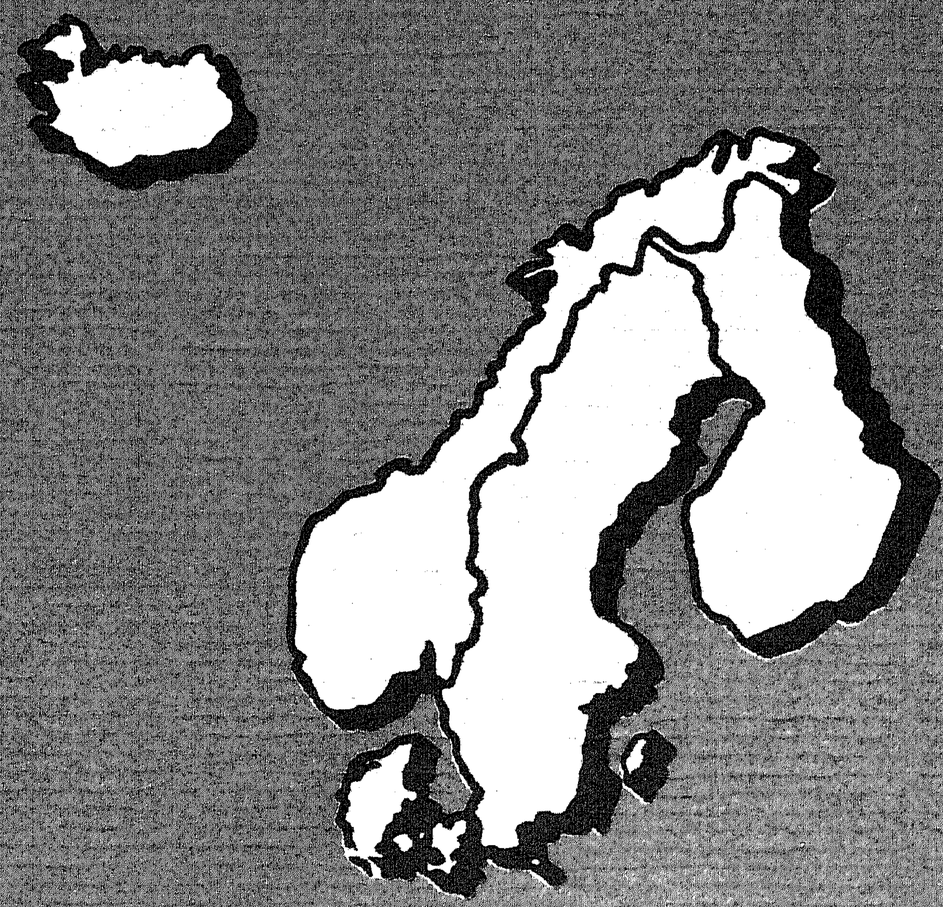
BIT

J. G. Verwer

The application of iterated defect correction to the LOD method for parabolic equations



Reprint

Bind 19 – Hefte nr. 3 – 1979

BIT 19 (1979), 384–394

THE APPLICATION OF ITERATED DEFECT CORRECTION TO THE *LOD* METHOD FOR PARABOLIC EQUATIONS

J. G. VERWER

Abstract.

The paper is concerned with the numerical solution of the initial boundary problem for a class of multi-dimensional parabolic partial differential equatic particular the time-integration of semi-discrete equations is investigated. An atte made to develop integration formulas being computationally attractive and c accuracy, while possessing unconditional stability properties. To that end iterated correction is applied to the LOD method. The convergence properties of this proc investigated. Numerical experiments are reported.

Key words and phrases: Numerical analysis, Parabolic partial differential equ Locally one-dimensional method, Iterated defect correction.

1. Introduction.

Let Ω denote a bounded and path-connected region in the k-dimen $(x_1, \ldots, x_k)$-space with boundary $\partial\Omega$ parallel to the coordinate axes. Consid parabolic partial differential equation of the non-linear type

$$u_t = \sum_{i=1}^{k} F_i(t, x_1, \ldots, x_k, u, u_{x_i}, u_{x_i x_i}) , \tag{1.1}$$

defined in the set $(0, T) \times \Omega$. Let a boundary condition be given in the fo

$$a(t, x_1, \ldots, x_k)u + b(t, x_1, \ldots, x_k)u_n = c(t, x_1, \ldots, x_k) , \tag{1.2}$$

$(t, x_1, \ldots, x_k) \in (0, T] \times \partial\Omega$, u_n normal derivative, and assume an initial func given at $t = 0$. In this paper we are concerned with the numerical solution initial boundary value problem when brought in an explicit, *semi-discretizec* i.e., we primarily discuss the numerical integration of the system of *or differential equations*

$$y' = f(t, y), \qquad t \in (0, T), \qquad y(0) = y_0 , \tag{1.3}$$

being obtained from discretizing the space variables in (1.1)–(1.2). It is as that the semi-discretization has been performed in such a way that f *satisf linear splitting relation*

$$f(t, y) = \sum_{i=1}^{k} f_i(t, y) , \tag{1.4}$$

Received August 18, 1978. Revised February 27, 1979.

where the ith *splitting function* f_i approximates the one-dimensional operator F_i (cf. [4]). Then each derivative $\partial f_i/\partial y$ is a band matrix, usually a *tridiagonal* one.

The paper has been written in order to investigate the application of iterated defect correction [2, 3] to the locally one-dimensional splitting formula [7,4]

$$\text{(1.5)} \qquad \begin{cases} y_{(0)} = y_\nu , \\ y_{(i)} = y_{(i-1)} + \tau f_i(t_{\nu+1}, y_{(i)}), \qquad i = 1(1)k , \\ y_{\nu+1} = y_{(k)} . \end{cases}$$

In this one-step integration formula $\tau = t_{\nu+1} - t_\nu$ denotes the steplength and y_ν denotes an approximation to the exact solution $y(t)$ of (1.3) at $t = t_\nu$. It is easy to see that the order of consistency of (1.5) is equal to 1 for every splitting (1.4). Observe that if $k=1$, (1.5) reduces to the implicit Euler formula.

The purpose of the investigation is to find integration formulas for systems (1.3)–(1.4), which are more *accurate* than the LOD formula (1.5) and which possess its attractive *unconditional stability* property [7,4], as well as its advantage of being *computationally attractive* (per integration step). The idea of iterated defect correction, when applied to (1.5), may lead to such integration formulas.

In our investigation we adopt the approach followed by Frank & Ueberhuber [2]. They investigated iterated defect correction for the efficient solution of stiff systems. Their basic formula is implicit Euler. Because our splitting formula is closely related to implicit Euler, many of their results carry over.

2. The IDEC-process for the LOD splitting formula.

In this section we shortly describe the IDEC (iterated defect correction) for the splitting formula (1.5). Details are omitted as these are clearly discussed in [2] and [6]. Let the solution of (1.3) be required on the interval $[0, T]$. Introduce the sequence of subintervals $[H_i, H_{i+1}]$, not necessarily equidistant, where $H_0 = 0$ and $H_{i_{\max}} = T$ for a suitable integer $i_{\max}$. We now restrict the discussion to the first subinterval $[0, H_1]$ on which we define the equidistant step points

$$\text{(2.1)} \qquad t_0 = 0, \quad t_\nu = \nu H_1/m, \quad \nu = 1(1)m ,$$

with $1 \leqq m \leqq 4$. For practical reasons we do not consider values of $m > 4$. Let j denote the iteration index of the IDEC. The process then consists of the following steps:

1°. Set $j=0$. Apply method (1.5) on the grid (2.1) to system (1.3) to obtain the row $\eta^0 = [\eta_0^0, \ldots, \eta_m^0]$ of approximation vectors η_ν^0. Here $\eta_0^0 = y_0$.

2°. Define the jth defect function

$$\text{(2.2)} \qquad d^j(t) = (P^j)'(t) - f(t, P^j(t)) ,$$

where $P^j(t)$ is the vector polynomial of degree $\leqq m$ interpolating η^j, i.e., $P^j(t_\nu) = \eta_\nu^j$ for $\nu = 0(1)m$, and compute the defects

(2.3) $$d^j(t_\nu) = (P^j)'(t_\nu) - f(t_\nu, \eta_\nu^j), \qquad \nu = 1(1)m .$$

3°. Define the initial value problem $y' = f(t,y) + d^j(t)$, $y(0) = y_0$ and apply (1.5) to this problem on the same grid (2.1) to obtain the row $\pi^j = [\pi_0^j, \ldots, \pi_m^j]$ with $\pi_0^j = y_0$. Herewith it is assumed that d^j has been added to the *first* splitting function f_1.

4°. Improve, i.e. compute the $(j+1)$th approximation row η^{j+1} by

(2.4) $$\eta^{j+1} = \eta^0 + \eta^j - \pi^j .$$

5°. Increase j and proceed with 2°.

We apply the *local connection* strategy [2], i.e., after the last iteration step on $[0, H_1]$ we simply repeat the whole process on $[H_1, H_2]$, and so on. The additional programming effort to implement the IDEC is small. The required derivative values $(P^j)'(t_\nu)$ are easily determined from differentiation of Lagrange's formula [1, p. 878]

(2.5) $$(P^j)'(t_\nu) = \tau^{-1} \sum_{\varkappa=0}^{m} w_{\nu\varkappa} P^j(t_\varkappa), \qquad \nu = 1(1)m .$$

For $m \leqq 4$ the constant weights $w_{\nu k}$ are given in [1, p.914].

It can be shown that the *fixed point* of our IDEC coincides with the solution of the polynomial collocation method in the step points (2.1), which, in turn, may be interpreted as the *fully implicit Runge–Kutta method*

(2.6) $$\eta_\nu^* = y_0 + \tau \sum_{\varkappa=1}^{m} \bar{w}_{\nu\varkappa} f(t_\varkappa, \eta_\varkappa^*), \qquad \nu = 1(1)m .$$

The coefficient matrix $\bar{W} = (\bar{w}_{\nu\varkappa})$ with $\nu, \varkappa = 1(1)m$ is the inverse of the weight matrix $W = (w_{\nu\varkappa})$. If $m > 1$, methods of this type are also called block methods. Each result η_ν^* is mth *order consistent*, i.e. the local truncation errors are of order $m+1$ in τ. Hence, it is expected that the order of the iterates η_ν^j is equal to or smaller than m. In our case the order of each iterate equals $\min(j+1, m)$. Thus the optimal order is reached after $m-1$ IDEC iterations.

For the stability test-equation $y' = \lambda y$, $y(0) = y_0$, $\lambda \in \mathbb{C}$, we have

(2.7) $$\eta_\nu^* = \varphi_\nu(s) y_0, \quad s = m\tau\lambda, \qquad \nu = 1(1)m ,$$

φ_ν being a rational function satisfying: $\varphi_\nu(s) \sim 1/s$, $\mathrm{Re}(s) \to -\infty$. The stability function of the method is φ_m which is A-acceptable for $m \leqq 2$. For $3 \leqq m \leqq 4$ the function φ_m is $A(\alpha)$-acceptable with α close to $\pi/2$ (cf. [2], fig. 6). As a consequence, the $1 \leqq m \leqq 4$-formulas (2.6) possess attractive *unconditional stability* properties for semi-discrete parabolic equations.

3. Convergence of the IDEC.

We investigate the convergence for the *linear test-model* being usually considered in the stability analysis of splitting methods [4], viz.

$$y' = Jy, \quad J = \sum_{i=1}^{k} J_i , \tag{3.1}$$

where the constant matrices J_i share the same eigensystem and are symmetric and negative definite. As in similar situations, it then suffices to consider a scalar equation [6]

$$y' = \lambda y = \lambda_1 y + \ldots + \lambda_k y, \quad \lambda_i < 0, \quad y(0) = y_0 , \tag{3.2}$$

where each λ_i represents the eigenvalues of J_i. The LOD method now yields

$$y_{\nu+1} = R(z_1, \ldots, z_k) y_\nu, \quad R(z_1, \ldots, z_k) = \prod_{i=1}^{k} (1 - z_i)^{-1}, \quad z_i = \tau \lambda_i . \tag{3.3}$$

When applied to the scalar equation $y' = \lambda y + d(t)$, the LOD method yields $y_{\nu+1} = R[y_\nu + \tau d(t_{\nu+1})]$, provided $d(t)$ has been added to $\lambda_1 y$ (compare point 3° of section 2). Because of the fact that implicit Euler gives recurrence relations of the same type (the stability function R is then replaced by $(1-z)^{-1}$, $z = z_1 + \ldots + z_k$), it thus follows that many results in the convergence analysis given in [2] carry over to our situation. In particular, when applied to (3.2) the IDEC process reduces to the recurrence relation

$$\tilde{\eta}^{j+1} = S_m \tilde{\eta}^j + V_m, \quad \tilde{\eta}^j = [\eta_1^j, \ldots, \eta_m^j]^T , \tag{3.4}$$

with S_m a constant matrix of order m and V_m a constant vector of length m. Obviously, we have *convergence* if the spectral radius $\sigma(S_m) < 1$. The explicit construction of S_m and V_m is carried out in appendix 1 of [2] (see also [6]). Here we do not need S_m explicitly, but merely state that its elements are functions of z_i, $i = 1(1)k$.

Let us proceed with the investigation of $\sigma(S_m)$ for $z_i < 0$, $i = 1(1)k$, and arbitrary splittings $z = z_1 + \ldots + z_k$, $k \geqq 2$. For the simple case $m = 1$ it can be shown analytically that, for $z_i < 0$, $0 < \sigma(S_1) < 1$. Further, $\sigma(S_1) \to 0$ if all $z_i \to 0$ and $\sigma(S_1) \to 1$ if all $z_i \to -\infty$. If $m > 1$, explicit expressions of $\sigma(S_m)$ are not available and we can only state a result for the two limit cases (cf. [2, 6]):

THEOREM. *For all* $m \geqq 1$ *the following relations hold*:
a) $\lim \sigma(S_m) = 0$ *if all* $z_i \to 0$,
b) $\lim \sigma(S_m) = 1$ *if all* $z_i \to -\infty$.

Let us now temporarily assume that indeed $\sigma(S_m) < 1$ for $z_i < 0$. Then, at first sight, for z_i-values close to zero we may expect a rapid IDEC convergence, whereas for the larger ones convergence is expected to be slow. We shall consider this point in more detail. The initial approximation $\tilde{\eta}^0$ in (3.4) is given by $\tilde{\eta}^0 = [Ry_0, \ldots, R^m y_0]^T$. According to (2.7), the νth element of the fixed point vector $\tilde{\eta}^* = (I - S_m)^{-1} V_m$ can be expressed as $\eta_\nu^* = \varphi_\nu(mz) y_0$. Hence, the νth initial iteration error, say $\varepsilon_\nu^0 = \eta_\nu^* - \eta_\nu^0$, is given by

$$\varepsilon_\nu^0 = [\varphi_\nu(mz) - R^\nu]y_0 \,. \tag{3.5}$$

From the damping properties of φ and R it thus follows that components of the solution of (3.1) belonging to large negative eigenvalues are damped in the initial IDEC iteration error. Often, these components (approximations to Fourier coefficients belonging to higher harmonics [6]) are already small in the initial vector of (3.1). Hence, at least in the initial phase of the iteration, it is expected that the decrease of the iteration errors ε_ν^j is governed by the small z_i-values. Unfortunately, the speed of convergence *decreases* if j increases. This is due to the fact that, for (3.1), the errors ε_ν^j tend to lie in subspaces spanned by dominant eigenvectors of J. For these eigenvectors convergence is slow. Consequently, it is of no use to perform a large number of iterations. We illustrate this phenomenon in the example at the end of this section.

For $k=2$ and $m=1(1)4$ we computed $\sigma(S_m)$ numerically at the points (z_1, z_2), $z_i = -l/4$, $l=0(1)80$. All computed σ are smaller than one. A plot of the maximal σ-values as a function of z_1 is given in fig. 3.1. We see that for small z-values the speed of convergence decreases with increasing m.

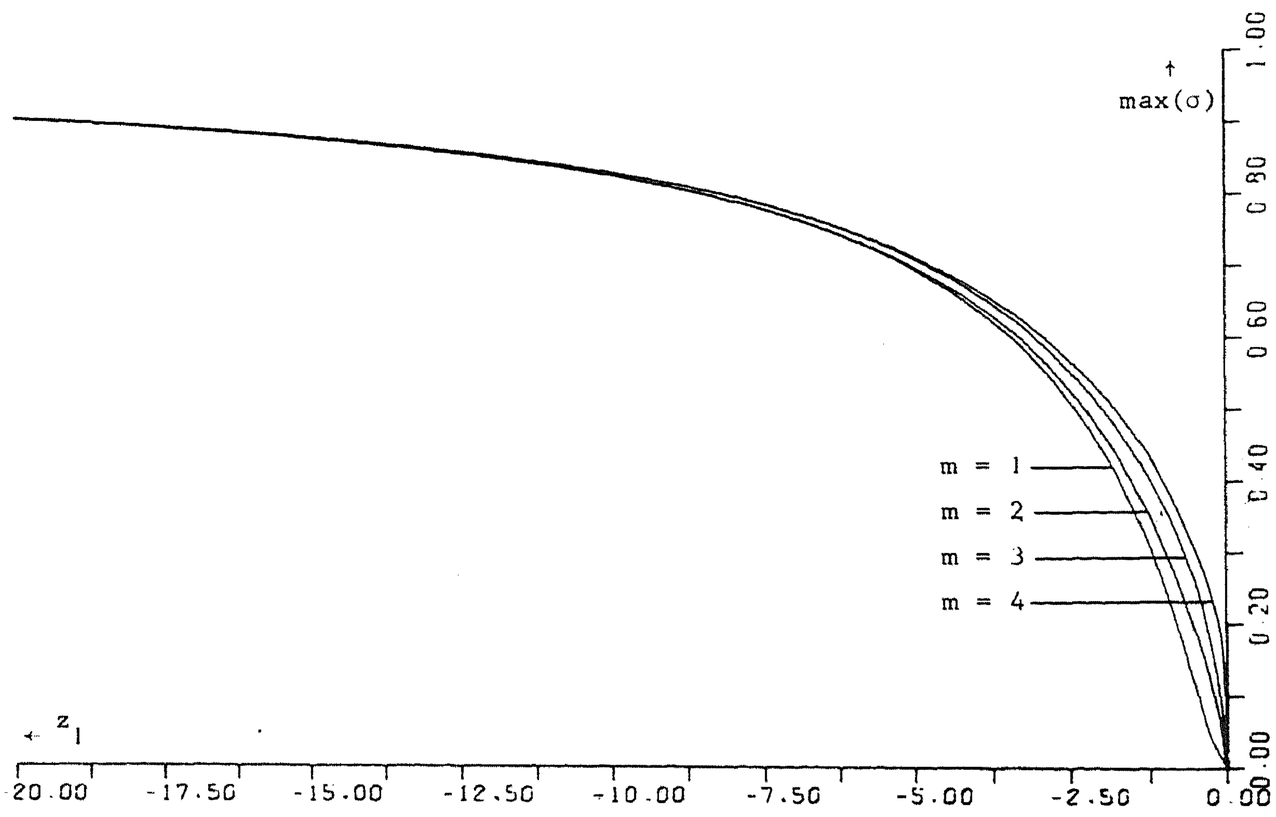


Fig. 3.1. Maximal σ-curves.

An illustrative example. Let the integer $N \geqq 1$, and denote $h = 1/(N+1)$. Suppose $k=2$ and let equation (3.1) originate from semi-discretization of $u_t = u_{x_1x_1} + u_{x_2x_2}$, defined on $(0, T] \times \{(x_1, x_2) \mid 0 < x_1, x_2 < 1\}$ with $u=0$ on the boundary and initial function $s(x_1, x_2)$. Assume that the semi-discretization has been performed on a uniform grid of size h with second order symmetrical finite differences. The LOD matrices J_1 and J_2 are then given by $J_1 = I \otimes A$, $J_2 = A \otimes I$, where I denotes the unit matrix of order N, A the standard finite difference matrix, and the symbol $\otimes$ direct product. These matrices satisfy the requirements of the test-model.

To get some insight in the convergence behaviour of the IDEC, we did some experiments for the special initial function

$$(3.6) \qquad s(x_1, x_2) = \frac{\sin \pi x_1 \sin \pi x_2}{(1 - 2\alpha \cos(\pi x_1) + \alpha^2)(1 - 2\alpha \cos(\pi x_2) + \alpha^2)}, \qquad \alpha^2 < 1 .$$

The Fourier coefficients of (3.6) are given by α^{i+j-2}, $i, j = 1, 2, \ldots$ (see [6]). Hence, if $\alpha \to 1$, we expect that the convergence slows down. We applied the IDEC for $m = 1(1)4$, each time on 4 subintervals $[0, H_1] = [0, m\tau]$ for $\alpha = 0, 0.1$ and 0.5. In table 3.1 we list the number of iterations necessary to satisfy $\|\tilde{\eta}^{j+1} - \tilde{\eta}^j\|_\infty < \theta$, θ being given in the table. In all experiments $N = 10$, i.e. $h = 1/11$. As the eigenvalues of both J_1 and J_2 are given by $-4h^{-2} \sin^2(j\pi h/2)$, $j = 1(1)N$, the z_1-values (see fig. 3.1) lie approximately between $-\pi^2\tau$ and -484τ. The number $-\pi^2$ approximates the smallest eigenvalue. If only the first harmonic is present in the initial function, i.e. $\alpha = 0$ in (3.6), the speed of convergence is completely determined by the product of τ and this smallest eigenvalue.

The results of the convergence experiment show that, despite the damping as pointed out in (3.5), the IDEC is rather *sensitive* with respect to the higher harmonics. The experiment also shows that mostly the speed of convergence decreases with the number of iterations (provided $\alpha \neq 0$). Because of these unwanted phenomena, it seems of less use to apply the IDEC while iterating until convergence. In the next section we therefore discuss some more experiments performed with a *fixed* number of iterations, viz. $m - 1$. In this approach we fully rely on the *order* of the formulas.

Table 3.1. *Results of convergence experiment.*

		$\tau = 1/10$			$\tau = 1/20$			$\tau = 1/40$			$\tau = 1/80$		
m	α \ θ	10^{-2}	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-6}
1	0.0	3	6	9	2	4	6	2	3	5	1	3	4
	0.1	3	7	14	2	5	10	2	4	7	2	3	5
	0.5	4	16	49	3	12	32	3	9	21	2	6	13
2	0.0	2	7	11	2	6	8	2	4	6	2	2	5
	0.1	2	7	14	2	6	10	2	5	8	2	3	6
	0.5	3	13	44	3	10	29	3	8	20	2	6	14
3	0.0	3	7	12	3	6	10	2	5	7	2	4	6
	0.1	3	7	13	3	6	10	3	5	8	2	5	8
	0.5	3	11	40	3	9	27	3	8	19	3	7	13
4	0.0	3	8	13	3	7	11	2	6	9	2	5	7
	0.1	3	8	13	3	7	12	2	6	11	2	5	9
	0.5	3	10	37	4	9	25	4	9	17	3	8	13

4. Numerical experiments.

4.1. *The examples used.*

We report numerical results for 3 examples of initial boundary value problems for 2-dimensional equations of type (1.1) chosen from two test families used in [5]. We first list the 2 families (in reduced form) and then the 3 examples. In all examples $(0, T] \times \Omega = (0, 1] \times \{(x_1, x_2) \mid 0 < x_1, x_2 < 1\}$. For simplicity, we confined ourselves to Dirichlet boundary conditions.

First family

$$\begin{cases} F_1(t, x_1, x_2, u, u_{x_1}, u_{x_1x_1}) = u^{2\nu}[u_{x_1x_1} + a(t, x_1, x_2)] + g(t, x_1, x_2) , \\ F_2(t, x_1, x_2, u, u_{x_2}, u_{x_2x_2}) = u^{2\nu} u_{x_2x_2} , \end{cases} \tag{4.1}$$

where

$$\begin{aligned} a(t, x_1, x_2) &= -2t^2(x_1 + \sin(2\pi t)) , \\ g(t, x_1, x_2) &= t[(x_1^2 + x_2)(2\sin(2\pi t) + 2\pi t \cos(2\pi t)) + 2x_1 x_2^2] , \end{aligned}$$

with solution $u(t, x_1, x_2) = 1 + t^2[(x_1^2 + x_2)\sin(2\pi t) + x_1 x_2^2]$.

Second family.

$$\begin{aligned} F_1(t, x_1, x_2, u, u_{x_1}, u_{x_1x_1}) &= \sqrt{(u)} u_{x_1x_1} - \frac{u}{2(1+t)} - 2u\sqrt{u} , \\ F_2(t, x_1, x_2, u, u_{x_2}, u_{x_2x_2}) &= \sqrt{(u)} u_{x_2x_2} , \end{aligned} \tag{4.2}$$

with solution $u(t, x_1, x_2) = \exp(-x_1 - x_2)/\sqrt{(1+t)}$.

EXAMPLE 1. The linear problem obtained from equations (4.1) with $\nu = 0$.
EXAMPLE 2. The non-linear problem obtained from equations (4.1) with $\nu = 1$.
EXAMPLE 3. The non-linear problem obtained from equations (4.2).

The problems were semi-discretized on a uniform grid, using second order symmetrical finite differences, with grid size $h = 1/(N+1)$, $N = 19$. The boundary expressions, appearing in the ordinary differential equations for the internal grid points next to the boundary, were evaluated at $t = t_{\nu+1}$ (see formula (1.5)). Note that the space errors for examples 1–2 are equal to zero.

4.2. *The algorithms used.*

As discussed in the previous section, it is of no use to perform a large number of IDEC iterations. Consequently, we applied the technique with a fixed number of $m-1$ iterations, so that the order of consistency of the resulting algorithm is equal to m. Observe that for $m = 1$ we thus applied the LOD method itself.

In case of non-linear problems the calculation of the $y_{(i)}$-vectors in (1.5) involves the solution of a system of non-linear equations. In actual applications it is of no

use to solve these systems very accurately as the LOD method is only of first order. Therefore, we performed one Newton-type iteration leading to the result

$$y_{(i)} = y_{(i-1)} + \tau(I - \tau\bar{J}_i)^{-1} f_i(t_{v+1}, y_{(i-1)}) , \tag{4.3}$$

where $\bar{J}_i$ approximates the partial derivative $\partial f_i/\partial y$ at (t_v, y_v). The resulting scheme remains first order consistent. The (tridiagonal) matrices $\bar{J}_1$ and $\bar{J}_2$ were computed by means of first order finite differences. This computation was (if necessary) performed only at the beginning of each IDEC step, hence at the times t_{lm}, $l = 0, 1, \ldots$.

To be able to compare the results of the various algorithms we need a measure, say ce_m, for the *computational effort per integration step* of length τ. It is convenient to express ce_m in the effort of the LOD-method. Therefore, we set $ce_1 = 1$. We now assume that the effort of one IDEC iteration, using m points, is equal to $2mce_1 = 2m$. This is justified by the observation that the defect calculations require the evaluation of a derivative and a weighted sum (2.5). Consequently, we have $ce_m = 2m - 1$. The computational labour involved in the calculation of the matrices $\bar{J}_1$ and $\bar{J}_2$ has been left out of consideration. For non-linear problems this favours the schemes where m is small, especially the LOD formula, because it integrates with matrices being updated every integration step. This will influence the stability (and possibly the accuracy) of the formula.

4.3. *The results.*

The 3 examples were integrated with all algorithms, i.e. with $m = 1(1)4$, for $\tau = 1/12, 1/24, 1/48, 1/96$. In the tables of results one finds, for two t-values, $ae = -\lg \varepsilon$ where ε is the maximum absolute error at t over the grid, and $sce_m = ce_m t/\tau$. In the tables the symbol $*$ means instability.

Table 4.1. (ae, sce_m)-*values for example* 1.

	$t = 0.5$				$t = 1$			
τ \ m	1	2	3	4	1	2	3	4
1/12	1.73, 6	2.13, 18	2.43, 30	2.73, 42	0.96, 12	1.36, 36	1.81, 60	2.07, 84
1/24	1.94, 12	2.51, 36	2.89, 60	3.12, 84	1.16, 24	1.76, 72	2.23, 120	2.46, 168
1/48	2.18, 24	2.87, 72	3.27, 120	3.49, 168	1.42, 48	2.15, 144	2.61, 240	2.84, 336
1/96	2.46, 48	3.21, 144	3.67, 240	3.92, 336	1.69, 96	2.51, 288	3.02, 480	3.28, 672

Table 4.2. (ae, sce_m)-*values for example* 2.

	$t=0.5$				$t=1$			
τ \ m	1	2	3	4	1	2	3	4
1/2	1.67, 6	0.83, 18	*	*	0.36, 12	*	*	*
1/24	1.83, 12	1.97, 36	*	*	0.99, 24	*	*	*
1/48	2.06, 24	2.57, 72	2.53, 120	1.60, 168	1.25, 48	*	*	*
1/96	2.34, 48	2.95, 144	3.37, 240	3.69, 336	1.49, 96	1.72, 288	*	*

Table 4.3. (ae, sce_m)-*values for example* 3.

	$t=0.5$				$t=1$			
τ \ m	1	2	3	4	1	2	3	4
1/12	1.67, 6	2.07, 18	2.39, 30	1.50, 42	1.76, 12	2.17, 36	2.50, 60	*
1/24	1.89, 12	2.39, 36	2.74, 60	2.97, 84	1.98, 24	2.48, 72	2.84, 120	3.06, 168
1/48	2.13, 24	2.71, 72	3.10, 120	3.34, 168	2.22, 48	2.81, 144	3.21, 240	3.45, 336
1/96	2.39, 48	3.05, 144	3.52, 240	3.78, 336	2.48, 96	3.15, 288	3.63, 480	3.89, 672

The results indicate that the following conclusions are justified:

1°. For non-linear problems the IDEC formulas are less stable than the LOD formula. This conclusion is justified by the results for example 2, and the result for example 3 obtained for $\tau = 1/12$ and $m=4$. We emphasize, however, that the LOD formula updates the Jacobian matrices every integration step. In practice this is very expensive and will seldom be done. Nevertheless, if the updating is not performed every step, it is still expected that the LOD formula is more stable.

2°. In case of stable computations the results become better with increasing m. This can be immediately verified by putting the (ae, sce_m)-values of examples 1, 3 in an accuracy-efficiency diagram. Between successive values of m the improvement is not large. If we compare the results obtained for $m=4$ with the results obtained by the LOD formula, however, the improvement is significant. Let us, for example, consider the results given in table 4.1 for $t=1$. Now, if we assume that further halving the stepsize in the LOD formula also halves the error—for τ small enough this is inevitable—we can write down the following (ae, sce_1)-values: (1.99, 192), (2.29, 384), (2.59, 768), (2.89, 1536), (3.19, 3072), etc. It is immediately seen that the corresponding results for $m=4$ are significantly better.

3°. The order of consistency of the IDEC formulas cannot be recovered from the results (note that in example 1 the space errors are equal to zero; further, from the experiments described below it can be seen that in the errors of table 4.3 the

time integration errors dominate clearly). To indicate that this phenomenon is probably not due to the defect correction, but *inherent in the collocation schemes*, we performed two further experiments. We integrated examples 1, 3 with the formula for $m=4$, using $\tau=1/24$, $1/48$ and $1/96$, but now performing 10 IDEC iterations in order to obtain a numerical approximation closer to the collocation approximation. The *ae*-values obtained at $t=1$ are given below:

τ	1/24	1/48	1/96
example 1	3.18	3.67	4.33
example 3	3.95	4.39	5.04

All errors are significantly smaller than the corresponding errors of the preceding experiments. Again, however, the order $p=4$ can not be recovered (this will be the case after an unacceptable decrease of τ). As in all computations the inequality $\|\eta_\nu^{10}-\eta_\nu^9\|_\infty<10^{-ae}$, $\nu=1(1)1/\tau$, was satisfied, we believe that the *effective order* of the collocation schemes themselves — when applied to semi-discrete parabolic equations with realistic stepsizes — is significantly smaller than the *theoretical order*. The reader should observe that this conclusion is in disagreement with the results reported in [2], section 6.

5. Concluding remarks.

The IDEC formulas, especially the ones of higher order (provided they remain stable when integrating non-linear problems) are more efficient than the basic LOD formula. A disappointment is that the effective order of the formulas is significantly smaller than the theoretical order when considering realistic stepsizes. This may imply that the additional computational effort needed to obtain the higher theoretical orders is better used when integrating (using relatively small stepsizes) with a simple *second* order splitting method, such as the method of alternating directions or the line hopscotch method [4]. These methods also possess unconditional stability properties and are computationally attractive (per integration step). Some results of the line hopscotch method, applied to examples 1, 3, are given in appendix 3 of [6]. It appears that for example 1 our formula for $m=4$ is slightly better, whereas for the non-linear example 3 the line hopscotch method is to be preferred. We did not investigate the application of defect correction to line hopscotch, or ADI, as these methods do not yield equations of type (3.3′) (cf. [4]). Because their stability functions do not vanish at infinity it is doubtful whether a modified IDEC process would lead to unconditionally stable schemes.

Acknowledgements.

The author wishes to thank Mrs. M. Louter-Nool for her programming assistance and Dr. C. den Heyer for carefully reading the manuscript.

REFERENCES

1. M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions*, National Bureau of Standards Applied Mathematics Series 55, U.S. Government Printing Office, Washington, 1964.
2. R. Frank and C. W. Ueberhuber, *Iterated defect correction for the efficient solution of stiff systems of ordinary differential equations*, Report No. 17/76, Institute for Numerical Analysis, Technical University, Vienna, 1976 (in a condensed form published in BIT 17 (1977), 146–159).
3. H. J. Stetter, *The defect correction principle and discretization methods*, Num. Math. 29 (1978), 425–443.
4. P. J. van der Houwen and J. G. Verwer, *One-step splitting methods formulated for semi-discrete parabolic equations*, Report NW 55/78, Mathematisch Centrum, Amsterdam (prepublication) 1978 (to appear in Computing).
5. P. J. van der Houwen, B. P. Sommeijer and J. G. Verwer, *Comparing time-integrators for parabolic equations in two space dimensions with a mixed derivative*, J. of Comp. and Applied Math. 5, No. 2, 1979.
6. J. G. Verwer, *The application of iterated defect correction to the LOD method for parabolic equations*, Report NW 58/78, Mathematisch Centrum, Amsterdam, 1978.
7. N. N. Yanenko, *The method of fractional steps*, Springer-Verlag, Berlin, 1971.

MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
1091 AL AMSTERDAM
THE NETHERLANDS