ELSEVIER

Journal of Electroanalytical Chemistry 406 (1996) 15-21

JOURNAL OF ELECTROANALYTICAL CHEMISTRY

Brute force digital simulation

Dieter Britz

Kemisk Institut, Århus Universitet, 8000 Århus C, Denmark

Received 21 June 1995; in revised form 2 November 1995

Abstract

The discretised forms of the partial differential electrochemical diffusion–reaction equations can be solved at each time step by simply leaving them as systems of linear equations, and solving by LU decomposition. Under favourable conditions, this can have efficiency comparable with more sophisticated methods of solution and in any case does not often lead to unacceptable execution times on modern computers. This is demonstrated by the examples of a simple first-order catalytic mechanism, a more complex second-order mechanism, and the two-dimensional simulation of a potential step at a microdisk electrode.

Keywords: Brute force method; Digital simulation; Electrochemical diffusion–reaction

1. Introduction

In electrochemical digital simulations [1,2], if reasonable efficiency is desired, an implicit algorithm is indicated. There are a variety of these, such as the Crank–Nicolson (CN) [3], the Laasonen (or 'fully implicit') [4], and the recently rediscovered Richtmyer scheme [5–7], which is also 'fully implicit' but uses a higher-order time derivative to improve the accuracy. One reason for using implicit schemes is that they are generally stable for a large range of interval sizes in both time and space, and this allows the use of unequal intervals [2,8], which greatly reduce the number of required sample points in space.

For some time, however, a problem remained to be solved. Systems that require digital simulation usually involve homogeneous reactions, and there is often coupling between species. That is, of the set of diffusion–reaction equations for the various species, some contain terms in more than one species' concentrations. In such cases, discretisation of the partial differential equations, using an implicit scheme, will lead to systems of equations that are not amenable to the simple Thomas algorithm for an implicit scheme applied to a system with a single species, which yields a tridiagonal system of linear equations [2]. With more than one species, even with judicious arrangement of the order of the species' concentration in the equation, one is faced with penta-, hepta-diagonal systems (or worse). Until the paper by Rudolph [9], electrochemists tended to avoid implicit methods for such multi-species reactions for this reason. Rudolph showed [9] how, by couching the discretised equations in terms of concentration vectors and coefficient matrices and vectors, the problem could be reduced once again to the simple tridiagonal form, solvable by an extension of the Thomas algorithm. In fact, his method has a prehistory [10–17] going back (at least) to 1968 [10] but had been forgotten by electrochemists. Rudolph [9] originally used the Laasonen scheme, and later switched to the more accurate modified Richtmyer scheme [18]. A CN adaptation of the method has been described [19].

The assumption that the above-mentioned pentadiagonal, etc. equation systems are awkward to solve is based on the idea that this would require too much computing time. Thus Rudolph [20] writes "...the method is prohibitively expensive... may require hundreds or thousands of concentration points...". It is true, as will be seen below, that multi-species reaction schemes do lead to rather large matrices, but modern computers have large memories and very fast execution. This, together with some efficient ways of solving matrix equations and some fortuitous circumstances, make this brute force approach feasible, as will be shown, and perhaps easier to program. The method will be explained in detail below by means of some examples.

SSDI 0022-0728(95)04439-6

As often, the method to be described is not new. It was used recently by Balslev and Britz [21] to directly solve for the steady state of a complex mechanism at a rotating disk electrode, by Mao and White [22] for general steady state problems in two dimensions, and by Compton et al. [23], again for a steady state, at a channel electrode. In the examples that follow, it will be applied to time-marching problems.

It is worth noting that when examining a particular method, this does not exclude alternatives. It has, for example, been pointed out [24,25] that in large all-purpose simulation packages, there should preferably be a wide choice of methods available. This is realised in the electrochemical digital simulation package ELSIM [26], where the user chooses the preferred scheme.

2. Examples

In the following two examples, the notation is as follows. T denotes time t normalised by an observation time τ, i.e. $T = t/\tau$. Distance x from the electrode is normalised to X by dividing by $\delta = \surd 2(D\tau)$. Concentrations c_S (species S) are normalised by the concentration of the starting species A, i.e. $C_S = c_S/c_A$. The Feldberg exponentially expanding grid [2,8] is used to achieve unequal intervals, by the transformation

$$Y = \ln(1 + aX) \tag{1}$$

where Y is the transformed space variable and a the adjustable parameter. A typical value for a in the simulations was 3.

2.1. Simple catalytic system

This is expressed by the reaction pair

$$\mathrm{A} + \mathrm{e}^- \longrightarrow \mathrm{B}$$
$$\mathrm{B} \longrightarrow \mathrm{A} \tag{2}$$

We assume a potential jump to a very negative potential, so that C_A at the electrode is forced to zero, and that the catalytic chemical reaction is irreversible, with rate constant k, normalised to $K = k\tau$. Using the unequal intervals transformation, this leads to the pair of partial differential equations

$$\frac{\partial C_A}{\partial T} = a^2\exp(-2Y)\left(\frac{\partial^2 C_A}{\partial Y^2} - \frac{\partial C_A}{\partial Y}\right) + KC_B$$
$$\frac{\partial C_B}{\partial T} = a^2\exp(-2Y)\left(\frac{\partial^2 C_B}{\partial Y^2} - \frac{\partial C_B}{\partial Y}\right) - KC_B \tag{3}$$

with

Assuming that there are m sample $(Y-)$ points in space, the above equations are discretised to

$$C'_{A,i-1} + a_1(i)C'_{A,i} + a_k(i)C'_{B,i} + a_2C'_{A,i+1} = b_{A,i}$$
$$C'_{B,i-1} + [a_1(i) - a_k(i)]C'_{B,i} + a_2C'_{B,i+1} = b_{B,i} \tag{4}$$

for $1 \le i \le m$, where the primes on C denote new concentrations, to be calculated at $T + \delta T$. The a coefficients depend on the discretisation scheme used and have been defined, for example, for CN in Ref. [19] for this mechanism. The b terms on the right-hand sides (also defined in Ref. [19]) contain known concentrations and a further a coefficient. The details of these are not of interest here, nor indeed a discussion of the particular implicit scheme used. The approach described by Rudolph [9,20] consists of the combination of these two equations into a single one,

$$C'_{i-1} + A_1(i)C'_i + A_2(i)C'_{i+1} = B_i \tag{5}$$

in which the C' now are vectors, A_1 a coefficient matrix, and A_2 and B_i vectors, all obvious. There are m such equations and the system is solved by the usual Thomas algorithm, now operating on matrices and vectors instead of scalars.

In this work, this step is not taken; instead, the two Eqs. (4) are simply placed into a large matrix equation, together with the discretised boundary conditions. We retain Rudolph's ordering of the C' terms; that is, the unknowns vector C' is $[C'_{A,0}\ C'_{B,0}\ C'_{A,1}\ C'_{B,1} \ldots C'_{A,m}\ C'_{B,m}]^T$. For the present example, the boundary conditions are

$$C'_{A,0} = 0$$
$$f_A + f_B = 0 \tag{6}$$

with f_S the flux of species S at the electrode. Assuming equal diffusion coefficients, the flux condition is equivalent to

$$\sum_{i=0}^{n-1} \beta_i C'_{A,i} + \sum_{i=0}^{n-1} \beta_i C'_{B,i} = 0 \tag{7}$$

using the n-point current approximation [2,27] and lumping the coefficients into β for convenience. Extension to species-dependent diffusion coefficients or the use of the two-point flux approximation are obvious. We do not simplify the flux discretisation by making use of the first boundary condition here, because, normally, the first boundary condition is not as simple as it is here (more commonly, it expresses Nernstian equilibrium, or the Butler–Volmer kinetic equation). These two boundary discretisations, and all m equation pairs (4) are now placed into the large matrix equation

$$AC' = B \tag{8}$$

with

$$A = \begin{bmatrix}
1 & 0 & \dots & & & & & & & \\
\beta_0 & \beta_0 & \beta_1 & \beta_1 & \dots & \dots & \beta_{n-1} & \beta_{n-1} & & \\
1 & 0 & a_1(1) & a_k(1) & a_2 & 0 & & & & \\
0 & 1 & 0 & a_1(1)-a_k(1) & 0 & a_2 & & & & \\
 & & 1 & 0 & a_1(2) & a_k(2) & a_2 & 0 & & \\
 & & 0 & 1 & 0 & a_1(2)-a_k(2) & 0 & a_2 & & \\
 & & & & \dots & \dots & & & & \\
 & & & & & \dots & \dots & & & \\
 & & & & & & 1 & 0 & a_1(m) & a_k(m) \\
 & & & & & & 0 & 1 & 0 & a_1(m)-a_k(m)
\end{bmatrix}$$

C' is already defined and $B = [0\ 0\ b_{A,1}\ b_{B,1} \cdots b_{A,m}\ b_{B,m}]^T$. The matrix Eq. (8) is solved directly (see below). Since unequal intervals are used, m typically lies between 15 and 50. With strong unequal stretching, such as used by the commercial program DigiSim 2.0 [18], one finds a very economical m value, typically 13. Let m be 50; this results here in 102 unknowns or a 102 × 102 matrix. The matrix is strongly banded (it is pentadiagonal except for the second row which has $2n$ terms).

A remark on the boundary conditions: with the Rudolph method, using the Thomas algorithm and using the n-point current approximation, these become rather messy, as described [19] (although slightly more elegant expressions are possible and will be described elsewhere). Here, as seen in the definition of the matrix A above, they enter, in a straightforward manner, into the first so many rows of the matrix.

2.2. *Three-species simulation of a second-order mechanism*

Pedersen and Svensmark [28] described a complex mechanism, that can be simplified to the form

$$\begin{aligned} &A + e^- \rightleftharpoons B \\ &2B + C \longrightarrow A + D \end{aligned} \qquad (9)$$

This, in contrast with the previous example which has only a first-order homogeneous reaction, has a second-order reaction. The first, electrochemical, step is assumed to be reversible, and the second irreversible. Species D is of no interest and need not be simulated, so this involves three species to be simulated. The discretisation of the ensuing partial differential equation set has been described in a previous paper [19] and will not be repeated here. These discrete equations, together with those for the boundary conditions (also described in Ref. [19]) are then, as with the example in 2.1, placed in a large matrix equation. With the three species involved here and assuming $m = 50$, this leads to a 153 × 153 matrix, still feasible. The actual experiment simulated was, again, a potential jump. For interest we here applied, as well as the CN scheme, the Richtmyer scheme described by Mocak and Feldberg [6], using three T-levels.

2.3. *Ultramicroelectrode (a two-dimensional example)*

The ultramicrodisk electrode, embedded in an insulating plane, was first simulated by Heinze [29], who used the alternating directions implicit/explicit (ADI) method (see Lapidus and Pinder [30]) to simulate a Cottrell experiment with this electrode. Later, some theory was developed for this electrode (see Aoki [31]), as well as more efficient simulation methods. Michael and Wightman [32] described a conformal map to transform the somewhat awkward coordinate system of the microelectrode into a simple rectangular one. Verbrugge and Baker [33] modified the transformation slightly, resulting in a closed rectangular transformed space extending to real-space infinity. They found that they obtained reasonable results with a grid divided into just 30 × 30 intervals. They used the explicit method, while Michael and Wightman [32] used hopscotch [34]. Neither of these methods is particularly good and indeed hopscotch has been shown by Feldberg [35] to have a major flaw, rendering it practically useless. A better method, such as an implicit scheme, would be preferable here. The application of straight CN, for example, even for a single species, results in a banded system of equations, which is the reason for using ADI. This system is, thus, an obvious candidate for the present brute force approach. First, a brief description of the notation and transformation used by Verbrugge and Baker: the coordinate system is (r,y), with r the radial distance in the plane of the disk from the disk centre, and y the height above that plane. These are normalised by the disk radius r_0, giving the dimensionless $X = r/r_0$ and $Y = y/r_0$ (the disk edge is now at $X = 1$). These are further transformed into (θ, Γ) space by

$$\begin{aligned} X &= \cos(\theta)\cosh\left(\frac{\Gamma}{1-\Gamma}\right) \\ Y &= \sin(\theta)\sinh\left(\frac{\Gamma}{1-\Gamma}\right) \end{aligned} \qquad (9)$$

so that we now have the finite rectangle with the disk surface in the range $0 \le \theta \le \pi/2$ (respectively the disk edge and centre), and the radial distance from the disk centre out into the solution, $0 \le \Gamma \le 1$, with $\Gamma = 1$ being at infinity. Here time t is normalised as $T = tD/r_0^2$, following Verbrugge and Baker. In this simple coordinate system, the diffusion equation for a single species is

$$\frac{\partial C}{\partial T} = \frac{1}{\sin^2\theta + \sinh^2 \dfrac{\Gamma}{1-\Gamma}} \times \left\{ \frac{\partial^2 C}{\partial \theta^2} - \tan\theta \frac{\partial C}{\partial \theta} + (1-\Gamma)^4 \frac{\partial^2 C}{\partial \Gamma^2} + \left[(1-\Gamma)^2 \tanh \frac{\Gamma}{1-\Gamma} - 2(1-\Gamma)^3 \right] \frac{\partial C}{\partial \Gamma} \right\} \quad (10)$$

and concentrations are sampled at an evenly spaced number of points, in $n_\theta + 1$ columns along θ and $n_\Gamma + 1$ rows along Γ (both including zero). Upon discretising Eq. (10), this yields $(n_\theta + 1)(n_\Gamma - 1)$ discrete equations (at $\Gamma = 0$ and 1, the rows are fixed-value boundary concentrations), each one with five unknowns, two of them at some distance along the row from the central cluster of three. This is why previous efforts have used either explicit methods or ADI. In the present approach, the grid of points was mapped into one long vector in an obvious way, effectively assembling rows of points end on end. Thus, if the coordinates order is (Γ_i, θ_j), the vector is $[C(1,0), C(1,1), \ldots C(1,n_\theta), C(2,0), \ldots, \ldots C(n_\Gamma - 1, n_\theta)]^T$. Note that the row for $\Gamma = 0$ is that lying along the electrode and, for this experiment, all values there are zero (one of the boundary conditions), and that the row for $\Gamma = 1$ lies at infinity and is another boundary (bulk concentrations). In the θ-direction, the remaining boundary conditions are zero flux at both ends, that is $\partial C/\partial \theta = 0$ or no diffusion in the θ-direction. The ordering of the concentration vector then determines the position of the matrix coefficients, and we have a rather large matrix. Verbrugge and Baker found [33] that a 31×31 grid was sufficient, and in the present work, using implicit CN as well as Richtmyer discretisation, this was confirmed. This nevertheless results in a roughly 900×900 matrix, which is not so small but it can be handled. It is not possible here, no matter how the concentrations are ordered, to ensure a tightly banded matrix.

The CN scheme was applied, with ADI for a comparison. In order to get some feeling for the grid spacing requirements and the current integration over the disk, some preliminary steady state calculations were carried out here. As with the rotating disk simulations [21], this was done by setting the left-hand side of Eq. (10) to zero, and directly solving the discretised set of equations. This yielded the current ratio I (ratio of the total current integrated over the disk surface to that at very long times), and it was very close to unity, as it should be.

3. Computations

The computations were carried out on a Digital Equipment Alpha 1000 4/200 (a RISC machine) running under OSF/1; programming was in Fortran and double precision was used throughout (but see below).

4. Results and discussion

4.1. Simple catalytic mechanism

For this reaction, there is a very efficient method, similar to the Thomas algorithm for a single species. Briefly, it consists of the backward reduction of the second of Eqs. (4) to two unknowns per line, making use, as usual [19] of the known value $C'_{\mathrm{B,m+1}}$. Having 'arrived' at the electrode, these new equations for species B can now generate coefficients $u_{\mathrm{B},i}$ and $v_{\mathrm{B},i}$ recursively, such that all $C'_{\mathrm{B},i}$ can be expressed as a linear expression in $C'_{\mathrm{B},0}$, that is $u_{\mathrm{B},i} + v_{\mathrm{B},i} \times C'_{\mathrm{B},0}$. Now the first of the two Eqs. (4) can be reduced in a similar backward scan to two unknowns for species A. These equations all also involve a term in C'_{B} and the linear expression in $C'_{\mathrm{B},0}$ is used instead. This results in a new equation with an extra coefficient for $C'_{\mathrm{B},0}$, generated recursively along with the others. Again 'arriving' at the electrode, there is now enough information, together with the boundary conditions, to solve for the vector C' in the usual way [19]. This is the most efficient solution for this system (using an implicit scheme) and is called ICN here (as·CN was used). One might expect that the next most efficient is the Rudolph method, followed by the present brute force method.

In this case, however, it is possible to make use of the fact that the matrix A (see its definition, Eq. (8)) is constant throughout. The most efficient way of solving the matrix Eq. (8) is by LU decomposition [36]. Moreover, it is possible here to LU-decompose matrix A at the start of the simulation, so that at every iteration, all that is needed is to give values to the B-vector elements and to do the LU back-substitution. This fortunate circumstance will be met in all mechanisms involving only first-order homogeneous reactions. In the present work, the efficient LUD routines, DEC/SOL, were used [37], which turned out to be about twice as fast as the routines provided in Press et al. [36]. The programs were run with a variable number of time intervals n_T in the range $10 \le n_T \le 10\ 000$, and the cpu time was measured. All three algorithms produced identical results. The unequal intervals stretching parameter was set at 3, λ_0 at 3 ($\lambda_0 = \delta T/H_0^2$, with H_0 the position of the first sample point away from the electrode). The dimensionless homogeneous rate constant K was set at 1. Fig. 1 shows the execution times against n_T. It is not surprising that the Thomas-like algorithm ICN is the fastest for all n_T; what might surprise is that for some lower n_T, the brute force approach is actually faster than that of Rudolph.

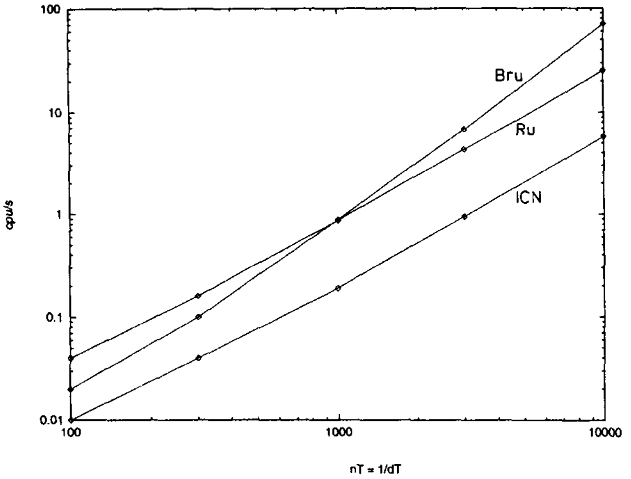

Fig. 1. Cpu time vs. n_T for the simple catalytic mechanism, for some methods (Ru denotes Rudolph; Bru is the present brute force approach; ICN is Crank–Nicolson with implicit boundary conditions).

This is no doubt due to the precalculation of the LU-decomposition of matrix A. As that matrix becomes larger with n_T, the matrix operations required with the brute force method begin to take more and more time and Rudolph becomes faster. The cross-over occurs at $n_T \approx$ 1000, and if this were a cyclic voltammetric simulation, this would be within the normal range of the total number of points. One might conclude, then, that the brute force approach is competitive in terms of cpu time with the Rudolph method for first-order homogeneous reactions. As well, programming for brute force is somewhat easier, especially in the expressions for the boundary conditions, although this is subjective. Since execution times were trivial (at the crossover point, both methods took about 1 s), ease of programming might be a bigger factor. The present author finds that it is easier to handle boundary conditions with the brute force approach; they simply go into the first few rows of the large matrix.

4.2. Three-species simulation of a second-order mechanism

With this mechanism, Eq. (9), there is a second-order homogeneous reaction and this results in matrix A varying with time (iteration steps) since it contains concentration terms [19]. Thus, it is necessary to carry out the complete LUD process at each iteration. The same simulation parameters were used as in the example in 2.1, again giving the homogeneous reaction rate a (dimensionless) value of 1. Again, as expected, the Rudolph and brute force methods gave identical results, and Fig. 2 shows the execution times vs. n_T. They are greater than in the example in 2.1, and here the brute force method requires more cpu time than Rudolph for all n_T. The advantage of somewhat easier programming and boundary condition handling remains, however. At $n_T = 1000$, the brute force method took 12.6 s cpu time and this is regarded as tolerable, and no great inconvenience compared with the 3.2 s taken by the Rudolph method.

The brute force approach, but using the three-level Richtmyer algorithm, was also tried out, just for interest (although the five-level variant is normally advocated [6,7], three levels are in fact consistent in discretisation error order with the spatial discretisation of the second-order term in the diffusion equation). It produced practically identical results (at $T = 1$, calculated current differed from those using CN only in the fourth decimal), and the cpu time taken was almost the same as for the CN algorithm (just a shade higher).

4.3. Two-dimensional, microelectrode

This example was chosen in order to explore the utility of the brute force method for a two-dimensional problem, retaining the accuracy of an implicit approach. At every iteration, the whole two-dimensional grid of points needs to be recalculated; with the brute force approach, this grid must be mapped into a linear array. This was implemented by use of an index mapping function which for any (i,j)th element in the grid, then pointed to both row and column in the large matrix and the row in the coefficient $(B-)$ vector. Here again, the matrix was constant with time and could be precalculated and LU-decomposed.

There is no analytical solution for the time dependence of the current ratio I, and the best one can do is to compare methods and previous results such as those of Verbrugge and Baker [33]. In the present work, the ADI method, as used by Heinze [29] was used as a comparison, and a steady state calculation was programmed as a check on the current at large T. This gave the result $I = 0.9998$, in satisfactory agreement with the expected unity. ADI and the present algorithms produced the same result; both produced a current at $T = 1$ ($I = 1.349$) a little higher than that shown in Fig. 3 of Verbrugge and Baker [33] (about

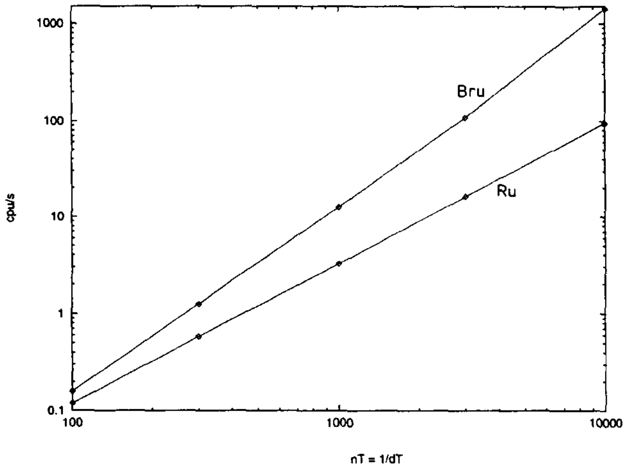

Fig. 2. Cpu time vs. n_T for the second-order PS mechanism, for the Rudolph and present methods, as marked.

1.2; they appear to have used the explicit scheme). Fig. 3 shows the cpu times for the two methods against grid size. Clearly, the present method takes considerably more time than ADI. As with the one-dimensional examples, however, if the system to be simulated included coupling of several species, then ADI would no longer be a reasonable option and the present approach might be a choice. On a fast computer, as was used in this study, such a simulation might take 1 h, which may not be too much. It must be said that around 10^6 numbers need to be stored for the present method (for example, the 31×31-point grid leads to a 899×899 matrix), and it is not certain that all personal computers are able to accommodate this.

One might ask why double precision is required here; this is not normally necessary except when testing for accuracy of a particular technique, and the use of single precision (REAL) might reduce execution time. Accordingly, the microdisk program was converted to REAL. For a particular set of simulation parameters, the execution time reduced to 767 s from 1046 s for the double precision version. Even though there were 10 000 iterations on a 31×31 grid (899×899 matrix), the final I value deviated from the double precision one only in the fourth decimal (1.350 instead of 1.349 at $T = 1$). Single precision, then, is a valid option and will cut down execution times to some extent.

Another variant might be to use matrix inversion instead of LU decomposition. Matrix A (in Eq. (8)) can be inverted once and for all, and the inverse multiplied at every iteration with vector B to directly yield the new concentration vector. This was tried as well, but turned out to require about double the execution time of the LUD approach (2218 s, compared with 1046 s).

One point to note here is that both ADI and 'brute force' (using CN) appeared to show that there may be unstable regimes. Fig. 4, for example, shows the present method applied to the Cottrell jump, using a 31×31 grid, with δT equal to 0.001 and 0.004. For the larger δT value, there are current oscillations, while for $\delta T < 0.001$ there are none. Similar behaviour is seen with ADI, usually considered unconditionally stable [30]. Much greater problems were encountered using the plain explicit method (apparently used by Verbrugge and Baker [33]), where, for the same grid, a δT as small as 10^{-6} had to be used to ensure stability. This makes some sense, if one denormalises the units used in the transformation, for the point in the disk plane, just outside the disk edge. For this point (see Eq. (9)), where (with a 31×31 grid), $\Gamma = 1/30$ and therefore (with $\theta = 0$), the distance δX from the disk edge is 0.6×10^{-3} which gives a value of about 3 for the parameter $\delta T / \delta X^2$. It is in fact surprising that the simulation was still stable; this probably has to do with the fact that it is around this point that intervals are smallest (largest $\delta T / \delta X^2$); regions around the other, more widely spaced points may have a stabilising effect. It is not clear how Verbrugge and Baker [33] were able to use much larger δT values. This stability problem warrants further study. No such instability signs were found, incidentally, for the modified three-level Richtmyer scheme.

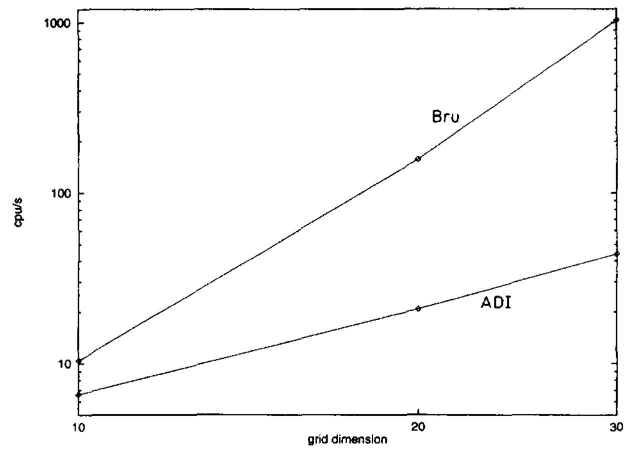

Fig. 3. Cpu time vs. grid dimension (log-scale, square grid, $n_\Gamma = n_\theta$) for the two-dimensional microdisk simulation, using ADI and the present method (as marked).

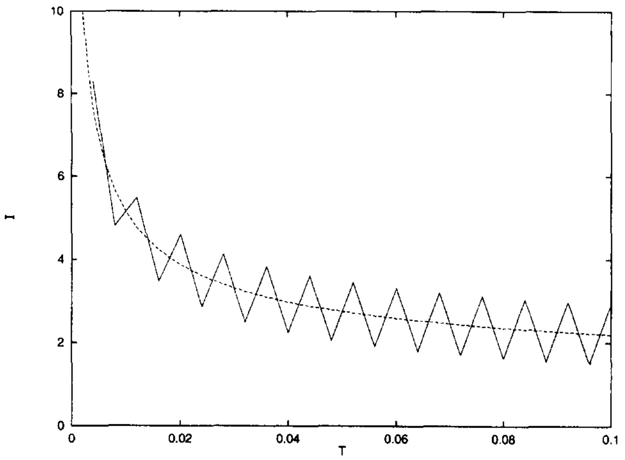

Fig. 4. Cottrell response at the microdisk using the present method, with the two δT values: 0.001 (smooth response) and 0.004 (oscillatory).

A quantity of interest is the log–log slope of the cpu time against some simulation parameters. In the case of the simple one-dimensional systems mentioned above, the Rudolph method, as well as ICN, shows the cpu time to be of order about 1.5 with respect to the number of iterations for the whole simulation, while the present method has about order 1.9 or close to 2.

In the microdisk system, we find for ADI an order 0.8 with respect to the number of grid points, and an order of 2 for the present method. This is quite serious and restricts computations to the roughly 30×30 grids used here. Fortunately, because of the strong nonlinearity in Cartesian space of the transformed grid, this appears to be sufficient, as also found by Verbrugge and Baker [33].

5. Remark

This has not been an exhaustive study of the possibilities of the present method. On the one hand, as noted above, for two-dimensional simulations, it would not be practical to use it for grids with many more points that used here (about 1000). For example, Compton et al. [23] describe a two-dimensional steady state problem with about 10^6 points, quite infeasible here. On the other hand, in all applications described in this paper, the non-zero parts of the large matrix are confined, even in the two-dimensional case, to a relatively narrow band and there exist methods to solve such systems much more efficiently than our present simple-minded LUD approach. In the case of the microdisk, discretised by a 31×31 grid of points, the band width is about 31, so in fact if a solver looked at (and stored) only the approximately 31×1000 non-zero matrix elements instead of all (about) 10^6, an efficiency gain by a factor of 30 or so could be achieved. This was not attempted at all in the present study.

6. Conclusions

For one-dimensional diffusion simulations, the present, brute force, method can be competitive with previous methods such as the Rudolph method, using what might be regarded as normal simulation parameter values. From a programming point of view the brute force method is more convenient, especially for the boundary condition expressions, although this impression may vary from one person to another. For the two-dimensional system studied here, programming was still reasonably easy, but here the rather high order of cpu time requirement with respect to the number of grid points, for the brute force method, is a clear drawback. However, if a fast computer is used, as here, cpu times of the order of 1 h, as expected for more complex systems, might be acceptable given the problems with more conventional methods such as ADI. Also, it appears that single precision is satisfactory, allowing some cpu time reduction, and further improvements appear possible.

Acknowledgement

The author gratefully acknowledges the many stimulating conversations he has had, both in person and by electronic mail, with Dr. L. Bieniasz, of Cracow, Poland.

References

[1] S.W. Feldberg, in A.J. Bard (Ed.), Electroanal. Chem., 3 (1969) 199.
[2] D. Britz, Digital Simulation in Electrochemistry, Springer, 1988.
[3] J. Crank and P. Nicolson, Proc. Cambridge Philos. Soc., 43 (1947) 50.
[4] P. Laasonen, Acta Math., 81 (1949) 309.
[5] R.D. Richtmyer and K.W. Morton, Difference Methods for Initial-Value Problems, Wiley, New York, 1967.
[6] J. Mocak and S.W. Feldberg, J. Electroanal. Chem., 378 (1994) 31.
[7] S.W. Feldberg and C. Goldstein, J. Electroanal. Chem., 397 (1995) 1.
[8] S.W. Feldberg, J. Electroanal. Chem., 127 (1987) 1.
[9] M. Rudolph, J. Electroanal. Chem., 314 (1991) 13.
[10] J. Newman, Ind. Eng. Chem. Fundam., 7 (1968) 514.
[11] J.S. Newman, Electrochemical Systems, Prentice-Hall, NJ, 1973, Appendix C.
[12] P.F. Greenfield, Simulation, 22 (1974) 152.
[13] A.C. Hindmarsh, Solution of block-tridiagonal systems of linear algebraic equations, Rep. UCID-30150, LLL, 1977.
[14] R.E. White, Ind. Eng. Chem. Fundam., 17 (1978) 367.
[15] R. Calinon and J. Ligou, Comput. Phys. Commun., 17 (1979) 317.
[16] F. Marsh and D.E. Potter, Comput. Phys. Commun., 24 (1981) 185.
[17] F. Dupoirieux and D. Scherrer, Numerical simulation of combustion phenomena, in R. Glowinski, B. Lavrouturon and R. Temam (Eds.), Lecture Notes in Physics, Vol. 241, Springer, Heidelberg, 1985, p. 32.
[18] M. Rudolph, D.P. Reddy and S.W. Feldberg, Anal. Chem., 66 (1994) 589A.
[19] D. Britz, J. Electroanal. Chem., 352 (1993) 17.
[20] M. Rudolph, in I. Rubinstein (Ed.), Physical Electrochemistry, Marcel Dekker, New York, 1995, p. 81.
[21] H. Balslev and D. Britz, Acta Chem. Scand., 46 (1992) 949.
[22] Z. Mao and R.E. White, J. Electrochem. Soc., 141 (1994) 151.
[23] R.G. Compton, R.A.W. Dryfe, R.G. Wellington and J. Hirst, J. Electroanal. Chem., 383 (1995) 13.
[24] B. Engquist and T. Smedsaas, in B. Engquist and T. Smedsaas (Eds.), PDE Software: Modules, Interfaces and Systems, North-Holland, Amsterdam, 1984, p. 399.
[25] A.D. Kowalski and R.L. Peskin, Math. Comput. Simul., 31 (1989) 343.
[26] L.K. Bieniasz, Comput. Chem., 17 (1993) 355.
[27] D. Britz, Anal. Chim. Acta, 193 (1987) 277.
[28] S.U. Pedersen and B. Svensmark, Acta Chem. Scand. A, 40 (1986) 607.
[29] J. Heinze, J. Electroanal. Chem., 124 (1981) 73.
[30] L. Lapidus and G.F. Pinder, Numerical Solution of Partial Differential Equations in Science and Engineering, Wiley, New York, 1982.
[31] K. Aoki, Electroanalysis, 5 (1993) 627.
[32] A.C. Michael and R.M. Wightman, J. Electroanal. Chem., 267 (1989) 33.
[33] M.W. Verbrugge and D.R. Baker, J. Phys. Chem., 96 (1992) 4572.
[34] D. Shoup and A. Szabo, J. Electroanal. Chem., 160 (1984) 1.
[35] S.W. Feldberg, J. Electroanal. Chem., 222 (1987) 101.
[36] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, Numerical Recipes in Fortran, Cambridge University Press, Cambridge, 1986, 2nd edn.
[37] A.C. Hindmarsh, L.J. Sloan and P.F. Dubois, DEC/SOL: solution of dense systems of linear algebraic equations, Rep. UCID-30137, LLL, 1978.