Parabolic multi-grid methods

Chapter · June 1984

Wolfgang Hackbusch
Max Planck Institute for Mathematics in the Sciences

Computing Methods in Applied Sciences and Engineering, VI
R. Glowinski and J.-L. Lions (Editors)
Elsevier Science Publishers B.V. (North-Holland)
©INRIA, 1984

PARABOLIC MULTI-GRID METHODS

Wolfgang Hackbusch

Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität
Olshausenstr 4o
D-23oo Kiel 1, RFA

A multi-grid iteration for solving parabolic partial differential equations is presented. It is characterized by the simultaneous computation of several time steps in one step to the computational process.

1. INTRODUCTION

1.1 MULTI-GRID METHODS FOR STATIONARY PROBLEMS

Multi-grid methods are very efficient solvers for elliptic equations. The iteration consists of a 'smoothing step' and a 'coarse-grid correction' involving a sequence of coarser grids (cf [1],[4],[5]). Let $\{h_l\}_{l\geq o}$ be a sequence of grid sizes (eg, $h_{l-1} = 2h_l$) and denote the linear discrete problem by

$$L_l u_l = f_l . \tag{1.1}$$

The multi-grid iteration al level l for solving Eq (1.1) is defined by the following recursive procedure performing one iteration step:

```
procedure MGM(l,u,f); integer l; array u,f;
if  l=o  then  u:=$L_o^{-1}$·f  else
begin integer j; array v,d;
        for j:=1 step 1 until $\nu$ do u:=$S_l$(u,f);
        d:=r·($L_l$·u-f);                                    (1.2)
        v:=o;
        for j:=1 step 1 until $\gamma$ do MGM(l-1,v,d);
        u:=u-p·v
end;
```

S_l denotes a 'smoothing iteration' (eg, the Gauß-Seidel iteration). The number ν of smoothing iterations is independent of l. p is a prolongation (interpolation) from the coarse grid of level l-1 into the fine one of level l. For elliptic problems of second order the usual choice is a piecewise linear interpolation. The restriction r from the fine into the coarse grid may be chosen as adjoint of p: $r = p^*$ (cf [4]).

The typical multi-grid convergence speed is uniform with respect to the discretization parameter (cf [4]).

Usually, the coarser grids are also used to provide good starting values. The resulting 'nested iteration' reads as follows:

```
$u_o := L_o^{-1} f_o$ ;
for k:=1 step 1 until l do
begin $u_k := \tilde{p} \cdot u_{k-1}$;                                  (1.3)
    for j:=1 step 1 until i do MGM(k, $u_k$, $f_k$)
end;
```

Here, the number i of iterations is often very small (eg, $i = 1$).

1.2 PARABOLIC EQUATIONS

A linear parabolic problem is given by

$$u_t + Lu = f \tag{1.4}$$

(L: elliptic operator) together with initial and boundary conditions. Assume that again different levels of discretizations are given characterized by a spatial grid size Δx_l. In addition there are time steps Δt_l. A simple discretization is the implicit Euler formula

$$P_l(t, u_l(t), u_l(t-\Delta t_l)) = f_l(t), \tag{1.5a}$$

where the discrete parabolic operator is defined by

$$P_l(u_l, u_l', t) := (u_l - u_l')/\Delta t_l + L_l(t)\, u_l\,. \tag{1.5b}$$

The advantage of the implicit scheme (1.5b) is the stability for all ratios $\Delta t_l / \Delta x_l^2$. One might prefer other implicit schemes of higher order of consistency (eg, the Crank-Nicholson scheme), but for the sake of simplicity we shall always consider (1.5a,b) in the sequel..

Also for simplicity we shall use fixed time steps Δt_l. Of course, a good parabolic solver has to use variable time steps, but we do not want to mix the discussion of time stepping strategies with the explanation of the multi-grid procedure for solving the implicit equations (1.5a,b).

1.3 EXISTING MULTI-GRID APPROACHES

Eq (1.5a,b) is a discrete elliptic problem for the unknown grid function $u_l(t)$. Hence, it can be solved by one or more iterations of the usual multi-grid methods for solving stationary problems (cf [3]).

A second approach is the 'frozen τ-technique' of Brandt [2]. It is restricted to the case, where u(t) approaches the stationary limit $u_\infty = \lim \{u(t) : t \to \infty\}$ and exploits the smoothness of the remainder $u(t) - u_\infty$. Numerical results are reported by Kroll [6].

The approach explained below will be able to converge to the discrete solution of (1.5) regardless of the smoothness or non-smoothness of the solution. But if $u_l(t)$ changes smoothly in time, it will be possible to save a part of the computational work.

2. PARABOLIC MULTI-GRID ITERATION

The conventional approach is to solve Eq (1.5) time step by time step; $u_l(t)$ is computed from $u_l(t-\Delta t_l)$, then $u_l(t+\Delta t_l)$ from $u_l(t)$ etc. The following process will be different. Assume that $u_l(t)$ is already computed or given as initial state. Simultaneously, we shall solve for $u_l(t+\Delta t_l)$, $u_l(t+2\Delta t_l)$, ..., $u_l(t+k\Delta t_l)$ in one step of the algorithm. The following step of the algorithm will yield $u_l(t+(k+1)\Delta t_l)$, ..., $u_l(t+2k\Delta t_l)$ etc. The number k may depend on l if Δt_l does so:

$$k = k_l = \delta t / \Delta t_l, \tag{2.1}$$

where δt is the time length by which the computation proceeds per step. In §2.1 we shall consider the case of $\Delta t_l=\Delta t_{l-1}=\ldots=\Delta t_o$, while the coarsening in time $(\Delta t_l<\Delta t_{l-1})$ is discussed in §2.2.

2.1 CASE OF EQUAL TIME STEPS

For simplicity consider only the two levels l and l-1 with time steps $\Delta t_l=\Delta t_{l-1}$, whereas the spatial grid size is coarsened, eg by $\Delta x_{l-1}=2\Delta x_l$. Fig 2.1a,b shows the situation for k=4 in the 1-D case.

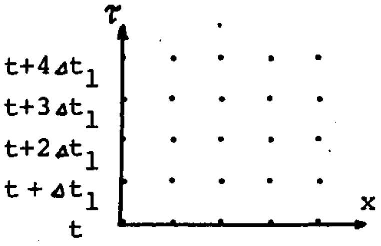

Fig 2.1a: Grid at level l

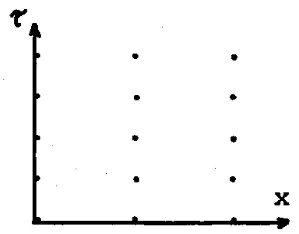

Fig 2.1b: Grid at level l-1

Let $u_l^{(\nu)}=S_{l,stat}(t,u_l^{(\nu-1)},f_l)$ be a stationary smoothing iteration as used in (1.2), where L_l is replaced by $L_l(t)+I/\Delta t_l$. Eg, $S_{l,stat}$ may be the Gauß-Seidel iteration. The corresponding iteration applied to Eq (1.5) is $S_l=S_l(t,u_l(t),u_l(t-\Delta t_l),f_l(t))$ defined by

$$S_l(t,u_l,u_l',f_l)=S_{l,stat}(t,u_l,f_l+u_l'/\Delta t_l). \tag{2.2}$$

$u_l^{(\nu)}=S_l(t,u_l^{(\nu-1)},u_l',f_l)$ is an iteration for $u_l=u_l(t)$, whereas $u_l'=u_l(t-\Delta t_l)$ is fixed.

We recall that the multi-grid algorithm is defined by the smoothing step and the coarse-grid correction. In the parabolic case the <u>smoothing step</u> consists of ν applications of S_l to $u_l(t+\Delta t_l)$, ..., $u_l(t+k\Delta t_l)$; eg by

```
for $\tau$:=t+$\Delta t_l$ step $\Delta t_l$ until t+k$\Delta t_l$ do
  for j:=1 step 1 until $\nu$ do                                (2.3)
    $u_l(\tau):=S_l(\tau,u_l(\tau),u_l(\tau-\Delta t_l),f_l(\tau))$
```

The <u>coarse-grid correction</u> of the parabolic multi-grid iteration reads as follows:

1) compute the defects

$$d_l(\tau) := P_l(\tau,u_l(\tau),u_l(\tau-\Delta t_l)) - f_l(\tau) \tag{2.4a}$$

for $\tau=t+j\Delta t_l$, $1\leq j\leq k$ (d_l is defined on the grid of Fig 2.1a);

2) restrict the defects to grid functions of level l-1:

$$d_{l-1}(\tau) := rd_l(\tau),\quad \tau=t+j\Delta t_l,\quad 1\leq j\leq k, \tag{2.4b}$$

where r is chosen as mentioned in §1.1 (d_{l-1} is defined on the grid of Fig 2.1b);

3) solve the following coarse-grid equation (parabolic initial-value problem):

$$v_{l-1}(t) = o, \tag{2.4c}$$
$$P_{l-1}(\tau,v_{l-1}(\tau),v_{l-1}(\tau-\Delta t_l)) = d_{l-1}(\tau),\quad \tau=t+j\Delta t_l,\quad 1\leq j\leq k;$$

4) prolongate v_{l-1} by

$$v_l(\tau) := p\cdot v_{l-1}(\tau), \quad \tau=t+j\Delta t_l, \quad 1\le j\le k \tag{2.4d}$$

with p from §1.1;

5) correct $u_l(\tau)$ by

$$u_l(\tau) := u_l(\tau) - v_l(\tau), \quad \tau=t+j\Delta t_l, \quad 1\le j\le k. \tag{2.4e}$$

The different time levels are coupled by (2.4c). The following note applies to the other parts of the coarse-grid correction:

NOTE 2.1 The steps (2.4a,b,d,e) of the coarse-grid correction can be performed by k parallel processors.

In the two-grid case the coarse-grid equation (2.4c) is solved exactly. If we replace the exact solution of (2.4c) by γ (cf (1.2)) iterations of the multi-grid process for the parabolic problem at level $l-1$ we obtain the multi-grid iteration for solving (1.5) at $\tau=t+j\Delta t_l$, $1\le j\le k$.

In the process (2.5a-e) we did not mention the boundary conditions explicitly. If $u_l(\tau)$ satisfies some Dirichlet conditions, $v_{l-1}(\tau)$ from (2.4c) has to fulfil homogeneous Dirichlet values. Then, the correction (2.4e) does not destroy the boundary conditions.

REMARK 2.2 It can be proved that the two-grid iteration (2.3), (2.4a-e) as well as the corresponding multi-grid iteration converge with the typical speed of multi-grid iterations. The contraction number is bounded by $\rho \ll 1$ independently of the step sizes Δt_l and Δx_l. This holds in particular for large ratios $\Delta t_l/\Delta x_l^2$, where eg the Gauß-Seidel iteration is very slow.

2.2 CASE OF COARSER TIME STEPS

Consider the case of Fig 2.2, where $\Delta t_{l-1}=4\Delta t_l$. A coarsening with

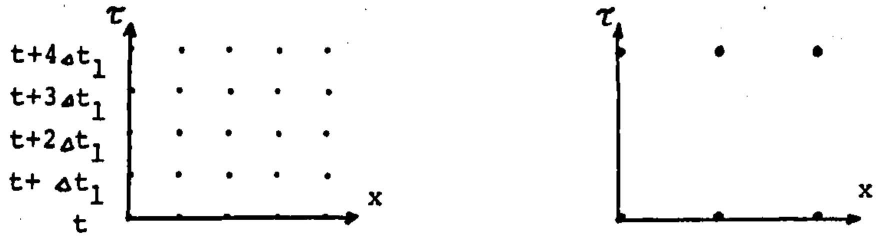

Fig 2.2a: Grid at level l Fig 2.2b: Grid at level l-1

respect to all (space and time) variables seemes to be natural for a multi-grid algorithm. The smoothing step is again (2.3). The coarse-grid correction (2.4) has rather to be changed. (2.4b) becomes

$$d_{l-1}(\tau) := r d_l(\tau), \quad \tau=t+j\Delta t_{l-1}, \quad 1\le j\le k_{l-1}, \tag{2.4b'}$$

where r is a restriction also in time. k_l and k_{l-1} are connected by $k_l\Delta t_l=k_{l-1}\Delta t_{l-1}=\delta t$ (cf (2.1)). In case of Fig 2.2 we have $k_l=4$ and $k_{l-1}=1$. Analogously, the coarse-grid equation has to be solved in steps of width Δt_{l-1}:

$$v_{l-1}(t) = o, \tag{2.4c'}$$
$$P_{l-1}(\tau, v_{l-1}(\tau), v_{l-1}(\tau-\Delta t_{l-1})) = d_{l-1}(\tau), \quad \tau=t+j\Delta t_{l-1}, \quad 1\le j\le k_{l-1}.$$

Step (2.4d) is formally the same as before, but p is a prolongation also in time direction. (2.4e) remains unchanged.

If $k_{l-1}=1$, a very simple restriction is given by

$$d_{l-1}(t+\delta t) := r_{spatial}\cdot d_l(t+\delta t),\quad \delta t=\Delta t_{l-1}=k_l\Delta t_l, \qquad (2.4b'')$$

with no weighting in time direction. In that case the defect (2.4a) has to be determined in $t+\delta t$, only.

REMARK 2.3 The two-grid iteration (2.3), (2.4a,b',c',d,e) does not converge with the usual multi-grid speed.

The reason of slow convergence (or divergence) are errors which are smooth in the spatial directions but nonsmooth in time. Assume, eg, in case of Fig 2.2 that the errors at level l are $e_l(t+j\Delta t_l)=o$ for $j=o$ and $j=2,3,4$, but $e_l(t+\Delta t_l)=v_l$ ($o\neq v_l$ smooth). Since the defects are smooth in space, the smoothing step does not improve the solution, significantly. On the other hand the coarse-grid correction does not provide a good correction because of $\Delta t_{l-1} > \Delta t_l$.

However, if the defects $d_l(\tau)$ are smooth in time, the coarse-grid correction (with or without preceding smoothing step) yields an efficient improvement.

REMARK 2.4 Assume that the exact discrete solution $u_l(\tau)$, $\tau=t+j\Delta t_l$, $o\leq j\leq k_l$, as well as the starting guess $u_l^o(\tau)$ is smooth in time. Then one step of the two-grid iteration (2.3), (2.4a,b',c',d,e) (with coarser time step $\Delta t_{l-1} > \Delta t_l$) yields a much better next iterate $u_l^{(1)}(\tau)$.

Proof. By assumption the error $u_l(\tau)-u_l^{(o)}(\tau)$ and thereby the defect is smooth.

After one iteration nonsmooth error components may dominate so that further iterations does not give the same success as the first one (cf Remark 2.3).

So far we only mentioned the two-grid iterations. There are different possibilities for the multi-grid iteration depending on the choice of $\Delta t_{l-2},\ldots,\Delta t_o$. One may proceed with $\Delta t_o=\ldots=\Delta t_{l-2}=\Delta t_{l-1}$ (this is the only choice if $\Delta t_{l-1}=\delta t$, ie $k_{l-1}=1$). Then, the iterations at lower levels are multi-grid iterations as explained in §2.1. The choice $\Delta t_{l-2}>\Delta t_{l-1}$ might give rise to slow convergence because of Remark 2.3 unless the defects $d_{l-1}(\tau)$ are smooth in time.

2.3 STARTING VALUES

The multi-grid iteration (either from §2.1 or 2.2) requires starting iterates $u_l^o(\tau)$, $\tau=t+j\Delta t_l$, $o\leq j\leq k_l$. A possible simple choice is

$$u_l^{(o)}(t+j\Delta t_l) = u_l(t),\quad o\leq j\leq k_l, \qquad (2.5)$$

where $u_l(t)$ is the initial value given either by the previous computation or by the initial value of the given parabolic problem. The choice (2.5) requires that the boundary values are constant in time. Otherwise, the boundary values have to be corrected in (2.5).

Another cheap approach corresponds to the technique of the nested iteration (1.3). By the simple injection to the coarser grid we can define initial values

$$v_{l-1}(t) := r_{injection}\cdot u_l(t) \qquad (2.6a)$$

at time t. Solve the discrete parabolic problem at level l-1 by the multi-grid process:

$$\text{Let } v_{l-1}(\tau),\ \tau=t+j\Delta t_{l-1},\ 1\leq j\leq k_{l-1}, \text{ be an approximate solution of } P_{l-1}(\tau,v_{l-1}(\tau),v_{l-1}(\tau-\Delta t_{l-1}))=f_{l-1}(\tau), \qquad (2.6b)$$

where possibly $\Delta t_{l-1} > \Delta t_l$ (cf Fig 2.1 or Fig 2.2, resp). Calculate

$$\left(v_l(t+\Delta t_l),\dots,v_l(t+\delta t)\right) := \tilde{p}\left(v_{l-1}(t+\Delta t_{l-1},\dots,v_{l-1}(t+\delta t)\right), \quad (2.6c)$$

where $\tilde{p}$ describes a suitable interpolation of the coarse-grid function v_{l-1} in the fine grid. If $\Delta t_l=\Delta t_{l-1}$, $\tilde{p}$ involves only a spatial interpolation. v_l is almost the starting guess at level l. However, $v_l(t)$ is not equal to $u_l(t)$ in general. Therefore, the final definition reads as

$$u_l^{(o)}(\tau) := v_l(\tau) + u_l(t) - v_l(t), \; \tau=t+j\Delta t_l, \; o\leq j\leq k_l. \quad (2.6d)$$

If v_l satisfies the Dirichlet boundary conditions, then $u_l^{(o)}$ does so, too. In more complicated cases one has to correct the boundary conditions suitably.

2.4 COMPLETE ALGORITHM

Combining the various parts explained above we obtain the following algorithms for the computation of the solution in $[t,t+\delta t]$.

first variant: (2.7)

a) compute the starting iterate $u_l^{(o)}$ by (2.6a-d),

b) apply i multi-grid iterations with $\Delta t_l=\Delta t_{l-1}$ (cf §2.1).

Due to Remark 2.2 the iteration converges fast. The computational work can be reduced by choosing $\Delta t_{l-1}>\Delta t_l$ in the first step of the iteration. According to Remark 2.4, the error of $u_l^{(o)}$ can greatly be reduced by one step with coarser time width Δt_{l-1}, provided that $u_l^{(o)}$ and u_l are smooth in time. The resulting algorithm is the

second variant: (2.8)

a) compute the starting iterate $u_l^{(o)}$ by (2.6a-d),

b) apply one multi-grid iteration with $\Delta t_{l-1}>\Delta t_l$ (cf §2.2),

c) apply i multi-grid iterations with $\Delta t_{l-1}=\Delta t_l$ (cf §2.1).

The number i of iterations should be as small as possible. The smallest choice i=o yields the

third variant: (2.9)

a) compute the starting iterate $u_l^{(o)}$ by (2.6a-d),

b) apply one multi-grid iteration with $\Delta t_{l-1}>\Delta t_l$ (cf §2.2).

In contrast to algorithms (2.6) and (2.7), the latter version is no iteration. It yields a result $u_l^{(1)}$, which is different from the exact solution of the discrete problem (1.5). Algorithm (2.9) defines a new discretization, which has almost the accuracy of the discretization (1.5) and is much more accurate than the algorithm (2.7) with i=o. Note that algorithm (2.7) with i=o yields the solution corresponding to the time step size Δt_{l-1} instead of Δt_l.

2.5 MODIFICATIONS

REMARK 2.5 The algorithms described above apply also to other discretizations, eg to the Crank-Nicholson scheme.

After some obvious modifications the algorithm can also be applied to multi-step discretizations.

In Note 2.1 we mentioned that the most parts of the coarse-grid correction can be performed in parallel. This does not hold for the smoothing iteration defined by (2.3). However, if the smoothing is performed by

```
for $\tau := t+\delta t$ step $-\Delta t_l$ until $t+\Delta t_l$ do
    for $j:=1$ step 1 until $\nu$ do
        $u_l(\tau) := S_l(\tau, u_l(\tau), u_l(\tau-\Delta t_l), f_l(\tau))$        (2.1o)
```

instead of (2.3), the computations at all k different time steps $t+j\Delta t_l$ are independent:

REMARK 2.6 In case of (2.1o) the smoothing step can be performed by k parallel processors.

Does the multi-grid iteration converge if (2.3) is replaced with (2.1o)? Assume that the errors are $e_l(t+j\Delta t_l)=\delta_{jm}v_l$ (δ_{jm}: Kronecker's symbol; $1\leq m\leq k_l$). Let v_l be an eigenfunction of $L_l(t)$. If v_l is of low frequency (ie, if the corresponding eigenvalue is small), this error will be reduced by the subsequent coarse-grid correction. Therefore, assume that v_l is of high frequency and that the sequential smoothing iteration (2.3) at time $t+m\Delta t_l$ reduces the error v_l to ρv_l with $\rho \ll 1$. The errors at $t+(m+1)\Delta t_l, \dots, t+\delta t$ generated by the further steps of (2.3) are even smaller. The result of the parallel smoothing (2.1o) is different. Again, the new error at time $t+m\Delta t_l$ is ρv_l. The starting defect at time $t+(m+1)\Delta t_l$ is $-v_l/\Delta t_l$, which is also reduced by the factor ρ. Hence, the new error at time $t+(m+1)\Delta t_l$ is αv_l with $\alpha=(1-\rho)/(1+\Delta t_l\lambda)$, where $\lambda v_l=L_l v_l$. Since ρ and λ are related by $1-\rho \approx \mathrm{const}\cdot\lambda\cdot\Delta x_l^2$, one concludes that $\alpha = O(\Delta x_l^2/\Delta t_l)$.

REMARK 2.7 If $\Delta t_l/\Delta x_l^2 \gg 1$ the parallel smoothing (2.1o) is successful. Note that $\Delta t_l/\Delta x_l^2 \gg 1$ is just the case, where implicit schemes are interesting because of their stability.

Although we want to avoid the discussion of possible time stepping strategies, we state that the present multi-grid algorithms yield the necessary dates.

REMARK 2.8 The starting step a) of (2.7), (2.8), or (2.9) yields $u(x,\tau;\Delta x_{l-1},\Delta t_{l-1})$. Eg, we can first compute $u(x,\tau;\Delta x_{l-1},\delta t)$ for $\Delta t_{l-1}=\delta t$, which serves as starting guess for the computation of $u(x,\tau;\Delta x_{l-1},\Delta t_l)$ with $\Delta t_l<\delta t$. Then, the remaining part of the algorithm yields $u(x,\tau;\Delta x_l,\Delta t_l)$. By these three values one can determine how the error depends on Δx and Δt. Hence, new adapted grid sizes can be guessed.

Stationary multi-grid methods can be applied not only to the linear problem (1.1) but also to nonlinear equations $L_l(u_l)=f_l$. The nonlinear multi-grid iterations is described, eg, in [2] and [4].

REMARK 2.9 Replacing the linear multi-grid scheme (1.2) by the nonlinear iteration, one can formulate nonlinear versions of the algorithms (2.7), (2.8), and (2.9).

3. NUMERICAL EXAMPLE: BUOYANCY-DRIVEN FLOW

We consider the time-dependent natural convection in a square cavity, which can be described by

$$-\Delta\psi_t + \mathrm{Pr}\,\Delta^2\psi + \psi_y\Delta\psi_x - \psi_x\Delta\psi_y - \mathrm{Ra}\cdot\mathrm{Pr}\cdot T_x = 0, \qquad (3.1a)$$

$$T_t - \Delta T + \psi_x T_y - \psi_y T_x = 0 \qquad (3.1b)$$

in $\Omega := \{(x,y): 0<x<1,\ 0<y<1\}$ and $0\le t\le 1$, where ψ: stream function, T: temperature, Ra=1000: Rayleigh number, Pr=0.71: Prandtl number. The boundary conditions are

$$\psi = \partial\psi/\partial n = 0 \quad \text{on } \Gamma = \text{boundary of } \Omega,\ 0\le t\le 1, \tag{3.2}$$

$$T=0 \text{ at } x=0, \quad \partial T/\partial n = 0 \text{ at } y=0 \text{ and } y=1. \tag{3.3a}$$

The right wall is heated according to

$$T(1,y,t) = t(2-t) \quad \text{for } 0\le y\le 1,\ t\ge 0. \tag{3.3b}$$

The initial value is

$$\psi(x,y,0) = T(x,y,0) = 0 \quad \text{for } (x,y)\in\Omega. \tag{3.4}$$

Although Eq (3.1) is not of the form of Eq (1.4), the equations (3.1 - 4) describe a well-behaving parabolic initial-boundary value problem.

The equations (3.1a,b) are discretized by central differences, Δ^2 is replaced by the square of the five-point scheme. The details of the multi-grid algorithm are as follows:

- ° smoothing iteration S_1: Gauß-Seidel iteration with chequer-board ordering of the grid points;
- ° p: piecewise linear interpolation for T (socalled nine-point prologation, cf [4]), piecewise cubic interpolation for ψ;
- ° r: for both T and ψ the restriction is chosen as nine-point restriction (adjoint of nine-point prolongation; cf [4]);
- ° $\nu=2$: number of smoothing iterations;
- ° $\gamma=1$: socalled V-cycle;
- ° $\Delta x_0=1/2$, $\Delta x_1=1/4$, $\Delta x_2=1/8$, $\Delta x_3=1/16$; $\Delta t_0=\ldots=\Delta t_3=1/4$; k=4;
- ° the nonlinearity is treated according to Remark 2.9.

The following table contains the values of ψ, T, and of the vorticity $\zeta=-\Delta\psi$ at the midpoint (1/2,1/2)'. The last column indicates the computational work measured by the unit 1 W, which is one Gauß-Seidel iteration or one evaluation of the defects at the finest grid with $\Delta x=1/16$. Rows 1 to 3 show the exact values for different grid sizes. One concludes that the discretization error with respect to Δt is smaller than the error arising from the spatial grid width. The further rows correspond to the finest grid parameters $\Delta t=1/4$, $\Delta x=1/16$ and different numbers of iterations:

- ° run A: multi-grid version of (2.7) with i=2;
- ° run B: same with i=3;
- ° run C: multi-grid version of (2.8) with two iterations in b) and two iterations in c);
- ° run D: same as C with three iterations in b) and c);
- ° run E: same as D, but the sequential smoothing (2.3) is replaced by the parallel smoothing (2.10).

One cbserves that the parallel smoothing (run E) yields almost the same results as the sequential smoothing (2.3). Even with a smaller number of iterations one obtaines good approximations to T.

parameters	ψ(1/2,1/2)	T(1/2,1/2)	ζ(1/2.1/2)	W
1 $\Delta t=1/4$, $\Delta x=1/16$; exact	1.203050	0.473829	31.86	
2 $\Delta t=1/8$, $\Delta x=1/16$; exact	1.208916	0.480620	32.04	
3 $\Delta t=1/4$, $\Delta x=1/8$; exact	1.317663	0.473752	32.55	
4 $\Delta t=1/4$, $\Delta x=1/16$; run A	1.3398	0.4761	-	39
5 $\Delta t=1/4$, $\Delta x=1/16$; run B	1.2226	0.4749	27.7	56
6 $\Delta t=1/4$, $\Delta x=1/16$; run C	1.1260	0.4739	22.1	44
7 $\Delta t=1/4$, $\Delta x=1/16$; run D	1.2090	0.4739	32.2	66
8 $\Delta t=1/4$, $\Delta x=1/16$; run E	1.2121	0.4738	32.2	66

REFERENCES

[1] Brandt, A., Multi-level adaptive solutions to boundary-value problems, Math. Comp. 31 (1977) 333-390.

[2] Brandt, A., Multi-level adaptive finite-element methods: variational problems, in: Frehse, J., Pallaschke, D., and Trottenberg, U. (eds.), Special topics of applied mathematics (North-Holland, Amsterdam, 1980).

[3] Hackbusch, W., A fast numerical method for elliptic boundary value problems with variable coefficients, in: Hirschel, E. H. and Geller, W. (eds.), Second GAMM conference on numerical methods in fluid mechanics (DFVLR, Köln-Porz, 1977).

[4] Hackbusch, W., Introduction to multi-grid methods for the numerical solution of boundary value problems, in: Essers, J.A. (ed.), Computational methods for turbulent, transonic, and viscous flows (Hemisphere, Washington, 1983).

[5] Hackbusch, W., and Trottenberg, U. (eds.), Multigrid methods (Springer, Berlin, 1982).

[6] Kroll, N., Direkte Anwendung von Mehrgittertechniken auf parabolische Anfangsrandwertaufgaben, Diplomarbeit, Bonn (1981).

[7] van der Houwen, P.J., and de Vries, H.B., Preconditioning and coarse grid corrections in the solution of the initial value problem for nonlinear partial differential equations, SIAM J. Sci. Stat. Comput. 3 (1982) 473-485.