ELSEVIER

Journal of Electroanalytical Chemistry 456 (1998) 13–23

JOURNAL OF ELECTROANALYTICAL CHEMISTRY

An exponentially expanding mesh ideally suited to the fast and efficient simulation of diffusion processes at microdisc electrodes. 2. Application to chronoamperometry

D.J. Gavaghan [a,b,*]

[a] *Oxford University Computing Laboratory, Wolfson Building, Parks Rd., Oxford,UK*
[b] *Nuffield Deptartment of Anaesthetics, University of Oxford, Radcliffe Infirmary, Oxford, UK*

Received 2 December 1996; received in revised form 21 April 1998

Abstract

In the first accompanying paper we derived an exponentially expanding mesh for use in the simulation of diffusion processes at microdisc electrodes. In this paper we use this strategy in obtaining a solution for chronoamperometry at a microdisc. We show that it is possible to obtain accuracy to better than 0.25% from a single simulation using the same mesh at all times of interest. Demands on computing resources are light, with a full simulation from switch on through to near steady state taking just a few minutes of CPU time. © 1998 Elsevier Science S.A. All rights reserved.

Keywords: Chronoamperometry; Exponentially expanding mesh; Microdisc electrodes

1. Introduction

In the first accompanying paper [1] we gave a derivation of an exponentially expanding mesh for use in the simulation of diffusion processes at microdisc electrodes. In this paper we make use of this technique to solve the problem of chronoamperometry at a microdisc, for which a wide variety of numerical simulation [2–8] and approximate analytical [9–12] solution methods have been proposed. Throughout this paper we will compare our numerical results to the approximate analytic solution given by Aoki and Osteryoung [9], which is generally accepted to be very accurate.

We begin by illustrating the effect of the boundary singularity on numerical convergence for chronoamperometry at two representative times, when no special technique is introduced to remove the effects of the singularity. We then make use of the refined meshes described in the first accompanying paper [1] to overcome the effects of the singularity. We consider two methods for dealing with the time derivative: the alternating direction implicit method (ADI) (first suggested by Peaceman and Rachford [21], and in the electrochemical literature by Heinze [5]); and a fully implicit method. The first of these is the two-dimensional analogue of the Crank–Nicolson method, and although it is second order accurate in time, it is also affected by initial oscillations due to the initial time singularity. We therefore discuss methods of adapting previous approaches to dealing with the initial time singularity [14–16] which causes these oscillations. However, it is well known that for very stiff one-dimensional problems it is not possible to overcome these difficulties [19,20]. For these problems a fully implicit method, which is not affected by initial oscillations, is more appropriate and its implementation is also described, even though it is slightly more expensive in terms of computing costs. The fully implicit method is also only first order accurate in time, and we therefore discuss the use of the Richtmyer modification as described by Rudolph [17,18] and Feldberg et al. [19,20] for one-di-

* Fax: +44 1865 273839; e-mail: david.gavaghan@comlab.oxford.ac.uk

PII S0022-0728(98)00226-5

mensional problems. These methods are combined with the generalised finite difference approximations for the spatial derivatives described in the first accompanying paper [1][1], and results are given for simulations using the specially designed mesh described in the first accompanying paper [1]. Simulated values of the non-dimensional flux are shown to be accurate to 0.25% at all non-dimensional times of interest. The results at all times are obtained from a single run on the same mesh, with the mesh generation requiring only two parameters, and computing demands are light.

As we will show in later papers in this series this technique can be straightforwardly extended to deal with the cases of membrane covered electrodes, and other electrochemical control techniques. In the next accompanying paper we consider the problem of linear sweep voltammetry at a microdisc electrode [22].

2. The model problem

Throughout this paper, we assume that semi-infinite mass transport occurs to a disc-shaped electrode and that the electrochemical reaction

$$\mathrm{A} \pm n\mathrm{e}^- = \mathrm{B} \tag{1}$$

takes place at the disc surface. Axial symmetry allows mass transport to the electrode to be modelled by the cylindrical diffusion equation

$$D\left(\frac{\partial^2 u}{\partial r^2}+\frac{1}{r}\frac{\partial u}{\partial r}+\frac{\partial^2 u}{\partial z^2}\right)=\frac{\partial u}{\partial t} \tag{2}$$

where D is the diffusion coefficient, r, z, represent the spatial coordinates, and t is time. Here we solve for the normalised concentration, $u(r, z, t) = c(r, z, t)/c_0$, where c_0 is the bulk concentration of species A.

2.1. Boundary conditions

Diffusion limited transport towards the electrode and semi-infinite diffusion conditions are assumed, so the boundary conditions are identical to those for the steady state case of the first accompanying paper [1]:

$$\begin{aligned}
&z=0 \quad r>a \quad \frac{\partial u}{\partial z}=0\\
&z=0 \quad r\le a \quad u=0\\
&z\ge 0 \quad r=0 \quad \frac{\partial u}{\partial r}=0\\
&z\to\infty \quad r\ge 0 \quad u=1\\
&z\ge 0 \quad r\to\infty \quad u=1
\end{aligned} \tag{3}$$

where a is the cathode (or electrode) radius. The singularity arises at $r = a$, $z = 0$ due to the discontinuity in the normal derivative at the disc edge.

2.2. The current

The current, $I(t)$, flowing through the disc, which is the electroanalytical response function, is given by

$$I(t)=2\pi n_e F D c_0 \int_0^a \left(\frac{\partial u}{\partial z}\right)_{z=0} r\,\mathrm{d}r. \tag{4}$$

Here n_e is the number of electrons involved and F is the Faraday constant. In order to facilitate the comparison with previous literature we introduce the non-dimensional time $\tau = 4Dt/a^2$, and we will calculate the dimensionless current, $f(\tau)$, given by

$$f(\tau)=I/4n_e F D c_0 a=\frac{\pi}{2a}\int_0^a \left(\frac{\partial u}{\partial z}\right)_{z=0} r\,\mathrm{d}r \tag{5}$$

3. Numerical methods

In order to obtain a numerical solution the semi-infinite region $[0, \infty] \times [0, \infty]$ is replaced by the finite region $[0, r_{\max}] \times [0, z_{\max}]$, and we follow Britz [23] in choosing $r_{\max}$ to be max $(2, 6\sqrt{(D\tau)})$ and $z_{\max}$ to be max$(1, 6\sqrt{(D\tau)})$ (since beyond this distance all concentration values will effectively equal the bulk concentration). This solution region is then discretised using a general rectangular mesh with mesh spacings

$$h_i = r_{i+1} - r_i, \qquad i = 0,\ldots,N_r - 1 \tag{6}$$

$$k_j = z_{j+1} - z_j, \quad j = 0,\ldots,N_z - 1 \tag{7}$$

in the r and z directions, as described in the first accompanying paper [1], where N_r, N_z are the number of mesh lines in the r and z directions, respectively.

3.1. The ADI method

We described the use of the ADI method on a regular rectangular mesh in an earlier paper [4]. Its use on a general rectangular mesh simply involves replacing the usual central difference approximations to the partial derivatives by the generalised forms given by equations 12 to 15 of the first accompanying paper [1]. Using a timestep Δt ($= \Delta\tau/4$), the approximate non-dimensional solution $U^n_{i,j}$ ($\approx u(r_i, z_j, t_n)$) is found at each time level $t_n = n\Delta t$ (where u is the continuous non-di-

[1] This approach has been used successfully for one-dimensional problems by Feldberg and Goldstein [20] who also combine the use of an exponentially expanding space grid with an increasing timestep. However, one-dimensional problems do not contain a spatial singularity at the electrode edge, and a mesh that will give accurate solutions for one dimensional problems will not be sufficiently fine for the two-dimensional problem. As described in the first of the accompanying papers [1], the spatial mesh for the two dimensional problem must be designed specifically to overcome the effects of the spatial singularity.

mensional solution in the finite domain, and $U^n_{i,j}$ is the corresponding solution to the discrete problem). The ADI method introduces an intermediate timestep at $(n+1/2)\Delta t$ to allow a splitting of the difference operators. This results in the pair of difference equations

$$\left[1-\frac{\Delta t}{2}\left(L_{rr}+\frac{1}{2}L_r\right)\right]U^{n+1/2}_{i,j}=\left[1+\frac{\Delta t}{2}L_{zz}\right]U^n_{i,j} \quad (8)$$

$$\left[1-\frac{\Delta t}{2}L_{zz}\right]U^{n+1}_{i,j}=\left[1+\frac{\Delta t}{2}\left(L_{rr}+\frac{1}{r}L_r\right)\right]U^{n+1/2}_{i,j} \quad (9)$$

where L_r, L_{rr}, L_{zz} are the generalised finite difference operators defined in equations 12 and 13 of the first accompanying paper [1]. The first of these equations is then a tridiagonal system in the r-direction at $(n+1/2)\Delta t$, and the second is a tridiagonal system in the z-direction at $(n+1)\Delta t$. Each of the corresponding matrices can be decomposed into LU form just once at the start of the calculation and stored in vector form. The resulting algorithm is very efficient computationally in terms of both storage and speed, with the added benefit that the overall method is unconditionally stable and second order accurate. Further details of this implementation are given in the Appendix.

3.2. The fully implicit method

As with the Crank–Nicolson method for 1-D problems, the ADI method is second order accurate in both time and space, but can suffer from oscillations in the concentration values when large timesteps are used. The fully implicit finite difference (FIFD) method (or Laasonen scheme) has recently been described for use in simulating one-dimensional problems by Rudolph [17,18] and has been improved upon by incorporating the Richtmyer modification (the 'FIRM' algorithm) [19,20]. The FIFD method does not give oscillations with large timesteps, but is only first order accurate in time. It can be straightforwardly extended to two-dimensional problems and differs from the ADI algorithm in that it uses only values at the current time level to calculate the finite difference equations for the spatial derivatives, rather than an average of the values at the old and new time levels. It again uses a simple backward difference to approximate the time derivative. This results in the following finite difference scheme

$$\frac{U^{n+1}_{i,j}-U^n_{i,j}}{\Delta t}=\left(L_{rr}+\frac{1}{r}L_r+L_{zz}\right)U^{n+1}_{i,j} \quad (10)$$

Re-arranging this gives

$$\left[1-\Delta t\left(L_{rr}+\frac{1}{r}L_r+L_{zz}\right)\right]U^{n+1}_{i,j}=U^n_{i,j} \quad (11)$$

Once the boundary conditions are discretised, Eq. (11) can be written as the matrix equation

$$\boldsymbol{A}\boldsymbol{U}^{n+1}=\boldsymbol{U}^n+\boldsymbol{b} \quad (12)$$

$\boldsymbol{U}^{n+1}$ is a vector of length $N=N_r\times(N_z-1)$ representing the unknown concentration values at time $n+1$, $\boldsymbol{U}^n$, $\boldsymbol{b}$ are also vectors of length N representing the known concentration values at time n and the Dirichlet boundary conditions, and A is an $N\times N$ banded matrix. This system can be solved straightforwardly using standard software packages which make use of the banded structure of the matrix. We have used routines F07BDF and F07BEF of the NAG subroutine library [27] which calculate the LU decomposition of a banded matrix and then solve the related system, respectively. It is worth noting that, as with the ADI method, the matrix A is time-independent, and so need only be decomposed into LU form once, and the system is then solved at each timestep using routine F07BEF. The number of operations required by this algorithm is approximately $6N\times\min(N_r,N_z)$ on each timestep, with a further $2N^2$ to perform the LU decomposition. The memory requirement to store the banded matrix is $3N\times\min(N_r,N_z)$. Full details of techniques for solving sparse, banded linear systems can be found in Golub and Van Loan [28].

3.2.1. The FIRM algorithm

The FIRM algorithm seeks to improve the time-accuracy of the FIFD method by using earlier (known) values of the concentration to calculate the approximation to the time derivative. The method as described by Mocak and Feldberg [19] and Feldberg and Goldstein [20] can be used in identical form for the two-dimensional problem, and requires only minor modification to the algorithm. Later in this paper where we compare the use of the FIFD and FIRM algorithms we will use the six-level Richtmyer modification, so that instead of using the simple two-point finite difference for the time derivative, as in Eq. (10), we use the six-point approximation given in Table 3 of Feldberg and Goldstein [20]. We use the same simple start-up protocol as these previous authors which assumes that prior to the initial timestep, all concentrations can be set to the bulk concentration. Details of this procedure are given in Section 3.3 of Feldberg and Goldstein [20].

3.3. Boundary conditions

The conditions on all boundaries are imposed in identical fashion to those described for the steady state problem in the first accompanying paper [1], with the exception of those on $r_{\max}$ and $z_{\max}$ which simply impose the bulk concentration via

$$U_{n,j}=1 \qquad j=1,\ldots,m-1$$

$$U_{i,m}=1 \qquad i=0,\ldots,n-1 \quad (13)$$

3.4. Calculation of the flux

At each timestep the flux is calculated in identical fashion to that described by equations Eqs. (18–20) of the first accompanying paper [1].

3.5. Computing

All programs used to generate results quoted in this paper are written in standard Fortran 77 and run on an IBM RS6000 model 550 which has a nominal clock speed of 41 MHz, but is specially designed for floating point arithmetic and is capable of a peak performance of 82 Mflops (manufacturer's figures) and a sustainable speed of 15–20 Mflops. This machine is 5 years old and is now roughly equivalent to the mid-range Pentium processors. All codes are available from the author on request.

4. Numerical convergence of the methods

In this section we consider the temporal and spatial convergence of the simulated flux calculated using each of the three numerical methods. In an earlier paper [13] we described for the steady state problem the effects of the spatial singularity at the electrode edge on the numerical convergence of the simulated flux. For the time dependent problem this same spatial singularity is present and we will again demonstrate its dramatic effect on numerical convergence. However, for the time-dependent problem there is also a time singularity at $t = 0$ due to the impulsive change in the boundary condition when the electrode is 'switched on'. As reported by Britz and Osterby [15], it has been known for some time that this second type of singularity can cause oscillations when using semi-implicit finite difference methods such as ADI to solve diffusion problems [14], and several methods of overcoming it are presented in the classic work of Crank [16]. Britz and Osterby give a graphic demonstration of the occurrence of oscillations when using the Crank–Nicolson method for one dimensional electrochemical problems, and since the ADI method can be considered as the two dimensional equivalent of the Crank–Nicolson method, we will begin by demonstrating that the same difficulty can also arise in simulating two-dimensional electrochemical problems. We then consider the effect of the spatial singularity on the convergence of the ADI method, going on to consider the temporal convergence of the FIFD and FIRM algorithms, before finally showing that all three algorithms are affected in identical fashion by the spatial singularity. These numerical experiments allow us to derive an expanding timestep method which can be combined with the refined mesh described in the accompanying paper [1] to give very accurate values of the simulated flux at all times of interest in a single run of the program for each of the three algorithms.

4.1. ADI method: convergence and oscillations on a regular grid

To illustrate the effects of the initial time and spatial singularities we use a simple regular rectangular mesh, with constant mesh spacing h in both the r and z directions, to solve the electrode problem by the ADI method with standard central difference approximations for each of the spatial derivatives as given in equation 11 of the first accompanying paper [1].

The effect of the initial time singularity when solving for the simulated flux is governed by the size of the mesh ratio, $\lambda = \Delta t/h^2 = \Delta\tau/4h^2$. This is illustrated in Fig. 1 which shows the results of solving up to time $\tau = 1.21$, using a mesh spacing $h = 0.05$ (40 points along the cathode surface), and with three values of the mesh ratio: $\lambda = 1.0$ is given by the solid line; $\lambda = 2.0$ is given by the dashed line; and $\lambda = 4.0$ is given by the dotted line. It is clear that the magnitude and duration of the oscillations increase with increasing λ as expected[2].

The effects of the spatial singularity at the electrode edge on numerical convergence are demonstrated in

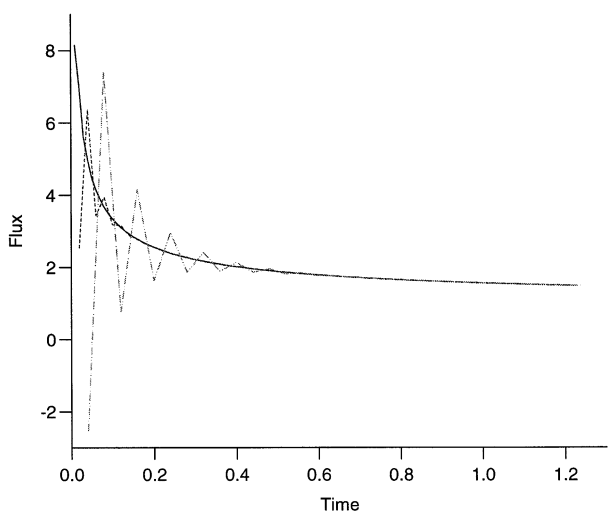

Fig. 1. Oscillations caused by the initial time singularity. The full line is calculated using $\lambda = \Delta\tau/4h^2 = 1.0$ and shows no oscillations, whilst for the dashed line $\lambda = 2.0$ we obtain small initial oscillations, and for the dotted line $\lambda = 4.0$ giving large initial oscillations and negative flux values. In both cases the oscillations decay with time, and their duration and magnitude increase with A.

[2] In numerical analysis, this effect is usually described in terms of a maximum principle i.e. it can be shown that (for example, chapter 3 Morton and Mayers [25]) both the Crank–Nicolson and ADI methods will give simulated values of the concentration that are bounded by the minimum and maximum values on the boundary (in our case 0.0 and 1.0), provided that $\lambda < 1.0$.

Table 1
Convergence results for the simulated flux $f(\tau)$ obtained by successively halving the mesh spacing h in both the r and z-directions on a regular rectangular mesh at two representative times ($\tau = 0.09$ and $\tau = 1.21$)

n_r	h	N_t	$\tau = 0.09$		N_t	$\tau = 1.21$	
			Mesh	$f(\tau)$		Mesh	$f(\tau)$
10	0.1	36	29×29	3.494	121	77×77	1.425
20	0.05	36	57×57	3.501	121	153×153	1.498
40	0.025	36	113×113	3.555	484	305×305	1.552
80	0.0125	144	225×225	3.611	1936	609×609	1.591
160	6.25×10^{-3}	576	449×449	3.656			
320	3.125×10^{-3}	2304	897×897	3.689			

Table 1, and are given at the two representative times of $\tau = 0.09$ and $\tau = 1.21$, similar to those used by previous workers [3,4,7]. The accurate analytic approximation as given by Aoki and Osteryoung [9] gives flux values of 3.768 and 1.692 at $\tau = 0.09$ and $\tau = 1.21$, respectively. The number of timesteps, N_τ, used in the numerical simulations is chosen to ensure both time accuracy and to prevent oscillations, so that for both times a value of $\lambda = 1.0$ is used for all values of $n_r \geq 40$. It is clear from the values given in Table 1 that convergence is very slow at both non-dimensional times considered, being accurate to only 2% at $\tau = 0.09$ and 6% at $\tau = 1.21$ on the finest meshes considered. Finer meshes were not used as the computing costs became prohibitive, taking 6.2 h of CPU time at $\tau = 0.09$, and 53 min of CPU time at $\tau = 1.21$ on the finest meshes used in Table 1.

It is worth noting that, as for the steady state problem in the first accompanying paper [1], the rate of convergence appears to be tending towards $h^{1/2}$ on the finer meshes, so that the effect of the spatial singularity is similar in the two problems. This suggests that the mesh refinement approach which gave such accurate and computationally efficient results for the steady problem might give equally accurate and efficient results for the time-dependent problem.

4.2. Spatial and temporal convergence of the FIFD and FIRM algorithms

Although the FIFD method does not give any oscillations in the simulated flux, or in the concentration values, since we have used a very simple backward difference formula to approximate the time derivative we obtain only first order accuracy in time i.e. the error in the approximation decays in proportion to $\Delta\tau$ rather than with the $\Delta\tau^2$ obtained with the second order ADI method. This is illustrated in Table 2 which shows the results of successively doubling the number of timesteps on a very coarse regular mesh ($h = 0.1$) at two representative times of $\tau = 0.09$ and $\tau = 1.21$. The very rapid time convergence of the ADI method contrasts with the slower first order convergence of the implicit method, although both converge to the same value. The values obtained using the six-level FIRM algorithm are also shown, and show an improvement over the FIFD algorithm but less rapid time convergence than the ADI technique[3]. It is clear that both methods give much more rapid time convergence at the longer time. This is because the truncation errors of the finite difference methods are proportional to the product of the timestep for FIFD (the timestep squared for ADI) and higher time derivatives of the concentration (chapter 2 of Morton and Mayers [24]). Since the concentration varies quite slowly with time at longer times, the overall truncation error due to the time discretisation decays very rapidly with time. As we describe below, this has important practical implications in choosing the most appropriate method for the solution of a particular electrochemical problem.

Finally we note that the ADI, FIFD and FIRM algorithms converge to the same value of the simulated flux on this very coarse mesh. Although not shown, this is also true for all of the regular meshes described in Table 1, so that all three methods are affected in identical fashion by the spatial singularity.

4.3. Obtaining time-accurate solution at both short and long times

Since we wish the methods that we have developed to be used as a general tool for simulating electrochemical problems, we would like to be able to obtain accurate results for the simulated flux over several orders of magnitude of the non-dimensional time τ (the range for

[3] For the FIRM algorithm we give the values at the end of a certain number of timesteps to allow direct comparison between the methods. However, Feldberg and Goldstein [20] give a lengthy discussion of the way in which the start-up protocol affects the 'temporal location' of the calculated concentration values, and this should be taken into account in any practical application of the FIRM algorithm. It is also worth noting that the simple start-up protocol for the FIRM algorithm results in oscillations in the simulated flux over the first few timesteps.

Table 2
Convergence results for the simulated flux $f(\tau)$ obtained by successively halving the mesh spacing h in both the r and z-directions on a regular rectangular mesh at two representative times ($\tau = 0.09$ and $\tau = 1.21$)

h	N_τ	FIFD	FIRM	ADI	FIFD	FIRM	ADI
		$f(0.09)$	$f(0.09)$	$f(0.09)$	$f(1.21)$	$f(1.21)$	$f(1.21)$
0.1	20	3.560	3.535	3.493	1.439	1.434	1.424
0.1	40	3.527	3.514	3.494	1.432	1.429	1.425
0.1	80	3.510	3.504	3.494	1.428	1.427	1.425
0.1	160	3.502	3.499	3.494	1.427	1.426	1.425
0.1	320	3.498	3.496	3.494	1.426	1.425	1.425
0.1	640	3.496	3.495	3.494			
0.1	1280	3.495	3.494	3.494			

which results are usually given is $\tau = 10^{-2}$ to 10^{2}). In the first accompanying paper [1] we described how to derive a mesh on which the effects of the spatial singularity could be effectively removed for the steady state problem. We will show later that this mesh can be used to give excellent spatial accuracy for chronoamperometry over this range of times, but first we consider the time convergence of the three solution methods on the final mesh that we derived, which uses a minimum mesh spacing of $h = 8 \times 10^{-5}$ immediately adjacent to the singularity, and a mesh expansion factor $f = 1.175$. It was shown that this approach leads to the mesh spacings h_i, k_j in the r, z directions as follows

$$
\begin{aligned}
h_{n_r - 1} &= h_{n_r} = h_{\text{last}} \\
h_{n_r - i} &= f h_{n_r - i + 1} \qquad \text{for} \quad i = 2,\ldots,n_r \\
h_{n_r + i} &= f h_{n_r + i - 1} \qquad \text{for} \quad i = 1,\ldots,N_r \\
k_0 &= h_{\text{last}} \\
k_{j+1} &= f k_j \qquad \text{for} \quad j = 0,\ldots,N_z.
\end{aligned} \tag{14}
$$

The values of N_r and N_z are determined automatically to ensure that $r_n = r_{n_r + N_r} > r_{\max}$ and z_{N_z} once the values of h_{last} and f have been chosen (this is used as the stopping criterion in the mesh-generation algorithm).

Table 3 gives the time convergence of the simulated flux, calculated on this mesh, at either end of the usually quoted range (the analytic values of the flux given by Aoki and Osteryoung [9] being $f(0.01) = 9.657$ and $f(100) = 1.073$). For the ADI method we would need to use a timestep of the order of 5×10^{-9} if we were to obtain a mesh ratio $\lambda = 1.0$ and avoid oscillations. This in turn implies the need for around 2×10^{10} timesteps to reach a final time of $\tau = 100$, which is clearly not feasible in reasonable computing time. Even if we were prepared to put up with oscillations provided that these had decayed away by the time of interest, we would still require around 2000 timesteps even for very short times, and for $\tau = 100$ we would require 200000 timesteps. For the fully implicit methods, at $\tau = 0.01$ we see very similar rates of time convergence to that on the regular mesh, with FIFD giving first order convergence and FIRM more rapid convergence. At the longer time we see the dramatic effect of the very slow variation of the concentration values with time, so that the contribution to the overall truncation error from the time terms is negligible and we can obtain extremely accurate results using just ten timesteps for both FIFD and FIRM.

It is clear from the results given in Table 3 that it will not be possible to use a fixed timestep to obtain accurate results at all values of τ of interest in a single run of the program for any of the three algorithms. The standard technique for overcoming this problem is to use an exponentially expanding timestep [20]. However this would involve re-inverting the matrices associated with each of the methods at each timestep, which is an extremely costly procedure in terms of computing time. We have therefore adapted a suggestion made by Pearson [14] for overcoming oscillations associated with the Crank–Nicolson scheme which involved using a special calculation on the first timestep, by subdividing the first interval into a number of smaller sub-intervals. A similar idea was used by Britz and Osterby [15] who replaced the first timestep by a series of exponentially increasing (they suggested a factor of two) subintervals whose sum is $\Delta\tau$. Whilst both of these techniques can also be used both to reduce the oscillations associated with the ADI method on the refined mesh, and to achieve time accuracy for all three algorithms, neither is sufficient to give accurate solutions within acceptable computing demands.

We therefore use an algorithm which begins with a timestep that is sufficiently small to remove all oscillations in the ADI method on the refined mesh (in practice we ensure that $\lambda < 1.0$ initially), but can also be used with both of the fully implicit methods to give accurate values at all values of τ of interest with reasonable computing demands. After some 'suitable' number of timesteps (say M) the timestep is increased by a suitable factor, γ. After a further M steps this is repeated with the timestep again increased by a factor

Table 3
Convergence results for the simulated flux $f(\tau)$ as a function of the number of timesteps for each of the three algorithms at short ($\tau = 0.01$) and long ($\tau = 100$) times

N_τ	FIFD	FIRM	ADI	N_τ	FIFD	FIRM	ADI
	$f(0.01)$	$f(0.01)$	$f(0.01)$		$f(100)$	f(100)	f(100)
125	9.680	9.671	Osc.	10	1.076	1.074	Osc.
250	9.667	9.662	Osc.	20	1.074	1.074	Osc.
500	9.660	9.658	Osc.	40	1.074	1.073	Osc.
1000	9.657	9.656	Osc.				
2000	9.655	9.655	9.653	1×10^5			Osc.
4000	9.654	9.654	9.654	2×10^5			1.073

γ, and the process of increasing the timestep exponentially continues until the final time of interest is reached. We choose the 'suitable' number of timesteps and factor by numerical experiment as described further below. In practice this algorithm is modified slightly in order to give values of the simulated flux at times which can be compared with the work of previous authors.

A simplified example of the effectiveness of this technique for removing the oscillations in the ADI method is shown in Fig. 2 which uses a regular mesh with $h = 0.05$. The full line is calculated using an initial value of $\Delta\tau_0 = 1.0^{-3}$ with a value of $\gamma = 10.0$, giving an initial value of $\lambda = 0.4$. We also choose $M = 20$, but use the slightly modified algorithm so that the timestep is increased by a factor of 10.0 after the first 10 steps at time $\tau = 0.01$ to $\Delta\tau = 0.01$ (with $\lambda = 4.0$), then again after a further 19 steps at time $\tau = 0.2$ to 0.1 ($\lambda = 40.0$). As can be seen no oscillations result, and the calculation up to $\tau = 1.3$ requires just 40 timesteps in total. The dashed line in Fig. 3 is calculated using the same mesh with a value of $\Delta\tau = 0.01$ ($\lambda = 4.0$) throughout the calculation, requiring 130 timesteps to reach $\tau = 1.3$ whilst giving large initial oscillations and negative flux values.

4.4. Combining mesh refinement with an increasing timestep

When combined with the very refined mesh derived in the accompanying paper [1] (using $h_{\text{last}} = 8 \times 10^{-5}$, and $f = 1.175$), a much smaller initial timestep must be used to overcome the oscillations in the ADI method, and the time convergence results given in Table 3 also suggest that we need to use at least 1000 timesteps to obtain an accurate value of the flux at $\tau = 0.01$ when using the FIFD method. Fig. 3 shows the results of our numerical experiments to determine the most appropriate values of M and γ on the refined mesh. Each of the curves shown gives the difference (as a percentage of the analytic value) between the simulated flux and the analytic approximation given by Aoki and Osteryoung [9][4], plotted against $\log_{10}(\tau)$. Curves A and B are calculated calculated using FIFD and FIRM, respectively and values of $M = 1000$ and $\gamma = 10.0$, starting with an initial timestep of $\Delta\tau = 10^{-6}$. Curves C, D and E are all calculated using the ADI method with $\gamma = 10.0$ and $M = 1000$, 2000, and 4000, respectively, and an initial timestep of $\Delta\tau = 10^{-8}$ (chosen sufficiently small to remove all oscillations). For values of $\tau < 10^{-2}$ increasing the timestep by a factor of 10 makes no difference for the ADI method, whilst for the FIFD and FIRM algorithms it results in a slight increase in the error — this is more pronounced for FIRM due to the effects of the startup protocol[5]. For values of $\tau > 10$, the position

[4] We use equation 8 of ref. [9] for $r < 1$, and equation 7 of [9] for $r > 1.0$.

[5] This could be improved by using a more sophisticated startup procedure such as those described by Feldberg and Goldstein, but for this problem the FIFD algorithm gives sufficiently accurate results (as discussed in the next section) so that this is not necessary.

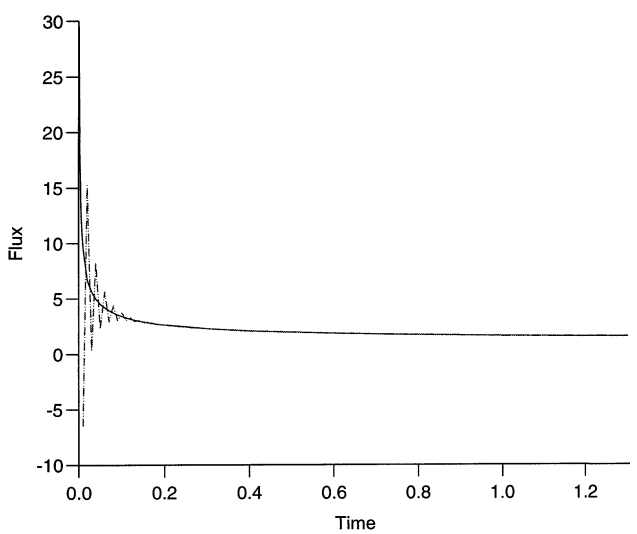

Fig. 2. Use of an increasing timestep to remove the oscillations caused by the initial time singularity on a regular mesh with $h = 0.05$. The full line is calculated using an initial value of $\lambda = 0.4$ which was increased to 4.0 after 10 timesteps, and to 40.0 after 30 timesteps, and shows no oscillations, and requires just 40 timesteps in total. The dashed line uses a value of $\lambda = 4.0$ throughout the 130 timesteps of the calculation, giving large initial oscillations and negative flux values.

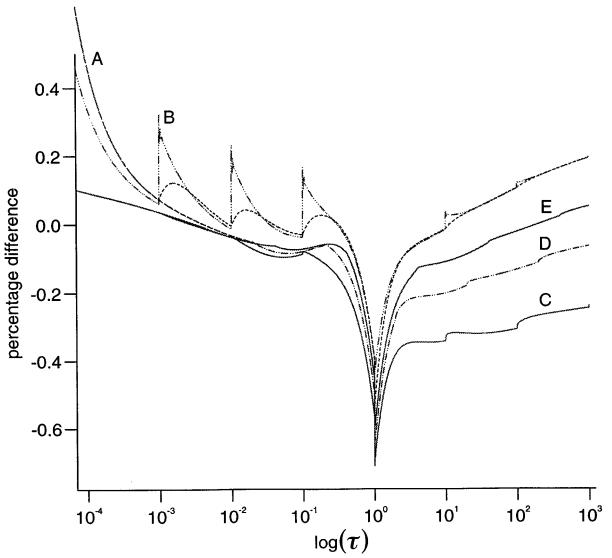

Fig. 3. Percentage difference between the analytic and simulated flux values obtained by combining mesh refinement with an increasing timestep for each of the three numerical algorithms. Curves A and B are calculated calculated using FIFD and FIRM respectively, and values of $M = 1000$ and $\gamma = 10.0$, starting with an initial timestep of $\Delta\tau = 10^{-6}$. Curves C, D and E are all calculated using the ADI method with $\gamma = 10.0$ and $M = 1000$, 2000, and 4000, respectively and an initial timestep of $\Delta\tau = 10^{-8}$.τ is plotted on a logarithmic scale.

of the timestep having no effect on the FIFD and little effect on FIRM. However, it is clear that for the ADI method, the size of the timestep has an increasingly marked effect on the error. Closer inspection reveals this to be a stability problem, with oscillations being introduced into the concentration values at longer times when the timestep is comparatively large. This problem can be reduced by lengthening the interval between increases in the timestep, as shown in curves C, D and E where M is 1000, 2000 and 4000 — as M is further increased, the curves calculated using the ADI method gradually come to lie on top of those for the FIFD and FIRM algorithms, but of course the computing time taken increases proportionately.

4.5. Spatial convergence and comparison with previous results

In discussing the results presented in Table 3 and in Fig. 3, we were concerned primarily with the time-convergence, and did not mention spatial convergence obtained on the refined mesh. However, it is clear from Fig. 3 that except in the range $0.5 < \tau < 2$ we obtain extremely good agreement with both the analytic solution of Aoki and Osteryoung [9] and with previously published numerical simulations. Indeed, in Table 4 where we compare our results with those of previous workers at the usually quoted non-dimensional times, it is clear that the spatial accuracy that we obtained in the first of the accompanying papers for the steady problem has carried through to the time-dependent problem, with maximum errors (other than around $\tau = 1.0$) being less than 0.1% for all three methods. These values were calculated using values of $M = 1000$, $\gamma = 10$ and an initial timestep of 10^{-5} for both the FIFD and FIRM (curves A and B in Fig. 3), and $M = 4000$, $\gamma = 10$ and an initial timestep of 10^{-7} for the ADI method (curve E in Fig. 3). The refined mesh was generated using values of $h_{\text{last}} = 8.0 \times 10^{-5}$ and $f = 1.175$ in Eq. (14), giving a mesh with 122 points in the r-direction and 74 in the z-direction, with 48 points covering the electrode surface. We consider, in Table 4, flux values up to time $\tau = 100$ to allow comparison with previous workers, and we therefore choose $r_{\max}$ and $z_{\max}$ to be greater than $6\sqrt{D\tau} = 60$. It is worth noting that the maximum error reported by Amatore and Fossett [2] using the accurate conformal mapping technique was at $\tau = 1.21$ (these authors did not give a value at $\tau = 0.64$) and that few authors using simulation methods have given a value of the simulated flux at $\tau = 1.0$. The mesh that we use to obtain the results in Fig. 3 and Table 4 is designed to strike a balance between various numerical errors, and should give its most accurate results in the middle of the range of τ values of interest i.e. around $\tau = 1.0$. In addition, $\tau = 1.0$ is the point at which we switch from using the short-time analytic solution, to the long-time analytic solution. It therefore seems likely that the maximum difference that we observe of 0.6%, occurring at $\tau = 1.0$ between the analytic and simulated fluxes is due to error in the analytic solution, and the lower simulated values of the flux in this region are also likely to be correct to within 0.1% i.e. the error is in the analytic solution for $0.5 < \tau < 2.0$. In Fig. 3, we have also included results up to $\tau = 10^3$, and it can be seen that the flux is overestimated by the fully implicit methods by about 0.2%. This is in line with the results presented in Section 4.5 of the accompanying paper [1], since we designed the mesh to give very accurate results for intermediate τ values. If we were primarily interested in near steady state values, then we could use a slightly smaller value of the expansion factor as described in that paper. For completeness, we have run the FIFD algorithm to a value of $\tau = 10^6$ (the stability and time accuracy at long times allows us to do this using a timestep of 10^4 or just 100 timesteps in around 40 s of CPU time) and obtained an overestimation of 0.27%, which can be considered to be the maximum error associated with simulation on this mesh, and is sufficiently accurate for most experimental situations.

4.6. Computing time

The values of the CPU times given in Table 5 are all for the calculations made to obtain the results given in

Table 4
Comparison of the analytic results of Aoki and Osteryoung [9] with those obtained using a conformal map [2] and the results from this work

Flux $f(\tau)$					
Time (τ)	Analytic	Conformal map	FIFD	FIRM	ADI
0.01	9.657	9.688	9.657	9.678	9.653
0.04	5.235	5.243	5.235	5.235	5.232
0.09	3.768	3.772	3.767	3.766	3.765
0.25	2.605	2.607	2.605	2.605	2.603
0.64	1.968		1.965	1.965	1.964
1.00	1.768		1.758	1.759	1.758
1.21	1.691	1.686	1.685	1.686	1.684
2.25	1.495	1.493	1.493	1.493	1.492
4.00	1.367	1.367	1.367	1.365	1.364
6.76	1.279	1.279	1.279	1.279	1.278
25.0	1.144		1.145	1.145	1.143
100.0	1.073		1.073	1.073	1.072

Table 4. Whilst the ADI method is an order of magnitude quicker per timestep than the fully implicit methods, it becomes inaccurate and unstable at longer times if the timestep is increased too rapidly. The single run of the codes used to generate the values given in Table 5 required 4591 timesteps for each of the fully implicit methods—equivalent to about 20 min of CPU time, but 22592 for the ADI method—equivalent to about 10 min of CPU time. However, comparing the computational efficiency of the methods in this simple fashion is not fair to any of them—the most appropriate method to use depends on the particular problem that is being solved. For example, if we were really interested in near-steady-state values, we could use either the FIFD or FIRM algorithms with 100 timesteps and obtain the value of the flux at $\tau = 100$ to 0.1% accuracy in 40 s of CPU time, and similarly using the ADI method we could obtain $f(0.01)$ using 1800 timesteps in 20 s. We discuss further below which of the methods might be considered most appropriate.

5. Discussion

In the first accompanying paper [1] we derived a mesh refinement strategy which was designed to give very accurate values of the simulated flux at a microdisc electrode when using a variety of electrochemical control techniques. In this paper we have described how to use this approach to solve the problem of chronoamperometry at a microdisc electrode. By adapting previous approaches to dealing with errors due to the time-dependency of the problem, and particularly the initial time singularity, we have shown that it is possible to obtain values of the simulated flux with an accuracy of about 0.1% in the range of $0 < \tau < 100$ for which the mesh was designed, and to about 0.25% for all values of τ.

We have described the use of this mesh with three methods of solving the resulting finite difference equations–ADI, FIFD and FIRM. We have shown that the ADI algorithm is an order of magnitude faster per timestep than the other fully implicit methods, but can give problems with stability when used with a large timestep. The ADI method is also much more accurate at short times than the fully implicit methods. However, the codes were run on a fairly old computer, and it seems likely that this slight advantage enjoyed by the ADI method will soon become irrelevant with current advances in technology. So, the ADI is the method of choice where computing time or short-time accuracy is the major consideration. It is also appropriate for problems which do not contain an initial time singularity such as linear sweep voltammetry (LSV), which we consider in the accompanying paper [22]. Of the two fully implicit methods, for problems such as chronoamperometry where the error due to the size of the timestep decays rapidly with time, there is also little to choose between the FIFD and FIRM algorithms. However, from a programming perspective, the difference between the two algorithms is trivial[6] and we would therefore recommend that any readers attempting to use the approach described here should consider coding

Table 5
Comparison of the CPU times taken by the FIFD, FIRM and ADI methods

CPU time per timestep (s)		
FIFD	FIRM	ADI
0.252	0.266	0.0255

[6] Indeed, the FIFD is simply a two level FIRM algorithm.

up an n-level FIRM algorithm to give maximum flexibility in adapting the code for other applications. Again, in the accompanying paper, we show that for simulations of LSV where the time truncation errors do not decay, the FIRM algorithm is extremely accurate and efficient, whilst the FIFD algorithm is much less accurate.

Acknowledgements

The author is pleased to acknowledge the financial support of the Wellcome Trust in the form of a Biomathematical Training Fellowship, and a Career Development Fellowship from the Medical Research Council which has allowed him to undertake this research.

Appendix A. An efficient implementation of the ADI algorithm

As described in the main text, the advantage of the ADI method is that we can obtain our solution at level $n+1$ from that at level n, by solving only tridiagonal systems in alternating directions at each half step. If we write either of Eqs. (8) and (9) in the form

$$a_v U_{v-1} + b_v U_v + c_v U_{v+1} = d_v \qquad \text{for} \qquad v = 1,\ldots,\kappa \tag{15}$$

then these become $A\mathbf{x} = \mathbf{d}$ with

$$A = \begin{pmatrix} b_1 & c_1 & \ddots & 0 \\ a_2 & \ddots & \ddots & \\ & \ddots & \ddots & c_{\kappa-1} \\ 0 & & a_\kappa & b_\kappa \end{pmatrix} \tag{16}$$

In his original paper describing the use of the ADI method in electrochemical simulations, Heinze [5] gave no details of how he solved the tridiagonal systems that arise from the splitting of the difference operators. The standard technique for solving tridiagonal systems is known as the Thomas algorithm (Morton and Mayers [24]), and this appears to have been essentially the technique used by Taylor et al. [25], although their algorithm was greatly complicated by the inclusion of the 'dual matrix method' of Britz et al. [26]. Since the matrix A of Eq. (16) is diagonally dominant, we could also use the Thomas algorithm. However, A is also a constant matrix in time, so that it is much more efficient to decompose each tridiagonal matrix into upper (U) and lower (L) triangular matrices such that $A = LU$ just once at the start of the calculation, and use them at each half time step to obtain the solution. It can easily be verified that

$$L = \begin{pmatrix} 1 & \ddots & 0 \\ e_2 & \ddots & \ddots \\ & \ddots & \ddots \\ 0 & e_\kappa & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} f_1 & g_1 & 0 \\ \ddots & \ddots & \\ & \ddots & g_{\kappa-1} \\ 0 & \ddots & f_\kappa \end{pmatrix} \tag{17}$$

with

$$f_1 = b_1$$

$$f_i = b_i - e_i c_{i-1} \qquad i = (1,\ldots\kappa)$$

$$e_i = a_i / f_{i-1} \qquad i = (1,\ldots\kappa)$$

$$g_i = c_i \qquad i = (1,\ldots,\kappa-1) \tag{18}$$

It is important to note that the mesh spacings are the same along all mesh lines in a given coordinate direction, and that the boundary conditions are identical for all rows (when solving in the r-direction), whilst for all columns (z-direction) we have just two types of boundary condition (above the electrode and above the insulator). We therefore obtain only three different tridiagonal matrices over the whole mesh, which means that we need to do just three decompositions and store only nine vectors (the e, f, g of Eq. (18)) of length k to calculate the solution for all mesh lines in each of the r and z-directions. Once in the form $LU\mathbf{x} = \mathbf{d}$, by writing $\mathbf{z} = U\mathbf{x}$ we may solve the system $L\mathbf{z} = \mathbf{d}$ by forward substitution using

$$z_1 = d_1$$

$$z_i = d_i - e_i z_{i-1} \qquad \text{for } i = 2, \kappa \tag{19}$$

and then obtain our solution $\mathbf{x}$ from $U\mathbf{x} = \mathbf{z}$ by backward substitution, using

$$x_\kappa = z_\kappa / f_\kappa$$

$$x_{\kappa-i} = z_{\kappa-i} - g_{\kappa-i} x_{\kappa-i+1} / f_{\kappa-i} \qquad \text{for } i = 1, \kappa-1 \tag{20}$$

This results in an extremely efficient algorithm both in terms of calculation and storage.

References

[1] D.J. Gavaghan, J. Electroanal. Chem. 456 (1998) 1.
[2] C.A. Amatore, B. Fossett, J. Electroanal. Chem. 293 (1990) 19.
[3] J. Galceran, D.J. Gavaghan, J.S. Rollett, J. Electroanal. Chem. 394 (1995) 17.
[4] D.J. Gavaghan, J.S. Rollett, J. Electroanal. Chem. 295 (1990) 1.
[5] J. Heinze, J. Electroanal. Chem. 124 (1981) 73.
[6] D. Shoup, A. Szabo, J. Electroanal. Chem. 140 (1982) 237.

[7] D. Shoup, A. Szabo, J. Electroanal. Chem. 160 (1984) 1.
[8] M.W. Verbrugge, D.R. Baker, J. Phys. Chem. 96 (1992) 4572.
[9] K. Aoki, J. Osteryoung, J. Electroanal. Chem. 160 (1984) 335.
[10] M. Fleischmann, J. Daschbach, S. Pons, J. Electroanal. Chem. 250 (1988) 269.
[11] C.G. Phillips, J. Electroanal. Chem. 333 (1992) 11.
[12] K.B. Oldham, J. Electroanal. Chem. 122 (1981) 1.
[13] D.J. Gavaghan, J. Electroanal. Chem. 420 (1996) 147.
[14] C.E. Pearson, Math. Comput. 19 (1965) 570.
[15] D. Britz, O. Osterby, J. Electroanal. Chem. 368 (1994) 143.
[16] J. Crank, The Mathematics of Diffusion, Clarendon, Oxford, 1975.
[17] M. Rudolph, J. Electroanal. Chem. 314 (1991) 13.
[18] M. Rudolph, J. Electroanal. Chem. 338 (1995) 85.
[19] J. Mocak, S.W. Feldberg, J. Electroanal. Chem. 378 (1994) 31.
[20] S.W. Feldberg, C.I. Goldstein, J. Electroanal. Chem. 397 (1995) 1.
[21] DW. Peaceman, H.H. Rachford, J. Soc. Indust. Appl. Math. 3 (1955) 28.
[22] D.J. Gavaghan, J. Electroanal. Chem. 456 (1998) 25.
[23] D. Britz, Digital Simulation in Electrochemistry, 2nd ed., Springer, Heidelberg, 1988.
[24] K.W. Morton, D.F. Mayers, Numerical Solution of Partial Differential Equations, Cambridge University Press, Cambridge, 1994.
[25] G. Taylor, H.H. Girault, J. McAleer, J. Electroanal. Chem. 293 (1990) 19.
[26] D. Britz, J. Heinze, J. Mortensen, M. Storzbach, J. Electroanal. Chem. 240 (1988) 27.
[27] The NAG Fortran Library Manual-Mark 14, The Numerical Algorithms Group Limited, 1990.
[28] G.H. Golub, C.F. Van Loan, Matrix Computations, 2nd ed., Johns Hopkins, London, 1989.