A SPACE-TIME MULTIGRID METHOD FOR PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS*

G. HORTON† AND S. VANDEWALLE‡

Abstract. We consider the solution of parabolic partial differential equations (PDEs). In standard time-stepping techniques multigrid can be used as an iterative solver for the elliptic equations arising at each discrete time step. By contrast, the method presented in this paper treats the whole of the space-time problem simultaneously. Thus the multigrid operations of smoothing and coarse-grid correction are defined on all of the space-time variables of a given grid level. The method is characterized by a coarsening strategy with prolongation and restriction operators which depend at each grid level on the degree of anisotropy of the discretization stencil. Numerical results for the one- and two-dimensional heat equations are presented and are shown to agree closely with predictions from Fourier mode analysis.

Key words. parabolic partial differential equations, massively parallel computation, multigrid, semicoarsening

AMS subject classifications. 65M06, 65M55, 65Y05

1. Introduction. We consider the problem of numerically computing an approximation to $u(x, t)$, the solution of the d-dimensional parabolic partial differential equation (PDE)

$$(1) \qquad u_t - \Delta u = f(x, t), \qquad x \in \Omega = (0, 1)^d, \quad 0 < t \le T,$$

subject to the usual initial and boundary conditions

$$u(x, 0) = g(x), \quad x \in \Omega,$$
$$u(x, t) = h(x, t), \quad x \in \partial\Omega, \quad 0 < t \le T.$$

The standard solution procedure is to fully discretize equation (1), obtaining a discrete elliptic problem at each time step when an implicit scheme is used for the approximation of the time derivative. These elliptic problems are then solved sequentially using an iterative method such as successive overrelaxation, conjugate gradients, or multigrid. The potential for parallelism in this type of computation is limited to the parallelism of the elliptic solver, since the time dimension is treated strictly sequentially. Ironically, it is often the time dimension which contains the largest amount of computational work: the number of discrete time steps can be many times larger than the sidelength of the spatial grid. Thus it seems natural to ask whether this situation can be remedied and a parallelization strategy for the time-dependent part of the problem be found.

In [7] Hackbusch proposed a scheme in which the elliptic multigrid method can be executed simultaneously on a set of successive time steps. This *parabolic multigrid method* was investigated in a series of experiments by Bastian, Burmeister, and Horton [1], [6], [9]. Unfortunately, the method only performs satisfactorily under certain conditions; see [14]. In [18] Womble considered the *parallel time-stepping method*, a scheme whereby standard iterations are performed simultaneously on equations at successive time steps. Here the possibility of

combining parallelism in space and time was also investigated. The success of the method is due to the possibility of overlapping iterations of the iterative elliptic solver on different time levels. However, with rapid multigrid solvers requiring just a very small number of iterations, the potential for time parallelism is rather restricted. Vandewalle and Van de Velde have recently shown in [16] that the *multigrid waveform relaxation method* [12], for which the possibility of space parallelism has already been demonstrated by Vandewalle and Piessens [15], also permits parallel execution with respect to the time axis. There it was shown that the addition of time parallelism can considerably improve the utilisation of message passing multicomputers. A massively parallel parabolic PDE solver based on multigrid waveform relaxation was presented by Horton, Vandewalle, and Worley in [10]. The algorithm uses a line relaxation smoother, with lines extending in the time direction. This smoother was shown to lead to a rapidly convergent multigrid iteration, yet it limited the obtainable parallel complexity by introducing a logarithmic factor — the logarithm of the number of time levels — into the parallel complexity formula.

In the current paper a multigrid algorithm that employs a perfectly parallel pointwise smoother instead of a line-wise smoother is developed. In the following section we introduce some notation and show how a standard multigrid approach with point-wise smoothing and coarsening in space and time is inappropriate for problem (1). In §3 we present the space-time multigrid method in terms of its individual components, considering several variants. Section 4 gives results obtained by Fourier mode analysis of various multigrid schemes applied to the model problem. These are compared with results obtained by numerical experiments. In §5 numerical results are presented for different first- and second-order time-discretization schemes. The parallel complexity of the method is analysed in §6. In the final section some conclusions are drawn.

2. Motivation. We consider the one-dimensional heat equation, discretized in space using central differences on a regular mesh with mesh spacing Δx, and discretized in time with the backward Euler method on a set of time levels with constant time increment Δt. This leads to a large linear system of equations in the unknowns $u_{i,j}$ with $i = 1, \ldots, 1/\Delta x - 1$ and $j = 1, \ldots, T/\Delta t$ that approximate the PDE solution values at the grid points (x_i, t_j) with $x_i = i \cdot \Delta x$ and $t_j = j \cdot \Delta t$. The equations are of the form

$$(2) \qquad -\frac{1}{(\Delta x)^2} u_{i-1,j} + \left(\frac{2}{(\Delta x)^2} + \frac{1}{\Delta t} \right) u_{i,j} - \frac{1}{(\Delta x)^2} u_{i+1,j} - \frac{1}{\Delta t} u_{i,j-1} = f(x_i, t_j)$$

or, with the parameter λ defined by $\Delta t / \Delta x^2$,

$$(3) \qquad -\lambda u_{i-1,j} + (2\lambda + 1) u_{i,j} - \lambda u_{i+1,j} - u_{i,j-1} = \Delta t \, f(x_i, t_j) \,.$$

The parameter λ can be considered as a measure of the degree of anisotropy in the discrete operator. Note that λ will usually vary from grid level to grid level within a multigrid scheme. We shall therefore denote the value of λ on grid level l, with mesh spacings Δx_l and Δt_l, by λ_l.

Figure 1 shows the convergence rate of a "standard" multigrid method applied to this model problem, with the time axis considered as just another spatial axis. This multigrid method on the *two*-dimensional space-time grid is characterised by the following components: red-black smoothing with one presmoothing and one postsmoothing step, standard coarsening (i.e., $\Delta x_{l-1} = 2 \cdot \Delta x_l$ and $\Delta t_{l-1} = 2 \cdot \Delta t_l$), full weighting restriction, and bilinear prolongation. The coarse-grid operator is the natural one similar to (3). Multigrid V-cycles are used. Shown is the averaged convergence factor $\bar{\rho}$, which is computed from the $\| \cdot \|_2$ norm of the residual vector. A mesh with a total of six grid levels is used. The convergence is measured over values of λ_6 from 2^{-8} to 2^8 on a finest grid of size 65×65 points.
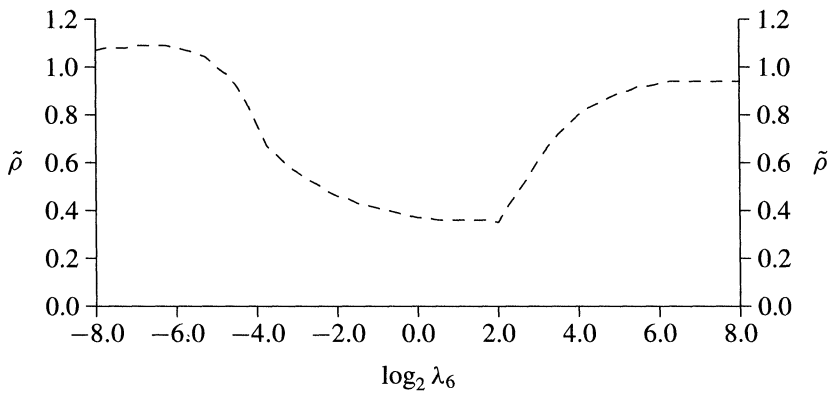
FIG. 1. *Convergence of standard multigrid applied to the space-time problem.*

It can be clearly seen that the method is not robust with respect to λ. For both $\lambda_6 < 2^{-4}$ and $\lambda_6 > 2^2$ we have either divergence or extremely slow convergence. Even in the range of values where the method performs best, the rate of convergence is not particularly fast. Thus we conclude that a naive approach to applying multigrid to the space-time problem does not succeed.

The unsatisfactory behaviour of the naive multigrid approach can be understood intuitively from analysing governing equation (3). For large and small values of λ the fully discrete PDE is a strongly anisotropic problem. Pointwise smoothing combined with standard coarsening is a notoriously slow procedure for such problems. Certain high frequency error components are not adequately smoothed out, since pointwise relaxation smoothes only in the direction of strongest coupling. The standard approach is then to resort to a different smoother, for example a linewise or incomplete factorization method, or to employ a suitable *semicoarsening* technique, with coarsening only in the dimensions where the pointwise smoother is successful. As the current paper is mainly motivated by parallel computing issues, we will follow here the second approach, which, most importantly, enables the use of perfectly parallel pointwise smoothers.

3. The space-time multigrid method. In this section we describe the various components of the new space-time multigrid method. For ease of presentation we restrict ourselves to the one-dimensional heat equation, i.e., the two-dimensional space-time problem. The operators for the three-dimensional space-time problem are defined analogously and are given in the Appendix. The method chooses a coarsening strategy, together with appropriate prolongation and restriction operators, which is based on the degree of anisotropy λ_l on the current grid level. The overall multigrid cycle will therefore use different strategies on different levels. Motivation for the choice is given by theoretical results described in §4.

3.1. Discretization. We consider central differences for the discretization of the space derivatives in equation (1) and both first- and second-order methods for the time derivative. The first-order method is the backward Euler scheme (3), which is equivalent to the first-order backward differentiation formula (BDF1). It leads to a four-point molecule, which, in stencil notation, is given by

$$(4) \qquad \begin{bmatrix} 0 & 0 & 0 \\ -\lambda_l & 2\lambda_l + 1 & -\lambda_l \\ 0 & -1 & 0 \end{bmatrix}.$$

The stencil notation of constant coefficient operators is explained in [13]. Note that the rows of the stencil correspond to the space dimension, and the columns to the time-dimension.

Two well-known second-order methods for the approximation of the time derivative will be analysed: the two-level Crank–Nicolson or trapezoidal method, which is given by the stencil

$$
(5) \qquad
\begin{bmatrix}
0 & 0 & 0 \\
-\lambda_l/2 & \lambda_l + 1 & -\lambda_l/2 \\
-\lambda_l/2 & \lambda_l - 1 & -\lambda_l/2
\end{bmatrix},
$$

and the three-level second-order backward differentiation method (BDF2), characterised by the stencil

$$
(6) \qquad
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & -\lambda_l & 2\lambda_l + 3/2 & -\lambda_l & 0 \\
0 & 0 & -2 & 0 & 0 \\
0 & 0 & 1/2 & 0 & 0
\end{bmatrix}.
$$

Note that the latter discretization cannot be used on the first time level of unknowns, since it requires values of two preceding time steps. There we will use the Crank–Nicolson formula instead, a purely technical choice which does not seem to impair our numerical results.

The same discretization technique is used to define the discrete operators on all grid levels in the multigrid hierarchy.

3.2. Smoothing. The smoother is a coloured pointwise Gauss–Seidel relaxation, whereby the definition of points of a certain colour is with respect to all dimensions of the problem. This smoother has the advantage of being very efficiently parallelizable. For this particular problem, other smoothers such as line-relaxation and block-incomplete LU factorization (ILU) are known to be efficient, but these do not have the above-mentioned property.

A two-colour scheme is used for the BDF1 method. A four-colour scheme and a three-colour scheme would be adequate for the Crank–Nicolson discretization and for the BDF2 method, respectively. Yet, for simplicity of implementation and analysis we choose to use a two-colour scheme, whereby updating of grid points is from high t-values to low t-values. In other words, when a point of a certain colour depends explicitly on other points of the same colour, the old values at these points are used for smoothing. This scheme has mixed Gauss–Seidel/Jacobi characteristics and can be easily implemented in parallel.

3.3. Coarsening. We choose an adaptive coarsening strategy based on the current value of λ_l. For each method we select a parameter λ_{crit} which is used as a switch for choosing the coarsening strategy at any particular grid level. We adopt the following strategy: "If $\lambda_l < \lambda_{\text{crit}}$ then *use semicoarsening in the time dimension,* else *use semicoarsening in the space dimension(s).*" The value of λ_{crit} is motivated by the results obtained by the two-grid Fourier mode analysis presented in §4. It is the value of λ_l at which the convergence rates of semicoarsening in the space and the time dimension are equal. When x-coarsening is performed we have $\lambda_{l-1} = 0.25\,\lambda_l$, whereas t-coarsening gives $\lambda_{l-1} = 2\,\lambda_l$. For comparison purposes, we shall also analyse the use of standard coarsening in space and time, in which case $\lambda_{l-1} = 0.5\,\lambda_l$.

Note that successive semicoarsening steps will ultimately lead to a coarse grid which has one or more sides of length one. In this case we continue to coarsen in the remaining directions until a grid consisting of only one variable is reached.

3.4. Prolongation. The prolongation is coupled to the choice of coarsening at each grid level. The stencils for the time-coarsening, space-coarsening, and standard-coarsening cases are given, respectively, by

$$
(7) \qquad \left]\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}\right[\ , \quad \frac{1}{2}\left]\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{array}\right[\ , \quad \frac{1}{2}\left]\begin{array}{ccc} 1 & 2 & 1 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{array}\right[\ .
$$

Note that the prolongation operator for the space-coarsening case is identical to the prolongation operator used in the multigrid waveform relaxation method [12], [15], and the one used in the parabolic multigrid method [7]. It is the standard operator used for the elliptic equation. In the case of time semicoarsening, prolongation is asymmetric, transferring no information backwards in time.

3.5. Restriction. The choice of the restriction operator is likewise coupled to the coarsening strategy. It is chosen, as usual, as the adjoint of the prolongation operator. The stencils for the time-coarsening, space-coarsening, and standard-coarsening cases are given, respectively, by

$$
(8) \qquad \frac{1}{2}\left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{array}\right] \ , \quad \frac{1}{4}\left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{array}\right] \ , \quad \frac{1}{8}\left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{array}\right] \ .
$$

The above observations on the prolongation operators apply equally to the restriction.

3.6. Computational complexity and storage requirements. The computational work of one multigrid cycle of the space-time multigrid method can be estimated as follows. The complexity of a multigrid cycle that uses both space and time coarsening is bounded by the complexity of the corresponding cycles that use only space coarsening or only time coarsening. The complexities of these two extremal cycles can be estimated easily. In both cases we assume a coarsening by a factor of 2 in all relevant directions. In the first case the ratio of the number of grid points from one grid to the next coarser grid is given by 2^d, where d is the spatial dimension of the problem. In the second case the ratio is 2. We denote by $\tilde{W}$ the ratio of the work in one cycle on a (large) set of grids and the work on the finest grid. The latter is, of course, a linear function of the total number of grid points. Approximate values of $\tilde{W}$ can be derived from Tables 8.3.1 and 8.3.2 in [17] and are presented in Table 1. These values are obtained when the work on the coarsest grid (i.e., the grid obtained when further space or time coarsening is no longer possible) is neglected. Note that a dash ("–") indicates that the complexity of the particular cycle is not a linear function of the total number of grid points. In the same table, we have also presented the complexity of a nested iteration of a full multigrid (FMG) algorithm that applies one multigrid V-cycle per grid level. The complexity of this algorithm is linear in the number of grid points and similar to that of an F-cycle.

The storage requirements of the algorithm can be estimated in a similar way by considering the storage requirements of the two extremal cases. With space coarsening the total number of storage locations is about $2^d/(2^d - 1)$ times the number of storage locations used to store the fine-grid variables. With time coarsening, which is the worst case situation, the ratio is about 2.

4. Analysis of the two-grid method.

4.1. Two-grid iteration operator. We consider a two-grid method for solving the one-dimensional model problem (1), using a fine grid Ω_h, defined by

$$
(9) \qquad \Omega_h = \{x \in \mathbb{R}^2 : x = jh \,, \ j = (j_1, j_2) \,, \ h = (h_1, h_2) \,,
$$
$$
j_1 = 0, 1, \ldots, n_1 \,, \ j_2 = 0, 1, \ldots, n_2 \,, \ h_1 = 1/n_1 \,, \ h_2 = T/n_2 \} \,,
$$

TABLE 1
Computational complexity ($\tilde{W}$-values) of a space-time multigrid V-cycle, F-cycle, and W-cycle, and of the full multigrid V-cycle algorithm, as a function of the spatial dimension d and the semicoarsening strategy.

	x-coarsening			t-coarsening		
d	1	2	3	1	2	3
V	2	4/3	8/7	2	2	2
F	4	16/9	64/49	4	4	4
W	–	2	4/3	–	–	–
FMG	4	16/9	64/49	4	4	4

where n_1 and n_2 are assumed to be even, and a coarse grid Ω_H, given by

$$(10) \qquad \Omega_H = \{x \in \mathbb{R}^2 : x = JH, \ J = (J_1, J_2), \ H = (H_1, H_2),$$

$$J_1 = 0, 1, \ldots, N_1, \ J_2 = 0, 1, \ldots, N_2, \ H_1 = 1/N_1, \ H_2 = T/N_2\}.$$

The coarse grid is derived from Ω_h by either standard coarsening ($H = (2h_1, 2h_2)$), x-coarsening ($H = (2h_1, h_2)$), or t-coarsening ($H = (h_1, 2h_2)$). Let $u_{k,l}$ denote an approximation to the PDE solution $u(kh_1, lh_2)$. The difference between the exact discrete solution $\bar{u}_{k,l}$ and the approximation is the error $e_{k,l} = u_{k,l} - \bar{u}_{k,l}$. By a two-grid cycle an error vector e^{old} is transformed into a new error vector e^{new}, with $e^{\text{new}} = M_h^H e^{\text{old}}$, where M_h^H is the two-grid iteration matrix. This matrix is given by

$$(11) \qquad M_h^H = S_h^{\nu_2} \left(I_h - I_H^h L_H^{-1} I_h^H F_h L_h\right) S_h^{\nu_1},$$

where S_h is the smoothing operator on Ω_h; ν_1 and ν_2 are the numbers of pre- and post-smoothing iterations; I_h, I_H^h, I_h^H, are the identity, prolongation, and restriction operators. L_H and L_h are discretized differential operators on the coarse and on the fine grid, respectively. They have been normalized by multiplication with h_2 or H_2, so that they are a function of λ_h ($=h_2/(h_1)^2$) or λ_H ($=H_2/(H_1)^2$) only, and not of the particular values of the discretization parameters h and H. F_h is a constant, introduced to correct for the different normalizations of the fine- and coarse-grid equations. It has the value 1 in the case of x-coarsening, and the value 2 otherwise.

4.2. Fourier mode analysis: Definitions and notation. The analysis of the two-grid algorithm applied to certain model problems, and, in particular, the analysis of the corresponding iteration matrix M_h^H, is often performed in the frequency domain. A comprehensive *model problem* analysis for a fairly large class of problems can be found in [13]. It is based the use of sinusoidal eigenmodes and the knowledge that M_h^H leaves certain linear spaces of sinusoidal modes invariant. Similar analyses can be found, for example, in [5], [8], and [11]. This type of analysis, however, is not applicable in the present case since the sine functions are not eigenfunctions of our discrete operators. In such cases one usually resorts to an analysis based on *exponential Fourier modes*. This kind of Fourier analysis was introduced by Brandt in his seminal paper [2] and made into a rigorous analysis in [4]. A large number of Fourier results are also presented in [13, Chaps. 9–10]. An exponential Fourier mode analysis is also pursued in Wesseling's book [17], where the smoothing properties of many common smoothing algorithms are investigated. In the remainder of this section we shall follow the approach of the last reference. This analysis can be regarded as an analysis for special model problems, namely those with *periodic boundary conditions* on (finite) rectangular domains.

The exponential Fourier mode $\psi_h(\theta)$ with *frequency* θ on Ω_h is given by

$$(12) \qquad \psi_h(\theta)_j = \exp(i \, j \cdot \theta),$$

where "·" denotes the usual $\mathbb{R}^2$ inner product; i is the imaginary unit ($\sqrt{-1}$); and

$$(13) \qquad j = (j_1, j_2), \quad j_\alpha = 0, 1, \ldots, n_\alpha - 1 \quad (\alpha = 1, 2),$$

$$(14) \qquad \theta \in \Theta_h = \{(\theta_1, \theta_2) : \theta_\alpha = 2\pi k_\alpha / n_\alpha, \ k_\alpha = -n_\alpha/2 + 1, \ -n_\alpha/2 + 2, \ldots, n_\alpha/2\}.$$

Note that the fine-grid Fourier mode $\psi_h(\theta)$ when injected into the coarse grid, aliases with coarse-grid mode $\psi_H(\bar{\theta})$, with $\bar{\theta}$ equal to $(2\theta_1, 2\theta_2)$ in the case of standard coarsening, with $\bar{\theta}$ equal to $(2\theta_1, \theta_2)$ in the case of x-coarsening, and with $\bar{\theta}$ equal to $(\theta_1, 2\theta_2)$ in the case of t-coarsening. (In all three cases equality for each θ-component is up to an integer multiple of 2π.)

For any $\theta^1 \in \Theta_{\bar{s}} \equiv \Theta_h \cap [-\pi/2, \pi/2)^2$, we define θ^2, θ^3, and θ^4 by

$$(15) \qquad \theta^2 = \theta^1 - \begin{pmatrix} \text{sign}(\theta_1^1)\pi \\ \text{sign}(\theta_2^1)\pi \end{pmatrix}, \ \theta^3 = \theta^1 - \begin{pmatrix} 0 \\ \text{sign}(\theta_2^1)\pi \end{pmatrix}, \ \theta^4 = \theta^1 - \begin{pmatrix} \text{sign}(\theta_1^1)\pi \\ 0 \end{pmatrix},$$

where $\text{sign}(t)$ is -1 if $t \leq 0$ and $+1$ otherwise. For any $\theta \in \Theta_{\bar{s}}$, we define the vector $\Psi_h(\theta) = \left(\psi_h(\theta^1), \psi_h(\theta^2), \psi_h(\theta^3), \psi_h(\theta^4)\right)^T$. Any periodic fine-grid function e_h – the error, for example – can be written as $e_h = \sum e_{h,\theta}^T \Psi_h(\theta)$, where the summation is over all $\theta \in \Theta_{\bar{s}}$ and where $e_{h,\theta}^T$ is a vector of dimension four. The linear space spanned by $\Psi_h(\theta)$ is invariant under the two-grid operator. If the initial error is $e_{h,\theta}^T \Psi_h(\theta)$ then after application of the two-grid cycle it is given by $\hat{M}_h^H(\theta) e_{h,\theta}^T \Psi_h(\theta)$, with $\hat{M}_h^H(\theta)$ a 4×4 matrix. The latter is called the *symbol* of the two-grid operator. Let $\hat{S}_h(\theta)$, $\hat{I}_h^H(\theta)$, $\hat{I}_H^h(\theta)$, $\hat{L}_h(\theta)$, and $\hat{L}_H(\theta)$ denote the symbols of the smoothing operator, restriction operator, prolongation operator, fine-grid PDE operator, and coarse-grid PDE operator. $\hat{M}_h^H(\theta)$ is then easily found to be

$$(16) \qquad \hat{M}_h^H(\theta) = \hat{S}_h^{\nu 2}(\theta) \left(I_h - \hat{I}_H^h(\theta) \hat{L}_H^{-1}(\bar{\theta}) \hat{I}_h^H(\theta) F_h \hat{L}_h(\theta) \right) \hat{S}_h^{\nu 1}(\theta).$$

The quality of a particular smoothing method is often judged by its so-called *Fourier smoothing factor* (see, e.g., [17, p. 149])

$$(17) \qquad \mu = \max\{\kappa(Q(\theta)\hat{S}_h(\theta)) : \theta \in \Theta_{\bar{s}}\},$$

where κ is the spectral radius. Matrix $Q(\theta)$ is a diagonal matrix and expresses the projection of $\Psi_h(\theta)$ onto the corresponding space of high frequencies. For standard coarsening, for x-coarsening, and for t-coarsening, respectively, $Q(\theta)$ is given by diag $(\delta(\theta_1) \cdot \delta(\theta_2), 1, 1, 1)$, diag $(\delta(\theta_1), 1, \delta(\theta_1), 1)$, and diag $(\delta(\theta_2), 1, 1, \delta(\theta_2))$. Function $\delta(\theta_\alpha)$ is 0 except when $\theta_\alpha = -\pi/2$, in which case it is 1. An indicative measure of convergence of the entire two-grid cycle, taking both smoothing and coarse grid correction into account, is given by

$$(18) \qquad \rho = \max\{\kappa(\hat{M}_h^H(\theta)) : \theta \in \Theta_{\bar{s}}\}.$$

The value of ρ is an approximation to the spectral radius of the iteration matrix M_h^H and usually shows very good agreement with actual convergence factors obtained on Ω_h. Its computation is straightforward, by numerically computing $\kappa(\hat{M}_h^H(\theta))$ and by optimizing this over the discrete set $\Theta_{\bar{s}}$. (Note that usually $\theta = 0$ is excluded from the above range, since, as mentioned in [4, p. 13], $\hat{L}_H(0)$ is most often 0. This causes no problems, however, since $\hat{I}_h^H(0)\hat{L}_h(0)$ is rank deficient too, in such a way that $\lim_{\theta \to 0} \kappa(\hat{M}_h^H(\theta))$ is finite.)

Our definitions of μ and ρ are different from the ones found in [13], [3], and [4]. There, the following smoothing factor and spectral radius are defined

$$(19) \qquad \bar{\mu} = \sup\{\kappa(Q(\theta)\hat{S}_h(\theta)) : \theta \in \Theta_{\bar{s}}\} \quad \text{and} \quad \bar{\rho} = \sup\{\kappa(\hat{M}_h^H(\theta)) : \theta \in \Theta_{\bar{s}}\},$$

with $\Theta_{\bar{s}} = (-\pi/2, \pi/2)^2$. One naturally arrives at these formulae by neglecting the boundaries and the boundary conditions of the PDE problem and by considering the operators as operators on infinite grids. The computation of $\bar{\mu}$ and $\bar{\rho}$ is more difficult numerically than the computation of μ and ρ, since they involve finding a maximum over an infinite set. Obviously, for the same h and H it follows that $\mu \leq \bar{\mu}$ and $\rho \leq \bar{\rho}$. However, for small enough values of the mesh width we may expect both to become equal.

4.3. Fourier mode symbols. For the reader's convenience, we shall recall the precise formulae for the operator symbols. Let the stencils of L_h, L_H, I_H^h, and I_h^H be given, respectively, by

$$
(20) \qquad
\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & s_k & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}, \quad
\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & S_k & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}, \quad
\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & r_k & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}, \quad
\begin{matrix} \cdot & & \cdot \\ \cdot & p_k & \cdot \\ \cdot & & \cdot \end{matrix}
$$

where k ranges over an index set $I \subset \mathbb{Z}^2$. It is easily verified that the Fourier modes $\psi_h(\theta)$ and $\psi_H(\theta)$ are eigenfunctions of L_h and L_H [13, pp. 121–122], with eigenvalues

$$
(21) \qquad \tilde{L}_h(\theta) = \sum_{k \in I} s_k \exp(i\, k \cdot \theta) \ \text{ and } \ \tilde{L}_H(\theta) = \sum_{k \in I} S_k \exp(i\, k \cdot \theta).
$$

The Fourier symbol of the fine-grid PDE operator is given by

$$
(22) \qquad \hat{L}_h(\theta) = \mathrm{diag}\left(\tilde{L}_h(\theta^1), \tilde{L}_h(\theta^2), \tilde{L}_h(\theta^3), \tilde{L}_h(\theta^4) \right),
$$

and the Fourier symbol of the coarse-grid PDE operator by

$$
(23) \quad \hat{L}_H(\theta) = \left(\tilde{L}_H(\bar{\theta}^1) \right), \quad \mathrm{diag}\left(\tilde{L}_H(\bar{\theta}^1), \tilde{L}_H(\bar{\theta}^3) \right), \quad \mathrm{diag}\left(\tilde{L}_H(\bar{\theta}^1), \tilde{L}_H(\bar{\theta}^4) \right),
$$

where the latter is for standard-, x-, and t-coarsening, respectively. Note that (23) is constructed by taking the coarse-grid aliasing of Fourier modes into account.

A general formula for the red-black smoother's symbol can easily be derived following the analysis in [17, pp. 150–151]. Red-black smoothing applied to Fourier mode $\psi_h(\theta)$ results in a function the value of which is given by $\alpha(\theta)\psi_h(\theta)$ at the red points and by $\beta(\theta)\psi_h(\theta)$ at the black points. $\alpha(\theta)$ and $\beta(\theta)$ are given by

$$
(24) \qquad \alpha(\theta) = -\sum_{k \in I_0} s_k \, \exp(i\, k \cdot \theta)/s_{(0,0)},
$$

$$
(25) \qquad \beta(\theta) = -\left(\sum_{|k|=\text{odd}} s_k\, \alpha(\theta) \exp(i\, k \cdot \theta) + \sum_{0 \neq |k|=\text{even}} s_k \, \exp(i\, k \cdot \theta) \right) / s_{(0,0)},
$$

with $I_0 = I \setminus \{(0,0)\}$ and $|k| = |k_1| + |k_2|$. By some elementary algebra, one obtains

$$
(26) \quad \hat{S}_h(\theta) = \frac{1}{2}
\begin{pmatrix}
\alpha(\theta^1)+\beta(\theta^1) & \alpha(\theta^2)-\beta(\theta^2) & 0 & 0 \\
\alpha(\theta^1)-\beta(\theta^1) & \alpha(\theta^2)+\beta(\theta^2) & 0 & 0 \\
0 & 0 & \alpha(\theta^3)+\beta(\theta^3) & \alpha(\theta^4)-\beta(\theta^4) \\
0 & 0 & \alpha(\theta^3)-\beta(\theta^3) & \alpha(\theta^4)+\beta(\theta^4)
\end{pmatrix}.
$$

The restriction and prolongation operators connect fine- and coarse-grid Fourier modes. The following formulae are taken from [13, p. 122]

$$
(27) \qquad I_h^H \psi_h(\theta) = \tilde{I}_h^H(\theta)\psi_H(\bar{\theta}), \quad I_H^h \psi_H(\theta) = \sum_{\alpha} \tilde{I}_H^h(\theta^\alpha)\psi_h(\theta^\alpha).
$$

The summation in the last formula is over the set of fine-grid modes that coincides with $\psi_H(\theta)$ on the coarse grid. With the stencils of restriction and prolongation given in (20), one finds [13, p. 122]

$$(28) \qquad \tilde{I}_h^H(\theta) = \sum_{k \in I} r_k \exp(i\, k \cdot \theta), \quad \tilde{I}_H^h(\theta) = \frac{h_1 h_2}{H_1 H_2} \sum_{k \in I} p_{-k} \exp(i\, k \cdot \theta) \ .$$

The Fourier symbols of the prolongation operators for standard-, x-, and t-coarsening are then given by

$$(29) \quad \hat{I}_H^h(\theta) = \begin{pmatrix} \tilde{I}_H^h(\theta^1) \\ \tilde{I}_H^h(\theta^2) \\ \tilde{I}_H^h(\theta^3) \\ \tilde{I}_H^h(\theta^4) \end{pmatrix}, \quad \begin{pmatrix} \tilde{I}_H^h(\theta^1) & 0 \\ 0 & \tilde{I}_H^h(\theta^2) \\ 0 & \tilde{I}_H^h(\theta^3) \\ \tilde{I}_H^h(\theta^4) & 0 \end{pmatrix}, \quad \begin{pmatrix} \tilde{I}_H^h(\theta^1) & 0 \\ 0 & \tilde{I}_H^h(\theta^2) \\ \tilde{I}_H^h(\theta^3) & 0 \\ 0 & \tilde{I}_H^h(\theta^4) \end{pmatrix} \ .$$

For the operators considered in this paper, it can be verified that the restriction symbol is just the transpose of the prolongation symbol

$$(30) \qquad \hat{I}_h^H(\theta) = \left(\hat{I}_H^h(\theta) \right)^T \ .$$

4.4. Fourier mode analysis results. Exponential Fourier mode analysis was performed for the two-grid methods with BDF1, BDF2, and Crank–Nicolson discretization. The discretization scheme dependent values of $\tilde{L}_h(\theta)$, $\alpha(\theta)$, and $\beta(\theta)$ are given below. The formula for $\tilde{L}_H(\theta)$ is easily found from the one for $\tilde{L}_h(\theta)$ by replacing λ_h by λ_H. For the BDF1 method,

$$(31) \qquad \tilde{L}_h(\theta) = 1 + 2\lambda_h(1 - \cos(\theta_1)) - \exp(-i\theta_2),$$

$$(32) \qquad \alpha(\theta) = (2\lambda_h \cos(\theta_1) + \exp(-i\theta_2))/(1 + 2\lambda_h),$$

$$(33) \qquad \beta(\theta) = \alpha^2(\theta);$$

for the BDF2 method

$$(34) \qquad \tilde{L}_h(\theta) = 3/2 + 2\lambda_h(1 - \cos(\theta_1)) + 1/2 \exp(-i\theta_2)(\exp(-i\theta_2) - 4),$$

$$(35) \qquad \alpha(\theta) = (2\lambda_h \cos(\theta_1) + 1/2 \exp(-i\theta_2)(4 - \exp(-i\theta_2)))/(3/2 + 2\lambda_h),$$

$$(36) \qquad \beta(\theta) = (2\lambda_h \cos(\theta_1)\alpha(\theta) + 1/2 \exp(-i\theta_2)(4\alpha(\theta) - \exp(-i\theta_2)))/(3/2 + 2\lambda_h);$$

and, finally, for the Crank–Nicolson scheme

$$(37) \qquad \tilde{L}_h(\theta) = (1 + \lambda_h(1 - \cos(\theta_1)))(1 + \exp(-i\theta_2)) - 2\exp(-i\theta_2),$$

$$(38) \qquad \alpha(\theta) = (\lambda_h \cos(\theta_1)(1 + \exp(-i\theta_2)) + (1 - \lambda_h)\exp(-i\theta_2))/(1 + \lambda_h),$$

$$(39) \qquad \beta(\theta) = (\lambda_h \cos(\theta_1)(\alpha(\theta) + \exp(-i\theta_2)) + (1 - \lambda_h)\alpha(\theta)\exp(-i\theta_2))/(1 + \lambda_h).$$

Smoothing factors μ are presented as a function of λ_h for each of the three methods in Fig. 2. They were computed with $n_1 = n_2 = 128$, over the range $2^{-8} \leq \lambda_h \leq 2^8$, sampled at intervals of $\log_2 \lambda_h = 0.25$. In the three pictures we show results obtained with standard coarsening (dotted line), with x-coarsening (solid line), and with t-coarsening (dashed line).
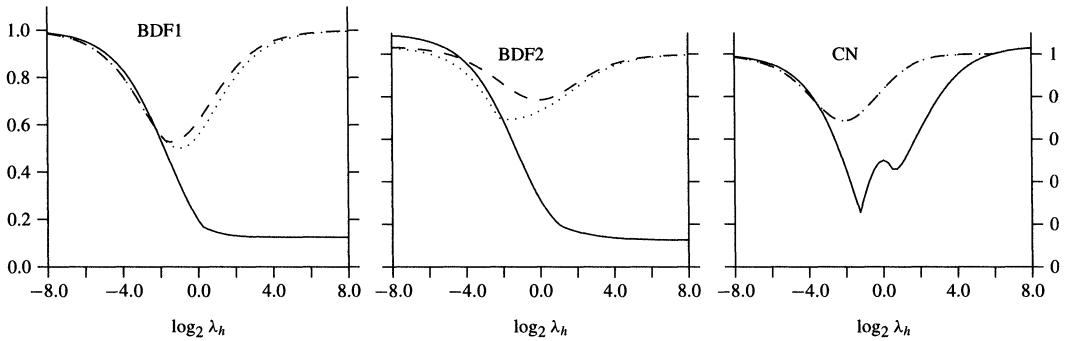
FIG. 2. *Smoothing factor μ for backward Euler method (BDF1), second-order backward differentiation method (BDF2), and Crank–Nicolson method (CN); solid line: x-coarsening, dashed line: t-coarsening, dotted line: standard coarsening.*

Figure 3 shows the computed values of the two-grid convergence factor ρ also determined for $n_1 = n_2 = 128$.

Figure 4 shows experimentally computed two-grid convergence factors, obtained with an implementation of the two-grid algorithm. The results prove to be in very close agreement with the results obtained from the Fourier mode analysis (Fig. 3, top left).

4.5. Discussion. In the case of *semicoarsening in space* the magnitude of the two-grid convergence factor is mainly determined by the smoothing characteristics of the red-black relaxation scheme. The latter follows from the qualitative similarity of the corresponding solid lines in Figs. 2 and 3. The red-black relaxation method performs very badly for small values of λ_h. It performs satisfactorily for large λ_h, with the BDF1 and the BDF2 methods. With these two discretization methods the fully discrete problem corresponds to a large set of essentially decoupled Poisson problems, one at each time level. The multigrid method with red-black relaxation and standard spatial coarsening is known to be a very good solver for such problems. For one-dimensional problems in particular, it is known to be an exact direct solver. The latter argument obviously does not hold for the Crank–Nicolson discretization, for which red-black relaxation fails as a smoother, even for large λ_h. (High frequency Fourier error components with $\theta_2 \simeq \pi$ are hardly smoothed.)

When combined with *semicoarsening in time*, red-black relaxation loses all of its smoothing qualities; see Fig. 2. The fast convergence of the two-grid scheme for small λ_h therefore cannot be explained by standard multigrid arguments. Instead, it is due to the fact that in this case the PDE degenerates into a set of almost decoupled linear ordinary differential equations (ODEs). Time discretization with BDF1 and Crank–Nicolson leads, in the limiting case of $\lambda_h = 0$, to a large set of bidiagonal systems, whose equations are of the following form: $(u_{i,j} - u_{i,j-1})/h_2 = f_{i,j}$. It can be shown that these equations are solved exactly by the two-grid method and also by the multigrid method, with our choice of restriction and prolongation operators, and with the use of red-black relaxation. The method is then essentially equivalent to a cyclic reduction method. The latter argument does not hold for the BDF2 discretization, however. For this method ρ is very small but different from zero even when $\lambda_h = 0$. Finally, we should point out the rapid growth of ρ with increasing λ_h in the Crank–Nicolson case, which is due to Fourier components $\theta = (\theta_1, \theta_2)$ with $\theta_2 \approx \pi/2$. Such components are disproportionally magnified by the coarse grid correction, since $\tilde{L}_h(\theta) = \mathcal{O}(\lambda_h)$ while $\tilde{L}_H(\theta) = \mathcal{O}(1)$.
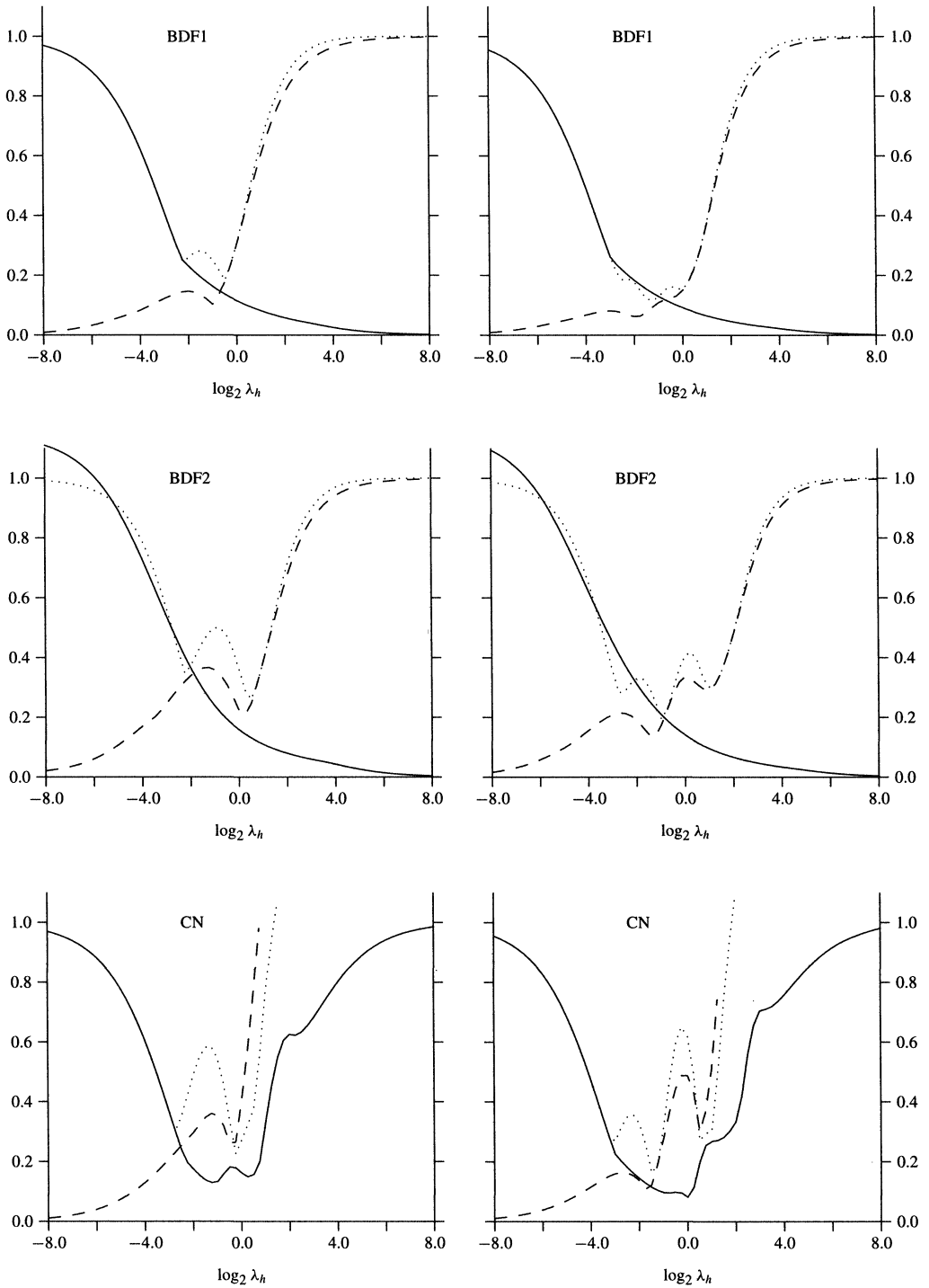
FIG. 3. *Two-grid Fourier mode convergence factor ρ for backward Euler method ($BDF1$), second-order backward differentiation method ($BDF2$), and Crank–Nicolson method (CN), with two smoothing steps (left) and 3 smoothing steps (right); solid line: x-coarsening, dashed line: t-coarsening, dotted line: standard coarsening.*
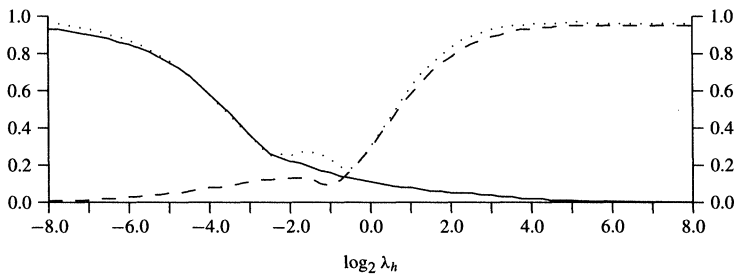
FIG. 4. *Experimentally computed two-grid convergence factor ρ for BDF1 method with two smoothing steps; solid line: x-coarsening, dashed line: t-coarsening, dotted line: standard coarsening.*
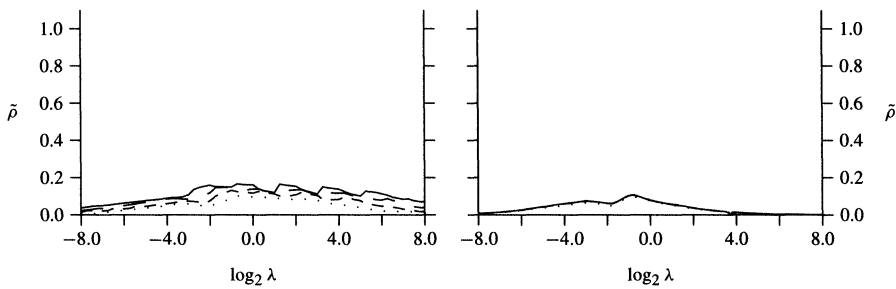

FIG. 5. *Experimental results for BDF1 method with three smoothing steps for V-cycle (left) and F-cycle (right), one space dimension; solid line: 256 × 256, long dashed line: 128 × 128, short dashed line: 64 × 64, dotted line: 32 × 32.*

The method with *standard coarsening in space and in time* is nowhere optimal. It performs very badly, except for a small region of λ_h around 1.

5. Numerical experiments. We have implemented the multigrid method described in §3 and applied it to both the one- and two- (space)-dimensional model problems. In our experiments, the number of smoothing steps used was given by $\nu_1 = 2$ and $\nu_2 = 1$. We tested V, F, and W multigrid cycle types. In order to test for grid-independent convergence rates, we performed the tests on grids of varying size. The value of λ_{crit} was chosen on the basis of the results of the Fourier analysis as the value at which curves for t-coarsening and x-coarsening coincide. We use as a performance measure the mean convergence rate $\tilde{\rho}$

$$(40) \qquad \tilde{\rho} = \sqrt[n]{\frac{\|d^{(n)}\|_2}{\|d^{(0)}\|_2}},$$

where n is the number of the final iteration in which convergence is achieved. The convergence criterion used in all cases was $\|d\|_2/N \leq 10^{-9}$, where N is the total number of variables on the finest grid.

Figure 5 shows the results obtained for equation (1) with one space dimension using BDF1 for the discretization of the time derivative. The range of λ, where λ denotes the "discrete anisotropy" on the *finest* grid, is given by $-8 \leq \log_2 \lambda \leq 8$, sampled in increments of 0.25. The rate of convergence of the BDF1 scheme using the V-cycle is evidently quite good for all values of λ, peaking at about 0.2. However, the grid sizes tested do not seem to yield a
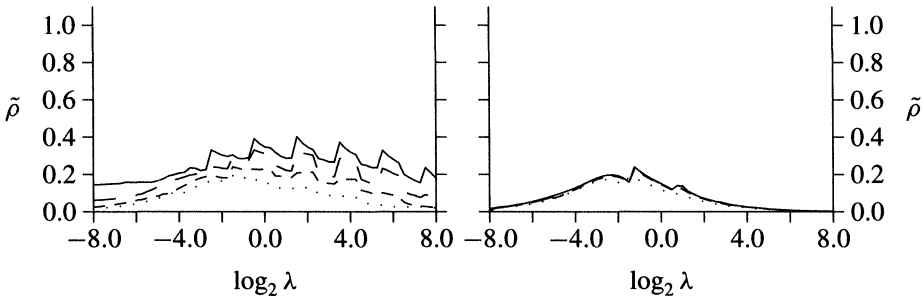
FIG. 6. *Experimental results for BDF2 method with three smoothing steps for V-cycle (left) and F-cycle (right), one space dimension; solid line:* 256×256, *long dashed line:* 128×128, *short dashed line:* 64×64, *dotted line:* 32×32.
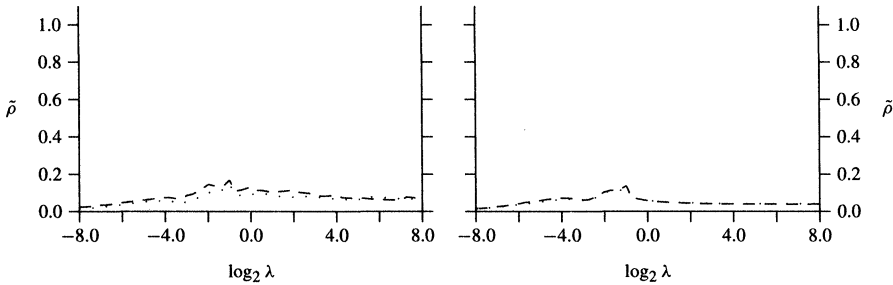

FIG. 7. *Experimental results for BDF1 method with three smoothing steps for V-cycle (left) and F-cycle (right), two space dimensions; dashed line:* $64 \times 64 \times 64$, *dotted line:* $32 \times 32 \times 32$.

grid-independent value of $\tilde{\rho}$. The F-cycle result is somewhat better than that of the V-cycle, indicating that the coarse-grid correction is superior. Here a grid-independent convergence rate is already achieved on the coarsest grid tested: the results for grids with sidelengths greater than 64 are indistinguishable. Experiments with W-cycles showed no further improvement over the F-cycle. For extreme values of λ we observed that the method approaches an exact solver, as expected.

The situation is similar for the BDF2 method (Fig. 6), where, however, the V-cycle convergence rate is not as good, attaining values of as much as 0.4 on the largest grid tested. Here the curves show pronounced "wiggles," which are due to the variation in the sequence of semicoarsening directions for different values of λ on the finest grid. This demonstrates a certain degree of sensitivity of the overall convergence speed on a particular choice of semicoarsening direction within the multigrid cycle. Here again, F-cycles are necessary to achieve grid-independent convergence rates, which yield $\tilde{\rho} \approx 0.2$ in the worst case. Various modifications to the V-cycle, including increasing the number of smoothing steps on coarser grids, although improving the convergence rate, did not result in grid-independent convergence.

Figures 7 and 8 show analogous results obtained for the three-dimensional (x, y, and t) model problem. As in the previous case, F-cycles achieve grid-independent convergence speeds of better than 0.2, whereas the V-cycle returns less good results. The most significant difference to the previous result is the asymmetry of the curves: the convergence rates for
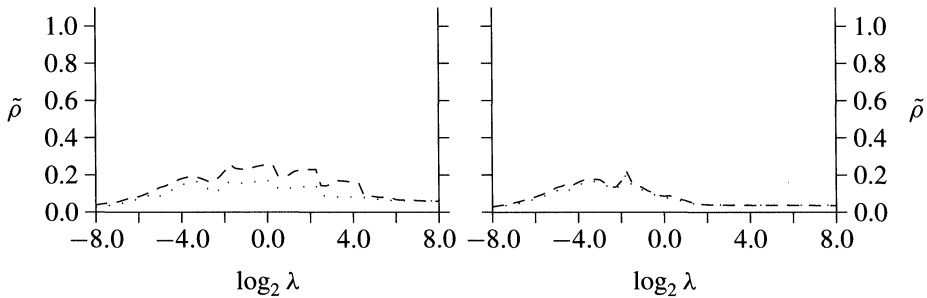
FIG. 8. *Experimental results for BDF2 method with three smoothing steps for V-cycle (left) and F-cycle (right), two space dimensions; dashed line: 64 × 64 × 64, dotted line: 32 × 32 × 32.*

TABLE 2
Averaged convergence factor of space-time multigrid V(2, 1)-cycle on rectangular grids extending in the time direction (one-dimensional model problem, BDF1 discretization).

Grid size	32 × 32	32 × 64	32 × 128	32 × 256	32 × 512	32 × 1024
$\lambda = 1/64$	0.02	0.03	0.05	0.06	0.07	0.09
$\lambda = 1/4$	0.05	0.08	0.11	0.15	0.14	0.14
$\lambda = 1/2$	0.08	0.13	0.14	0.13	0.13	0.13
$\lambda = 1$	0.10	0.11	0.12	0.13	0.13	0.12
$\lambda = 2$	0.09	0.10	0.10	0.10	0.09	0.08
$\lambda = 4$	0.08	0.09	0.09	0.09	0.08	0.08
$\lambda = 64$	0.02	0.02	0.02	0.02	0.02	0.02

$\lambda \gg 1$ are no longer close to 0 but approach an asymptotic value of about 0.06. In this case, the equation degenerates into a set of almost decoupled two-dimensional elliptic Poisson equations. The multigrid method treats the system accordingly and achieves the typical convergence rate for this well-studied problem.

For practical problems, one is often interested in computing a very large number of time steps. The PDE solution is then to be found on a rectangular space-time grid with many more grid lines in the time direction than in the spatial directions. The results of such an experiment are presented in Table 2, for the two-dimensional model problem with BDF1 discretization. The results show that there is essentially no change in convergence caused when the integration interval is extended. Thus we conclude that the method performs equally well, regardless of the length of the time integration.

6. Parallel complexity. Parallel complexity is a theoretical measure of an algorithm based on the assumption that an unlimited number of processors is available for its execution and disregarding communication requirements. It describes, under the above assumptions, the asymptotic dependency of the computation time of the algorithm as a function of the size of the input.

The multigrid method presented in this paper is designed to be fully parallelizable. This is achieved by its requiring only pointwise smoothing rather than more sophisticated techniques such as incomplete decomposition methods. We may therefore assert the parallel complexity of the multigrid V-cycle to be of the order of

(41) $$\mathcal{O}(\log N_s + \log N_t) ,$$

where N_s denotes the spatial sidelength of the space-time grid and N_t the sidelength in the time direction. This is easily seen by observing the parallel complexity of each multigrid operation at any grid level to be of the order $\mathcal{O}(1)$ and the number of grid levels processed during the cycle to be $\mathcal{O}(\log N_s + \log N_t)$.

Standard multigrid arguments show the multigrid nested iteration (FMG) to achieve discretization accuracy with a constant number of V-cycles per grid level, given that the convergence rate is independent of the grid size. Summing the total number of grids visited by the FMG-V iteration gives an overall parallel complexity of

$$\text{(42)} \qquad\qquad \mathcal{O}\left((\log N_s + \log N_t)^2\right) \ .$$

Similarly, we obtain as the parallel complexity of the space-time FMG-F cycle

$$\text{(43)} \qquad\qquad \mathcal{O}\left((\log N_s + \log N_t)^3\right) \ .$$

Note that the same parallel complexities are found when simply performing V-cycles or F-cycles to convergence.

This result compares favourably with that of the standard time-stepping technique, where problems at each time step are solved with a parallel elliptic FMG or multigrid cycling scheme, but the time steps are processed sequentially. For this method parallel complexities of $\mathcal{O}(N_t \cdot \log^2 N_s)$ for V-cycles and $\mathcal{O}(N_t \cdot \log^3 N_s)$ for F-cycles are obtained. Finally, we wish to recall the complexity of the multigrid waveform relaxation method. In [10], it is shown that the complexity of the FMG-V waveform algorithm is given by

$$\text{(44)} \qquad\qquad \mathcal{O}\left(\log^2 N_s \cdot \log N_t\right) \ ,$$

which is a polylogarithmic function of similar order as (43).

7. Conclusions.
In this paper a multigrid method for the solution of parabolic PDEs has been presented. The novelty of the scheme stems from its treatment of the entire space-time problem, as opposed to the standard time-stepping approach. The method is characterized by a parameter-dependent choice of coarsening strategy, together with appropriate prolongation and restriction operators.

Results from Fourier two-grid analysis show that good rates of convergence can be expected for the scheme and that a naive approach to the problem fails. Numerical results are presented for the one- and two-dimensional heat equations for both first- and second-order discretizations of the time derivative. These proved to converge quickly, although at present the F-cycle seems to be necessary to achieve grid-independent rates.

The algorithm is fully parallelizable in all problem dimensions, i.e., in both space and time, in contrast to standard time-marching methods. It is shown that this leads to an improved parallel complexity for parabolic problems. In addition, MIMD implementations of the method will benefit from the increased parallelism and the improved surface-to-volume ratio as has already been shown, albeit in a slightly different context, in [16].

Further work will include the implementation of the space-time multigrid method on a massively parallel computer in order to determine whether the theoretically derived parallel complexities can be achieved in practice. In addition, an FAS-like version of the method will be developed in order to be able to solve nonlinear problems.

Appendix. We give the prolongation and restriction operators for the three-dimensional space-time problem. To this end, we adopt the notation that a sequence of stencils is given, whereby successive stencils are applied to successive time steps. Each stencil has the standard interpretation on the space grid at one time step.

Prolongation.

Case 1 (semicoarsening in t direction).

$$
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.
$$

Case 2 (semicoarsening in x and y directions).

$$
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad
\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.
$$

Case 3 (standard coarsening in x, y, and t).

$$
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad
\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad
\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.
$$

Restriction.

Case 1 (semicoarsening in t direction).

$$
\frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad
\frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.
$$

Case 2 (semicoarsening in x and y directions).

$$
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad
\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.
$$

Case 3 (standard coarsening in x, y, and t).

$$
\frac{1}{32}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad
\frac{1}{32}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.
$$

REFERENCES

[1] P. BASTIAN, J. BURMEISTER, AND G. HORTON, *Implementation of a parallel multigrid method for parabolic partial differential equations*, in Parallel Algorithms for PDEs, Proc. 6th GAMM Seminar Kiel, January 19–21, 1990, W. Hackbusch, ed., Wiesbaden, 1990, Vieweg Verlag, Braunschweig, pp. 18–27.

[2] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[3] ———, *Multigrid techniques: 1984 guide, with application to fluid dynamics*, GMD Studien 85, GMD-AIW, Postfach 1240, D-5205 St.-Augustin, Germany, 1984.

[4] ———, *Rigorous quantitative analysis of multigrid*, I: *Constant coefficients two-level cycle with L_2-norm*, SIAM J. Numer. Anal., 31 (1994), pp. 1695–1730.

[5] W. BRIGGS, *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1987.

 [6] J. BURMEISTER AND G. HORTON, *Time-parallel multigrid solution of the Navier-Stokes equations*, in Multigrid Methods III, Proc. 3rd European Multigrid Conf., Bonn, 1990, W. Hackbusch and U. Trottenberg, eds., Birkhäuser Verlag, Basel, 1991, pp. 155–166.

 [7] W. HACKBUSCH, *Parabolic multigrid methods*, in Computing Methods in Applied Sciences and Engineering VI, R. Glowinski and J.-L. Lions, eds., North–Holland, Amsterdam, 1984, pp. 189–197.

 [8] ———, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, 1985.

 [9] G. HORTON, *The time-parallel multigrid method*, Comm. Appl. Numer. Methods, 8 (1992), pp. 585–595.

[10] G. HORTON, S. VANDEWALLE, AND P. WORLEY, *An algorithm with polylog parallel complexity for solving parabolic partial differential eqations*, SIAM J. Sci. Comput., 16 (1995), pp. 531–541.

[11] C. J. KUO AND B. LEVY, *Two-color Fourier analysis of the multigrid method with red-black Gauss-Seidel smoothing*, Appl. Math. Comput., 29 (1989), pp. 69–87.

[12] C. LUBICH AND A. OSTERMANN, *Multigrid dynamic iteration for parabolic equations*, BIT, 27 (1987), pp. 216–234.

[13] K. STÜBEN AND U. TROTTENBERG, *Multigrid methods: Fundamental algorithms, model problem analysis and applications*, GMD-Studien 96, Gesellschaft für Mathematik und Datenverarbeitung, St.-Augustin, Germany, December 1984.

[14] S. VANDEWALLE AND G. HORTON, *Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods*, Computing (1995).

[15] S. VANDEWALLE AND R. PIESSENS, *Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1330–1346.

[16] S. VANDEWALLE AND E. VAN DE VELDE, *Space-time concurrent multigrid waveform relaxation*, Ann. Numer. Math., 1 (1994), pp. 347–363.

[17] P. WESSELING, *An Introduction to Multigrid Methods*, John Wiley & Sons Ltd., Chichester, England, 1992.

[18] D. WOMBLE, *A time-stepping algorithm for parallel computers*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 824–837.