algorithms when τ-extrapolation was used.

The iterative improvement correction is obtained by solving the equation

$$L^M v^M = r^M + r^M - L_4^M u^M.$$

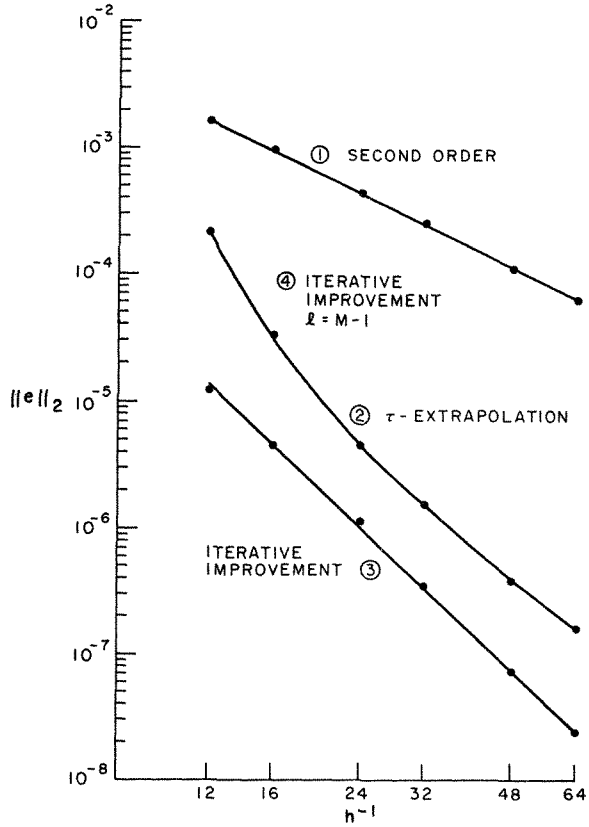

Figure 1. Error for case 1 of Table I. Curve ① second-order scheme, ② τ-extrapolation, ③ iterative improvement. L_2 error

We also tried using the correction term $L^M u^M - L_4^M u^M$ in place of $r^M - L_4^M u^M$. This altered correction term caused the error to increase by about a factor of ten, unless we used more relaxation sweeps in the multigrid iteration so that the residue $L^M u^M - r^M$ was reduced. Obviously, if the residue is taken to zero the two correction terms are equivalent. Therefore, the first correction term, $r^M - L_4^M u^M$, appears to be superior.

In Figure 1 the L_2 error is plotted for three schemes applied to the first case of Table I. Here $\sigma = \exp(-x - y)$ and $u = \sin \pi(x + y)$ is the solution (see equation (1)). The resolution varies from $h = 1/12$ to $h = 1/64$. This plot uses a log–log scaling. The second-order solution was obtained using the single-cycle multigrid iteration of scheme 7 except without τ-extrapolation. The τ-extrapolation was obtained with the double cycle scheme 3 and iterative improvement with the double cycle of scheme 8. If a line is determined by the second-order error at $h = 1/12$ and $h = 1/64$, then the slope is 1·99. The line determined by the iterative improvement error at the same points is 3·74. The τ-extrapolation curve is not quite so linear and yields a larger error than iterative improvement. The iterative improvement requires twice as much computing time as τ-extrapolation, but the improvement in accuracy more than justifies the increased cost, at least for this problem.

A comparison of the error measured using the maximum norm with that using the L_2 norm is given in Figure 2. The curves whose points are marked by ○ and ● were obtained using iterative improvement with the double cycle iteration. The results in this figure are also

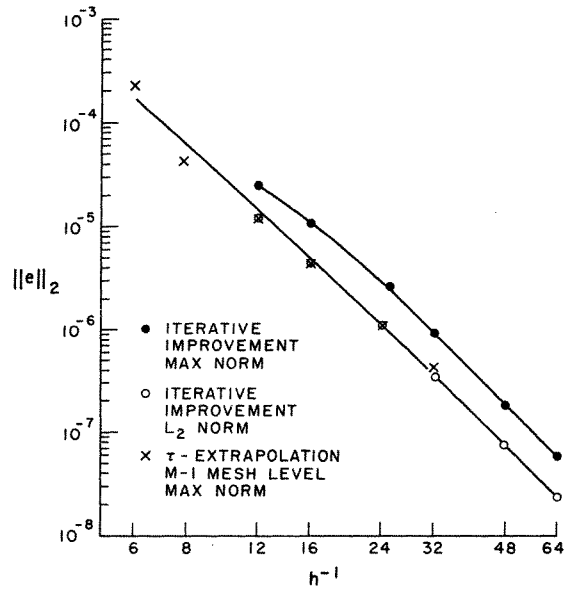

Figure 2. Error for case 1 of Table I. ● iterative improvement, maximum norm. × τ-extrapolation, maximum norm on $M-1$ mesh level. ○ iterative improvement, L_2 norm

for the first test case of Table I. The points marked × in Figure 2 give the maximum norm error for τ-extrapolation using the double cycle iteration. Note that τ-extrapolation yields a fourth-order solution only on the $M-1$ mesh level. To obtain a higher order solution we used six-point interpolation from the $M-1$ level to the M level. The intermediate solution which is obtained by relaxation on the M level during the double cycle iteration is used only to improve the $M-1$ level solution and define the correction in the τ-extrapolation. After the six-point interpolation there is no relaxation smoothing performed on the M level. The '×' point in Figure 2 at $1/h=32$ gives the error $\|u^{M-1}-U\|_\infty/\|U\|_\infty$ on the $M-1$ level where the fine mesh (M level) resolution is $1/h_M=64$. Thus the maximum norm error in τ-extrapolation on the $M-1$ level agrees well with L_2 norm error in iterative improvement provided τ-extrapolation at $1/h^\tau_{M-1}$ is compared with iterative improvement at $1/h^i_M$, that is $h^\tau_{M-1}=h^i_M$. Note that the τ-extrapolation error curve is linear (on a log–log scale) provided the error is measured on the $M-1$ level. On the M level, the τ-extrapolation curve, even in the L_2 norm, is not linear. This leads to the conclusion that the extrapolation from mesh level $M-1$ to level M is the weak link in the τ-extrapolation.

The next comparison concerns problems whose solution is not smooth, that is, one whose lower order derivatives are not continuous. We can compare iterative improvement to τ-extrapolation by application to the two-point boundary value problem

$$\frac{\mathrm{d}^2u}{\mathrm{d}x^2}=r(x)$$

The τ-extrapolation correction to the $M-1$ level mesh can be written

$$\tfrac{4}{3}(L^{M-1}I_M^{M-1}u^M-I_M^{M-1}L^Mu^M)=\tfrac{1}{3}(u^M_{i-2}-4u^M_{i-1}+6u^M_i-4u^M_{i+1}+u^M_{i+2})/h^2\simeq\frac{h^2_{M-1}}{12}u^{(4)}$$

The method of deferred corrections requires the addition of an approximation to this fourth derivative to the right side of the second-order difference equation.[6] Therefore, τ-extrapolation and deferred corrections seem to be the same in this case, except the former applies the correction only to the $M-1$ level mesh. This correction term is a second-order approximation to a fourth derivative. By comparison, iterative improvement uses a fourth-order approximation to a second derivative.[7]

This term has the following form

$$-(-u^M_{i-2}+16u^M_{i-1}-30u^M_i+16u^M_{i+1}-u^M_{i+2})/24h^2_M$$

We do not have any theory to show which of these corrections is less affected by discontinuous derivatives in the solution. However, our computational results seem to indicate that τ-extrapolation is superior to iterative improvement.

Our test case when non-smooth derivatives is the same as that used by Haidvogel and Zang.[9] The Poisson equation with zero boundary values is solved on a rectangle.

$$\begin{aligned} &u_{xx}+u_{yy}=r(x,y) && -1\le x\le 1\\ & && -1\le y\le 1 \qquad (4)\\ &u=0 \quad \text{for} \quad |x|=1 \quad \text{or} \quad |y|=1 \end{aligned}$$

The solution is the following piecewise polynomial

$$u(x,y)=V(x)V(y)$$

$$V(s)=\begin{cases} b(s+1) & s<t \\ (s-1)(s+a)/2 & s\ge t \end{cases}$$

Here t is a small positive input parameter and

$$a=(1-2t-t^2)/2$$
$$b=t+(a-1)/2$$

The forcing function is

$$r(x,y)=H(x)V(y)+V(x)H(y)$$

where

$$H(s)=\begin{cases} 0 & s<t \\ 1 & s\ge t \end{cases}$$

The second derivative of the solution is discontinuous along the lines $x=t$ and $y=t$. Haidvogel and Zang used the value $t=h/2$ where h is the mesh spacing. In one case they

Table II. Max norm relative error for equation (4)

Scheme / h^{-1}	Single cycle. Scheme 7. $t=h/2$	Single cycle. Scheme 7. $t=h/4$	τ-extrapolation Scheme 3. $t=h/4$	τ-extrapolation Scheme 3. Error on mesh level $M-1$. $t\times h/4$	Iterative improvement Scheme 8. $t=h/4$
8	2·80E–2	1·22E–1	—	—	—
16	6·17E–3	6·05E–2	3·26E–2	1·60E–3	5·83E–2
32	1·58E–3	2·97E–2	1·28E–2	7·86E–4	2·93E–2
64	3·92E–4	1·48E–2	5·62E–3	3·90E–4	1·47E–2

Table III. Solution time in seconds for equation (4)

	Single cycle Scheme 7 DEC–KL10	Double cycle Scheme 3 DEC–KL10	Direct solver DEC–KL10	Double cycle Scheme 3 CDC–6400 (FTN)
$h^{-1}=16$	0·25	0·42	0·15	—
$h^{-1}=32$	0·96	1·7	0·79	2·6
$h^{-1}=64$	3·94	6·8	3·85	—

used a fast direct method to solve the linear equations obtained from a second-order difference scheme. In this case they obtained second-order convergence. We obtained essentially the same results for this case. However, when we used $t=h/4$ so that the discontinuity does not fall in the middle of mesh cells, then the convergence was only first order in the mesh spacing. The results are shown in Table II. Note that τ-extrapolation did yield some improvement, however, it also appears to be only first-order accurate. Also note that τ-extrapolation was again much better on the $M-1$ mesh which indicates that the six-point interpolation from level $M-1$ to level M is the major problem here, as one would expect. Iterative improvement did very poorly.

In Table III a comparison of solution times is given. The direct solution was obtained from the SEPELI code written by John Adams at the National Center for Atmospheric Research. The SEPELI code can only be used for elliptic problems with separable coefficients. This code is a fast direct method based on factorization. This method is more general than most direct solvers, and is therefore somewhat slower. The multigrid method seems to compare quite well with this direct method.

The results in Tables I and II were obtained on a DEC KL10. The results in Figures 1 and 2 were obtained on a CDC 6400, since these results required higher precision than that provided by single precision 36 bit words.

APPLICATION TO A GEOTHERMAL MODEL

This model describes saturated single phase fluid flow in a porous medium which is heated from below. We restrict our model to a rectangular domain with no flow through the boundary, temperature specified at the top and bottom ($y=-1, y=0$), and non-heat conducting sides ($x=0, x=1$). The model is described more completely in an earlier paper.[4] The equations (2) are written for a stream function $\psi(x, y)$ and the deviation $\theta(x, y)$ in the temperature from the linear, steady state, zero flow conduction solution. The previous paper[4] assumed a viscosity independent of temperature so that the coefficient $\sigma(x, y)$ in the elliptic equation for the stream function was a constant (we assume the permeability is constant). In this paper we use a temperature dependent viscosity so that this coefficient is a function of θ. This makes direct methods or iterative methods which require an expensive 'pre conditioning' either inapplicable or expensive. Therefore, we have used the multigrid method, although an ICCG method might have done just as well.

The dependence of the coefficient σ on temperature is given by the relation

$$\sigma(y, \theta)=\frac{38}{38-(y-\theta)\Delta T}[1+a_1(y-\theta)\,\Delta T-a_2(y-\theta)^2\Delta T^2]$$

where $\Delta T = T(-1) - T(0)$ is the temperature difference between the top and bottom and the constants a_i have the values $a_1 = 3{\cdot}17\text{E}{-4}$ and $a_2 = 2{\cdot}56\text{E}{-6}$. The results are evaluated by inspection of the contour plots of $\theta(x, y)$ and $\psi(x, y)$ and also by comparison of the Nusselt number, denoted by Nu. This number is the ratio of the total heat transmission to that transmitted by conduction alone. Its definition is

$$\text{Nu}\,(y, t) = 1 - \int_0^1 [\theta_y(x, y, t) + \text{Ra}\,\theta(x, y, t)\psi_x(x, y, t)]\,dx$$

Our main objective here is not an evaluation of the accuracy and efficiency of the overall method used to solve the geothermal problem for $\theta(x, y)$ and $\psi(x, y)$. Instead we show that the results obtained from the multigrid method appear to converge with the mesh spacing for a Rayleigh number low enough to yield steady state solutions. The oscillatory solutions obtained at higher Rayleigh numbers are more difficult to check due to the expense of running the finer resolution cases (33×33 or 49×49 grids) at the small time steps required. However, we will show that the 'cycling' method (case 2 of Table I) gives the same result as the 'FAS' method (case 8 of Table I) for a single oscillatory case ($\text{Ra} = 86{\cdot}08$) on a 33×33 grid.

In Table IV results for a steady state case are shown. The Rayleigh number is $\text{Ra} = 64{\cdot}56$, which is six times the 'critical Rayleigh number' as computed by Zabib and Kassoy.[11] The integrals required for the table are approximated by Simpson's method. The computing time refers to the CPU time in seconds required to solve the elliptic equation for the stream function $\psi(x, y)$ on a CDC 6400. The fourth-order calculations should require slightly more than twice as long as the second order. The FAS algorithm based on the second cycling algorithm used for column 10 of Table I is also used here. Iterative improvement is used to obtain fourth-order accuracy for the stream function. A fourth-order ADI scheme based on five-point approximations for the derivatives is used for the temperature equation.[4]

If the convergence is sufficiently smooth we could use Richardson extrapolation to improve the accuracy of the results given in Table IV. If $f(h)$ is the quantity computed using a second-order scheme we may assume $f(h) = f(0) + ch^2$. Using the pairs (1/16, 1/32), (1/24, 1/32) and (1/32, 1/48) for (h_1, h_2) we can estimate c from the equation

$$C = (f(h_2) - f(h_1))/(h_2^2 - h_1^2).$$

Table IV. Steady state solutions of equations (2) for $\text{Ra} = 64{\cdot}56$, $\Delta T = 200°$

	Second-order scheme		
$N = h^{-1}$	$\int \text{Nu}\,(g)\,dy$	$\int\int \theta(x, y)\,dx\,dy$	CPU time for ψ on one time step
16	3·8655	0·083062	0·94
24	3·84383	0·0853539	2·1
32	3·83894	0·0860060	4·0
48	3·83755	0·0864684	9·0
	Fourth-order scheme		
16	3·8217	0·084775	1·9
24	3·84680	0·0857457	4·4
32	3·84400	0·0862337	8·4
48	3·84073	0·0865472	19·0

With $f(h)$ given by the integral of the temperature field in Table IV, the values for these pairs are $(-1{\cdot}05, -0{\cdot}85, -0{\cdot}85)$. For the integral of the Nusselt number the three estimated values of c are $(10{\cdot}0, 6{\cdot}4, 2{\cdot}6)$. Thus Richardson extrapolation for the integral of θ gives a correction $ch^2 = 3{\cdot}7E{-}4$ and an extrapolated value of 0·086830. The correction is probably accurate to 20 per cent which would give an error in the extrapolated value of 7·0E–5. However, it is impossible to extrapolate the integral of the Nusselt number with this data. It is also impossible to extrapolate the data from the fourth-order accurate calculations since the estimated values of the coefficient c do not agree in the first digit. If we take the extrapolated value of $0{\cdot}08683 \pm 0{\cdot}00007$ as the correct value, then the second-order scheme using $N = 48$ gives a more accurate result than the fourth-order scheme with $N = 32$ and the two schemes require about the same computer time. These results do not show any advantage in the use of the fourth-order scheme. In the constant viscosity case, our previous paper[4] did indicate some advantage in the use of a fourth-order scheme.

Calculations were also made at a higher Rayleigh number, namely $\mathrm{Ra} = 86{\cdot}08$ which is eight times the critical value. Here also $\Delta T = 200°$. In this case a steady state solution is not obtained, instead there is an oscillation in time which appears to consist mainly of a single harmonic, at least in the Nusselt number. The period of this oscillation measured in the Nusselt number is about 1·5 time units.

To test the effect of the multigrid-iterative improvement method used to solve for the stream function $\psi(x, y)$ we compared the results for the fixed iteration used to obtain column 10 in Table I with the dynamic scheme used to obtain column two. In the latter scheme the multigrid iteration is continued until the residue on the fine mesh is less than an input tolerance ε. These comparisons were made with temperature difference $\Delta T = 200°$, Rayleigh number $\mathrm{Ra} = 86{\cdot}08$, time step $\Delta t = 0{\cdot}05$, mesh spacing $N = h^{-1} = 32$. To obtain uniform initial values a run was made with $\Delta t = 0{\cdot}1$ until the oscillation had settled down to a fairly uniform harmonic. The temperature field $\theta(x, y)$ was saved on a file and used to restart the computation based on a fixed iteration and also those based on the dynamic iteration using various values of ε. The comparisons were run over 10 time units, $0 \leq t \leq 10{\cdot}0$. The period of the oscillation is about 1·5. The comparison can be based on the instantaneous value of the integral of the temperature perturbation $\theta(x, y)$ or the integral of the stream function $\psi(x, y)$ at the end of the integration, or perhaps better on the amplitude of the oscillation in the $\theta(x, y)$ or $\psi(x, y)$ fields or in the integral of the Nusselt number. Note that the maximum value of θ is around 0·6 and the maximum of the stream function over the grid is around 0·15. The results in Table V indicate that the fixed multigrid iteration is sufficiently accurate. There appears to be little difference between the results for $\varepsilon = 0{\cdot}001$ and the fixed iteration. The change in the results caused by an increase in the time step from 0·50 to 0·1 or in the grid resolution from $N = 16$ to $N = 32$ is much greater.

These oscillation amplitudes were obtained by saving, at each grid point, a 'running' minimum and maximum value of θ and ψ over a time interval of length two. Then the amplitude at each grid point is $\theta_{\max} - \theta_{\min}$. The maximum amplitude is then obtained over the entire grid.

By running experiments on an oscillatory case with various time steps Δt and various grid resolutions N we can gain some idea of the accuracy of the integration, which allows a rough comparison between the second- and fourth-order methods. These cases were also run at Rayleigh number $\mathrm{Ra} = 86{\cdot}08$ and temperature difference $\Delta T = 200°$. A run was made at $N = 16$ until a steady oscillation was obtained, then the temperature field was saved. This field was used to start the runs for $N = 16$ which used the different time steps. A similar method was used for the cases with $N = 32$. The period of the oscillation is about 1·5, and in

Table V. Dependence of the oscillation on the accuracy of the iteration. Here $N = h^{-1} = 32$, $\Delta T = 200°$, Ra = 86·08, $\Delta t = 0{\cdot}05$, $t = 10{\cdot}0$

Method	Oscillation amplitude in θ_{max}	Oscillation amplitude in $\int \mathrm{Nu}\,(y)\,dy$	$\int\int \theta\,dx\,dy$	$\int\int \psi\,dx\,dy$
Fixed multigrid-iterative improvement	0·2471	0·2406	0·07676	0·04559
ε = tolerance for residue				
$\varepsilon = 0{\cdot}001$	0·2474	0·2405	0·07669	0·045560
$\varepsilon = 0{\cdot}004$	0·2482	0·2389	0·07660	0·04562
$\varepsilon = 0{\cdot}02$	0·2577	0·2261	0·07677	0·04544

each case the amplitude was reasonably constant after the first cycle, that is for $2 \leq t \leq 10$. The maximum Nusselt number measured at the top ($z = 0$) varied in the range $4{\cdot}444 \leq \mathrm{Max}\,(\mathrm{Nu}\,(t, 0)) \leq 4{\cdot}460$ and the minimum $4{\cdot}292 \leq \mathrm{Min}\,(\mathrm{Nu}\,(t, 0) \leq 4{\cdot}302$. This variation was taken in the interval $2 \leq t \leq 10$.

The amplitude of the oscillation in the temperature perturbation $\theta(x, y)$ and the amplitude of the oscillation in the integral of the Nusselt number $\int \mathrm{Nu}\,dy$ were measured and are given in Table VI. These results again show little improvement in the fourth-order results over the second order. At $N = 16$ the results are rather poor, if we assume the $N = 32$ results are reasonably accurate. The good agreement between the second- and fourth-order results at $N = 32$ would lead us to this latter conclusion. Note that the maximum value of the temperature perturbation $\theta(x, y)$ over the grid is around 0·6 and the average (in time) of the integral of the Nusselt number is around 4·45. Thus, the relative oscillation amplitude in the Nusselt number is much smaller than that in the temperature perturbation $\theta(x, y)$.

Another check on the accuracy of the integration is obtained from a comparison of the Nusselt number oscillations measured at the top and bottom. If the temperature equation is

Table VI. Dependence of an oscillatory case (Ra = 86·08, $\Delta T = 200°$) on time step Δt and grid resolution $N = h^{-1}$

		Second order		Fourth order	
N	Δt	Oscillation amplitude in θ_{max}	Oscillation amplitude in $\int \mathrm{Nu}\,dy$	Oscillation amplitude in θ_{max}	Oscillation amplitude in $\int \mathrm{Nu}\,dy$
16	0·1	0·32	0·47	0·20	0·27
	0·05	0·31	0·44	0·17	0·22
	0·25	0·30	0·44	0·15	0·20
32	0·1	0·27	0·33	0·27	0·29
	0·05	0·26	0·28	0·25	0·24
	0·025	0·26	0·28	0·24	0·23

integrated over space and time the following relation is obtained

$$\int_0^1 \int_{-1}^0 \theta(t+P, x, y)\,dx\,dy - \int_0^1 \int_{-1}^1 \theta(t, x, y)\,dx\,dy$$
$$= \int_t^{t+P} \int_0^1 \theta_y(t, x, 0)\,dx\,dt - \int_t^{t+P} \int_0^1 \theta_y(y, x, -1)\,dx\,dt$$

If the time dependence is periodic with period P, then this relation implies that the average Nusselt number taken over the period should be the same at the top and bottom of the domain. For $N=32$, $\Delta T=200°$, $\Delta t=0{\cdot}025$, $R=86{\cdot}08$, the average Nusselt number at the top is

$$\frac{1}{P}\int_t^{t+P} \mathrm{Nu}\,(t, -1)\,dt = 4{\cdot}52$$

compared to 4·45 at the bottom. This large discrepancy is probably due to the one sided derivatives used to compute θ_y at the vertical boundaries. A graph of the Nusselt number at the bottom, Nu $(t, -1)$, at the top, Nu $(t, 0)$, and the integral $\int_{-1}^{0} \mathrm{Nu}\,(t, y)\,dy$ is given in Figure 3 for the interval $94 \le t \le 98$. This was obtained using $N=32$, $\Delta t=0{\cdot}025$, $\Delta T=200°$, Ra $=86{\cdot}08$. The average value of the $T(t, x, y)$ field is given in Figure 4 ($T(t, x, y)=-z+\theta(t, x, y)$). These average values are defined by

$$T_{\mathrm{av}}(t_0, x, y) = 0{\cdot}5*\left[\min_{t_0-P\le t\le t_0} T(t, x, y) + \max_{t_0-P\le t\le t_0} T(t, x, y)\right].$$

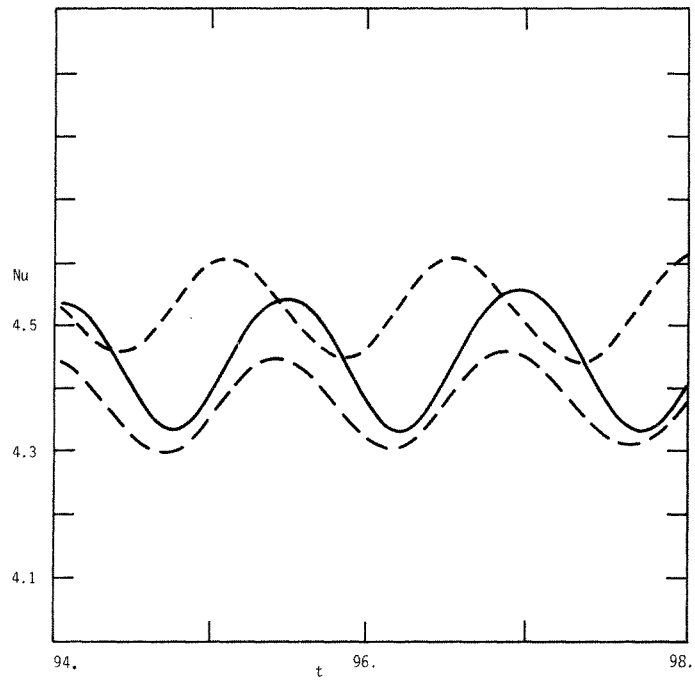

Figure 3. Nusselt numbers for $94 \le t \le 98$, $N=32$, $\Delta T=200°$, Ra $=$ $86{\cdot}08$, $\Delta t=0{\cdot}025$. —— integral $\int_{-1}^{0} \mathrm{Nu}\,(t, z)\,dt$. – – – value at the top Nu $(t, 0)$. — — value at the bottom Nu $(t, -1)$

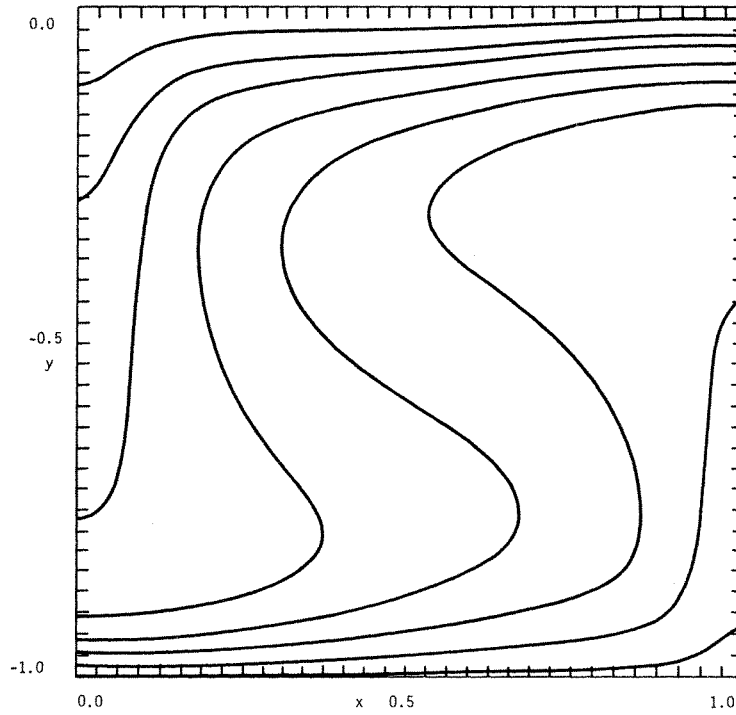

Figure 4. Average temperature $T_{av}(x, y)$ for $96 \leq t \leq 98$, $N = 32$, $\Delta T = 200°$, $Ra = 86{\cdot}08$, $\Delta t = 0{\cdot}025$. Contour interval = 0·12

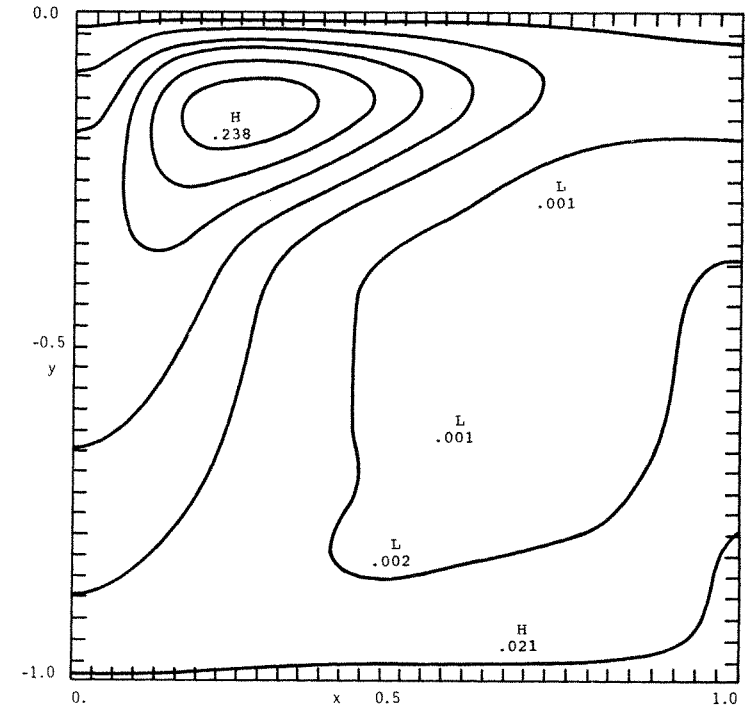

Figure 5. Temperature oscillation $T_{osc}(x, y)$, for same case as Figure 4. Contour interval = 0·03

The oscillation amplitude for $T(t, x, y)$ is given in Figure 5, and is defined by

$$T_{osc}(t_0, x, y) = \max_{t_0-P \le t \le t_0} T(t, x, y) - \min_{t_0-P \le t \le t_0} T(t, x, y).$$

For these results the time interval $P = 2$ was used.

ACKNOWLEDGEMENT

This research was supported by NSF grant ENG77–26893.

APPENDIX

Symbols

f	right side of difference equation
h	mesh spacing
H	function used to define test case
L^k	finite difference operator
M^k	k-level grid
N	number of mesh intervals
Nu	Nusselt number
P	period of oscillation
r	right side of elliptic equation
Ra	Rayleigh number
t	time
T	temperature
u	elliptic grid function
U	elliptic variable
v	elliptic grid function
V	function used to define test case
x	horizontal co-ordinate
y	vertical co-ordinate
σ	diffusion coefficient
θ	temperature perturbation
ψ	stream function for geothermal model
τ	'tau extrapolation' correction
ε	tolerance used for multigrid
Δt	time step
ΔT	temperature interval used for viscosity

REFERENCES

1. A. Brandt, 'Multilevel adaptive solutions to boundary value problems', *Math. Comp.*, **31,** 333–391 (1977).
2. P. Roach, *Computational Fluid Dynamics*, Hermosa Press, Albuquerque, N.M., 1972.
3. P. Cheng, 'Heat transfer in geothermal systems', in *Advance in Heat Transfer Vol. 14*, Academic Press, 1978.
4. J. Gary and D. Kassoy, 'Computation of steady and oscillatory convection in saturated porous media', submitted to *J. Comp. Phys.*
5. A. Brandt and N. Dinar, 'Multigrid solutions to elliptic flow problems', *ICASE Rpt. 79–15*, NASA Langley Research Center, Hampton, Virginia (1979), also in Madison meeting proceedings.

6. V. Pereyra, 'Highly accurate numerical solution of casilinear elliptic boundary value problems in n dimensions', *Math. Comp.*, **24,** 771–782 (1970).
7. B. Lindberg, 'Error estimation and iterative improvement for the numerical solution of operator equations', *Rpt. UIUCDCS–R–7–820,* Dept. Computer Science, Univ. Illinois, Urbana, Ill. (1976).
8. N. Dinar, 'Fast methods for the numerical solution of boundary value problems', *Ph.D. Thesis,* Weizmann Institute of Science, Rehovat, Israel, 1979.
9. D. Haidvogel and T. Zang, 'The efficient solution of Poisson's equation by expansion in Chebyshev polynomials', *J. Comp. Phys.* **30,** 167–180 (1979).
10. R. Alcouffe, A. Brandt, J. Dendy and J. Painter, 'The multigrid methods for the diffusion equation with strongly discontinuous coefficients', *LA–UR–80–1463,* Los Alamos Scientific Lab., N.M. (1980).
11. Z. Zabib and D. Kassoy, 'Three dimensional natural convection motion in a confined porous medium', *Phys. Fluids,* **21,** 1–3 (1978).