An Algorithm for ODES from Atmospheric Dispersion Problems.

I. Ahmad and M. Berzins

School of Computer Studies, The University of Leeds, Leeds LS2 9JT, UK.

Abstract

The solution of large systems of ordinary differential equations o.d.e.s, arising from atmospheric dispersion problems is considered. An algorithm using a method due to Klopfenstein is adopted as the main method and combined with an approximate Jacobian factorisation and a Gauss-Seidel iteration to provide an efficient solver. The approach is contrasted with that of using implicit-explicit multistep methods. Numerical experiments are presented to illustrate the performance of the method.

1 Introduction.

The increasing level of air pollution makes it ever more desirable to use simulation to help increase awareness and understanding of the problem. One example is that of power station plumes which are concentrated sources of NO_x emissions, [7]. The photo-chemical reaction of this NO_x with polluted air leads to the generation of ozone at large distances downwind from the source. The transport of the plume and the chemical reactions are modelled by the atmospheric diffusion equation:

$$\frac{\partial c_s}{\partial t} = -\frac{\partial u c_s}{\partial x} - \frac{\partial w c_s}{\partial y} + \frac{\partial}{\partial x}\left(K_x \frac{\partial c_s}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial c_s}{\partial y}\right) \tag{1}$$
$$+R_s(c_1, c_2, ..., c_q) + E_s - (\kappa_{1s} + \kappa_{2s})c_s,$$

where c_s is the concentration of the s'th compound, u and w are wind velocities, K_x and K_y are diffusivity coefficients and κ_{1s} and κ_{2s} are dry and wet deposition velocities respectively. E_s describes the distribution of emission sources for the s'th compound and R_s represents the chemical reaction rates which may contain nonlinear terms in c_s. The simple chemical mechanism used by [5] contains only 10 species, and 8 coupled p.d.e.s, but does however represent the main features of a tropospheric mechanism, namely the competition of the fast inorganic reactions with the slower reactions of volatile organic compounds. This separation in time-scales generates severe stiffness and so requires the use of an implicit o.d.e. method.

The general trend in atmospheric chemistry is to use models incorporating an ever large number of reactions in the chemical schemes describing the atmospheric chemistry. The chemical kinetics arising from the atmospheric chemistry are non-autonomous, because generally the reaction rates are both time and temperature dependent, thus giving rise to abrupt and sudden changes in the concentration of the chemical species and hence also the timesteps.

The spatial discretization scheme used by [5] for the p.d.e. (1) results in a system of differential equations, which can be written as the initial value problem:

$$\underline{\dot{Y}} \ = \ \underline{F} \ \ (\, t, \ \underline{Y}(t) \ ) \ \ , \ \underline{Y}(0) \ \ \textit{given} \ , \qquad (2)$$

where the vector, $\underline{Y}(t)$, is defined by $\underline{Y}(t) \ = \ [Y_1(t), ..., Y_N(t) \ ]^T$. $Y_i(t)$ is a numerical approximation to the exact solution to one of the p.d.e.s evaluated at the centroid of a triangle. A method of lines approach is used to numerically integrate equation (2) thus generating an approximation, $\underline{y}(t)$, to the vector of exact p.d.e. solution values at the mesh points. At present the Theta method with the iterative method of [4] has been used with some success to solve atmospheric dispersion problems, [5]. Recently the second order backward differentiation method (BDF2 from hereon) with Gauss-Seidel iteration has worked well in solving o.d.e.s from atmospheric chemistry, see [14].

The aim of the paper is to consider alternatives to both the Theta method and to BDF2 by investigating whether or not the NDF method of Klopfenstein [11], as suggested by Shampine [12] [13] forms a viable alternative. and to try and understand the best approach to be used in dealing with the combination of transport and chemistry in atmospheric problems. The form of the paper is that Section 2 contains a description of the method of Klopfenstein [11] and discusses the differences between this method and the BDF2 method. Section 3 describes the method used to solve the large systems of nonlinear equations arising at each timestep, including an adaptive Gauss-Seidel iteration. The numerical experiments in Section 4 on realistic chemistry problems show that the NDF method provides a good alternative to the existing methods. In section 5 consideration is extended to the IMEX approach of Ascher et al. [1] and the properties of the NDF method considered in relation to the BDF2 method. This makes it possible to understand the differences between the IMEX and nonlinear equations splitting approaches in terms of both convergence of the iteration and accuracy. The result of this comparison is that a single constraint is identified which appears to be important for both approaches.

2 Klopfenstein NDF Method

In 1971 Klopfenstein published a class of multistep methods, modified from BDF methods in such a way that they have better stability properties and lower error constants in some cases. For example the second order method stepsize may be 20% larger then BDF2, see [11]. The methods are termed numerical differentiation methods or NDFs. The method was recommended by Shampine many years ago [12], but does not appear to have been implemented in general purpose software until recently [13]. The cost of the improvements over BDF2 is that one more back value is used. No extra storage is required however as the back value is already present in the Milne-type error estimate already used by the BDF code. For simplicity the Klopfenstein method of order 2, NDF2 hereafter, will be described by starting from the BDF2 formula given by

$$\frac{\underline{y}_{n+1} - \sum_{i=1}^{2} \underline{y}_{n+1-i}\alpha_i}{k\gamma} - \underline{F}(t_{n+1}, \underline{y}_{n+1}) = 0 \tag{3}$$

where the coefficients α_i and γ are well known, e.g. [2]. Using the Nordsieck vector form of the BDF the predicted values of the solution and the first two time derivatives are given by $\underline{y}^p_n(t_n)$, $\underline{y}^{(1)p}_n(t_n)$, and $\underline{y}^{(2)p}_n(t_n)$ are given in terms of the existing derivatives by

$$\underline{y}^{(l)\ p}_{n+1} = \sum_{i=l}^{2} \frac{\underline{y}^{(i)}_n k^{i-l}_{n+1}}{(i-l)!} \quad l = 0, \ldots, 2 \tag{4}$$

The corrected vector, $\underline{\alpha} = \frac{\underline{y}_{n+1} - \underline{y}^p_{n+1}}{k\gamma}$ can be found by solving the system of non-linear equations, see [2],

$$\underline{y}^{(1)p}_{n+1} + \underline{\alpha} - \underline{F}(t_{n+1}, \underline{y}^p_{n+1} + k\gamma\underline{\alpha}) = 0 \ , \quad \gamma \ = \ 2/3 \ , \tag{5}$$

The predictor values are then corrected by

$$\underline{y}^{(i)}(t_{n+1}) = \underline{y}^{(i)p}(t_{n+1}) + \gamma_i k^{1-i}_{n+1}\underline{\alpha} \tag{6}$$

where $\gamma_0 = \gamma_2 = 2/3$ and $\gamma_1 = 1$. The NDF2 formula [11] may be implemented in the same Nordsieck framework by writing it as a correction to equation (4)

$$\frac{\underline{y}_{n+1} - \sum_{i=1}^{2} \underline{y}_{n+1-i}\alpha_i}{k\gamma} - \hat{\alpha}\frac{(\underline{y}_{n+1} - \underline{y}^p_{n+1})}{k\gamma} - \underline{f}(t_{n+1}, \underline{y}_{n+1}) = 0, \tag{7}$$

where $-1/9 \leq \ \hat{\alpha} \ \leq \ 1/3$. We define the new variable $\gamma^* = \frac{\gamma}{(1-\hat{\alpha})}$, and write equation (7) as

$$\underline{y}^{(1)p}_{n+1} + \underline{\beta} - \underline{f}(t_{n+1}, \underline{y}^{p}_{n+1} + k\gamma^* \underline{\beta}) = 0 \qquad (8)$$

where β is defined by $\frac{y_{n+1} - y^p_{n+1}}{k\gamma^*}$. Hence on comparing the coefficients of $\underline{y}_{n+1}$, in equations (7) and (8) it follows that $\gamma^* \ = \ \frac{\gamma}{(1-\hat{\alpha})}$ and as $\hat{\alpha} \ = \ -\frac{1}{9}$, then $\gamma^* \ = \ 0.9\gamma$. One property of the q th order NDF method is that the leading term of the truncation error is $\hat{\alpha} + 1/(q+1)$, where for BDF $\hat{\alpha} = 0$. Hence the leading term of local truncation error associated with the BDF2 ($\hat{\alpha}$=0) is $\frac{1}{3}$ and that of Klopfenstein formula ($\hat{\alpha} = -\frac{1}{9}$) is $\frac{1}{6}$, which is still twice that of the trapezoidal rule however. Hence the local error estimate is halved and the stepsize at the order two is increased by the factor of about 1.26 over BDF2 for the same error. A recent comparison of BDF and NDF methods by Shampine and Reichelt [13] shows that on a range of stiff test problems the NDF code is on average about 8% faster and uses an average of about 11% fewer steps on all problems except one.

2.1 Stability Properties

The fixed step NDF2 formula, ($\hat{\alpha} = -1/9$), as will be used below is:

$$\underline{F}(t_{n+1}, \underline{y}_{n+1}) - \frac{1}{6k}(10\underline{y}_{n+1} - 15\underline{y}_{n} + 6\underline{y}_{n-1} - \underline{y}_{n-2}) = 0. \qquad (9)$$

In order to determine the stability property of the method Klopfenstein considered the single equation $\dot{y} = \ \eta y$ and showed that A-stability of the method requires that $-\frac{1}{9} \leq \ \hat{\alpha} \ \leq \ \frac{1}{3}$. The stability region has been plotted for three different values of $\hat{\alpha}$ i.e. $\hat{\alpha} \ = \ 0, \ -\frac{1}{9}, \ -\frac{2}{9}$ as shown in Figure (1). Recalling

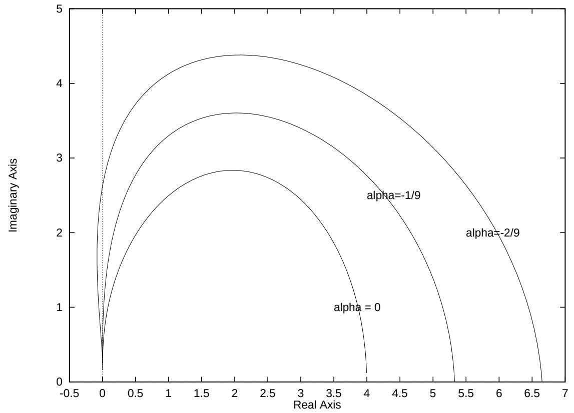

Fig. 1. Stability Region

that $\hat{\alpha} = 0$ is BDF2 and $\hat{\alpha} = -\frac{1}{9}$ is NDF2, the comparison of stability region corresponding to these two values of α shows that the stability region in the right half plane corresponding to $\hat{\alpha} = -\frac{1}{9}$ is (desirably) smaller than the BDF2 stability region corresponding to $\hat{\alpha} = 0$, see Klopfenstein [11].

3 Nonlinear Equations Splitting Algorithm.

In the case when a modified Newton method is used to solve the nonlinear equations at each timestep, the system of linear equations to be solved for the $m + 1$ th correction to the solution $\underline{\Delta y}$ are:

$$[I - k\gamma J]\ \underline{\Delta y}_m \ = \ \underline{r}\left(t^m_{n+1}\right) \tag{10}$$

where $J = \frac{\partial\, \underline{F}}{\partial \underline{y}}$, $\underline{r}\left(t^m_{n+1}\right) \ = \ -\,\underline{y}^m_{n+1} + \underline{z}_n + \gamma k \underline{F}(t_{n+1}, \underline{y}^m_{n+1})$, $\underline{\Delta y}_m \ = \ \left[\underline{y}(t^{m+1}_{n+1}) - \underline{y}(t^m_{n+1})\right]$ and $\underline{z}_n \ = \sum_{i=1}^{2} \underline{y}_{n+1-i}\alpha_i$ in the fixed timestep case. The solution of the system of equations (10) constitutes the major computational task of a method of lines calculation. In cases where large o.d.e. systems that result from the discretization of the flow problems with complex chemistry the c.p.u. times may be excessive unless special iterative methods are used to solve the system linear equations given by (10).

One common approach, e.g. [4] and the references therein, is to consider the o.d.e. function $\underline{F}\ (t,\ \underline{y}(t)\ )$ defined by equation (2) and decompose it into two parts:

$$\underline{F}\ (t,\ \underline{y}(t)\ ) \ = \ \underline{F}^f\ (t,\ \underline{y}(t)\ ) \ + \ \underline{F}^s\ (t,\ \underline{y}(t)\ ) \tag{11}$$

where $\underline{F}^f\ (t,\ \underline{y}(t)\ )$ represents the discretization of the advective flux terms uc_s and wc_s in equation (1) and $\underline{F}^s\ (t,\ \underline{y}(t)\ )$ represents the discretization of the of the diffusion and source terms in the same equation. The nonlinear equations splitting method uses the approximate factorisation of Jacobian matrix employed by the time integrator method within a Newton iteration:

$$I - k\gamma J \ \approx \ [I - k\gamma\ J_f]\ \ [I - k\gamma\ J_s\ ] \ + \ O(k^2). \tag{12}$$

$$\textit{where} \quad J_f \ = \ \frac{\partial\ \underline{F}^f}{\partial \underline{y}}\ \ , \ \textit{and}\ \ J_s \ = \ \frac{\partial \underline{F}^s}{\partial \underline{y}}\ .$$

The approach taken here follows [4] in neglecting the advective terms J_f and thus in the case when no source or diffusion terms are present corresponds to using functional iteration for the advective calculation, see [3]. In the case when diffusion is absent or sufficiently small to be neglected in the Jacobian matrix, the matrix $I - k\gamma J_s$ is the Jacobian matrix of that part of the o.d.e. system corresponding to the discretization of the time derivatives and the source terms. This matrix is thus block-diagonal with as many blocks as there are spatial elements and with each block having as many rows and columns as there are p.d.e.s. The fact that the block relate only to the chemistry within each cell means that each block's equations may be solved independently.

The nonlinear equations splitting iteration may thus be written as

$$[I - k\gamma J_s] \;\; \underline{\Delta y^*}_m \;\; = \;\; \underline{r}\left(t_{n+1}^m\right) \tag{13}$$

where $\underline{\Delta y^*}_m$ is an approximation to $\underline{\Delta y}_m$. As the splitting is only used to speed up the solution of the nonlinear equations and providing that the iteration is continued until the residual $\underline{r}\left(t_{n+1}^m\right)$ is sufficiently small this splitting error does not have the same impact as introducing splitting at the p.d.e. level. In order for the nonlinear equations splitting iteration defined by equation (13) to converge with a rate of convergence r_c it is necessary, [10] p.77, that

$$||\; [I - k\gamma J_s]^{-1} \; k \; \gamma J_f \; || \;\; = \;\; r_c \qquad \textit{where} \;\; r_c < 1. \tag{14}$$

This condition will also turn out to be important for the IMEX method considered in Section 5.

3.1 Gauss-Seidel Iterations

Recently the Gauss-Seidel method, has been used with great success for atmospheric chemistry problems, [14] . The Jacobian matrix $[I - k\gamma J_s]$ is split into L, the strictly Lower triangular, D, the Diagonal and U the strictly Upper triangular matrices. and equation (13) rearranged to get

$$(I - \gamma k D - \gamma k L)\Delta \underline{y}^*_{m+1} = \gamma k U \Delta \underline{y}^*_m + \underline{r}(t^p_{n+1}). \tag{15}$$

Although one approach is to use a fixed number of iterations, an adaptive approach based on that of DASSL [6] has been used here. A minimum of two iterations is used to estimate the convergence rate, ρ,
where $\rho = \left(\frac{||\underline{\Delta y}^*_{m+1} - \underline{\Delta y}^*_m||}{||\underline{\Delta y}^*_1 - \underline{\Delta y}^*_0||}\right)^{1/m}$ The iteration is continued only if $\rho < 0.95$ and if $||\underline{\Delta y}^*_{m+1} - \underline{\Delta y}^*_m|| <$ ITOL, where ITOL $= 0.01$ or 0.001, then the iteration is halted successfully. The effect of varying ITOL is considered by Verwer [14] and in the next section.

4 Numerical Results

In this section four test problems are used to compare the performance of the NDF2 method with the Theta method and the BDF2 method. The code described here was implemented by at first changing a few constants in a BDF2 code. Testing on standard DETEST problems with analytic solutions confirmed the predicted accuracy improvement was achieved in practise. It was observed that the NDF and BDF codes used more timesteps for atmospheric problems than the Theta method code we had been using [3], but did less work per step. This hypothesis will be explored on three test problems and the effect of using the Gauss Seidel method demonstrated.

(1) This problem [14] consists of 20 species and 25 reactions with constant reaction rates from atmospheric chemistry For this ODE system the Lipschitz constant is about 1.5x10^7 and the simulation period is (0,60) minutes, which makes the ODE system stiff.

(2) Simplified Chemistry [5] only 8 species and 12 reactions photolysis rates time-dependent. The simulation time is 1.810^5 .

(3) This problem with lumped chemistry was obtained from systematic reduction of the Extended Carbon Bond Mechanism CBMEx [7] and has 29 species and 55 reactions with nonconstant reaction rates. The full Extended Carbon Bond Mechanism ,see [7], with 205 reactions (nonconstant reaction rates) and 90 species was also used as a test problem but produces almost identical performance profiles to the lumped version. The simulation time is 1.810^5 .

The following notation is used to present the test results:
Steps = the number of integration steps,
Fun = the number of residual evaluations,
Jac = the number of Jacobian evaluations,
TOL = error tolerance used to define RTOL and ATOL
ATOL = Absolute error tolerance $= \ 10^{-6}TOL$
RTOL = Relative error tolerance = TOL
ITOL = Gauss Seidel Tolerance
G-S = Number of Gauss Seidel iterations
SD_T= The number of significant digits for the maximum relative error at the specified time T, defined by $SD_T = \ -log_{10}\left(max_i\left(\frac{|y_i^n - y_i(T))|}{y_i(T)}\right)\right)$
represents the accuracy of the calculated results, and $y_i(T)$ is the "exact" solution at the specified time as estimated by using a DASSL [6] high accuracy run to approximate the true solution. The Jacobian matrix in all cases is formed directly from the chemistry without residual evaluations.

The numerical results on Problem 1 show that the number of Gauss-Seidel iteration per step are comparable with those in [14] albeit obtained using a

Full	Problem 1			Problem 2			Problem 3		
Matrix	Theta	BDF2	NDF2	Theta	BDF2	NDF2	Theta	BDF2	NDF2
Steps	36	40	39	780	973	1019	744	968	946
Jac	11	14	13	48	69	70	67	77	71
Fun	354	352	337	2418	1074	1115	2370	1126	1083
SD	2.33	1.36	1.66	2.23	2.21	2.28	1.41	1.42	1.43

Table 1
Results with Full Linear Algebra TOL = 0.1

Full	Problem 1			Problem 2			Problem 3		
Matrix	Theta	BDF2	NDF2	Theta	BDF2	NDF2	Theta	BDF2	NDF2
Steps	59	74	64	794	1064	1034	700	983	1069
Jac	20	14	16	46	74	73	60	76	75
Fun	630	399	426	2464	1187	1144	2257	1152	1214
SD	2.53	2.88	2.68	2.42	2.21	2.28	1.43	1.42	1.44

Table 2
Results with Full Linear Algebra TOL = 0.01

Gauss	NDF2	Prob.1		Prob.2		Prob.3	
Seidel	ITOL	0.01	0.001	0.01	0.001	0.01	0.001
TOL	Steps	51	71	991	1019	1034	979
	Jac	17	29	69	68	83	81
0.1	Fun	431	715	1082	1116	1186	1120
	G-S	221	460	2174	2325	2925	3398
	SD	1.99	2.02	2.71	2.14	1.40	1.44
TOL	Steps	71	112	987	1045	1004	1034
	Jac	18	42	73	70	80	93
0.01	Fun	476	1045	1136	1165	1198	1217
	G-S	379	647	2340	2617	3451	4545
	SD	2.71	2.84	2.70	2.7	1.44	1.45

Table 3
Results with Gauss-Seidel Iteration

somewhat different nonlinear Gauss-Seidel method without Aitken Extrapolation. A particular point to note n the results here is that when $ITOL = 0.001$ for Problem 1 then the work increases by 50 % . A comparison between the Theta and NDF2 methods shows that NDF2 uses less function evaluations but takes more steps and Jacobian evaluations. This is probably due to the Theta method's sophisticated error estimator [3] and its double or halving stepsize strategy. It is also worth noting that the Theta codes error estimator requires one extra function evaluation and backsolve per step, and so accounts for the much larger number of function calls. A comparison between Tables 1 and 2 shows that when the Gauss Seidel method is used there are a few more Jacobian evaluations for Problem 3. This seems to arise because of the fact that for Problem 3 19 of 29 equations are not diagonally dominant. Analysis of the Jacobian matrices shows that species 8, the OH radical, destroys the diagonal dominance of a large part of the matrix. Table 2 also shows that there is a significant cost penalty in terms of numbers of iterations associated with using ITOL = 0.001 but that there is also an increase in accuracy.

5 IMplicit EXplicit Multistep Methods

This section will begin to consider if the improved performance of the NDF method over BDF methods carries across to the IMEX approach used by Ascher et al. and to characterise the relationship between the IMEX approach and the nonlinear equations splitting method described above.

The approach of Ascher , [1] , effectively replaces the nonstiff part of the o.d.e with an explicit method e.g. for the NDF2 in equation (9) the term

$$\underline{F}(t_{n+1}, \underline{y}_{n+1}) \; = \; \underline{F}^f(t_{n+1}, \underline{y}_{n+1}) \; + \; \underline{F}^s(t_{n+1}, \underline{y}_{n+1})$$

is replaced by

$$2\underline{F}^f(t_n, \underline{y}_n) \; - \; \underline{F}^f(t_{n-1}, \underline{y}_{n-1}) \; + \; \underline{F}^s(t_{n+1}, \underline{y}_{n+1})$$

or alternatively, see [9], by a form which is equivalent for linear F^s, and which gives the NDF IMEX method considered here:

$$\underline{F}^f(t_{n+1}, \underline{y}^*_{n+1}) + \underline{F}^s(t_{n+1}, \hat{\underline{y}}_{n+1}) - \frac{1}{6k}(10\hat{\underline{y}}_{n+1} - 15\underline{y}_n + 6\underline{y}_{n-1} - \underline{y}_{n-2}) = 0 \tag{16}$$

where $\underline{y}^*_{n+1} \; = \; 2\underline{y}_n - \underline{y}_{n-1}$ and $\hat{\underline{y}}_{n+1}$ is the solution computed by this method at the end of the step. The Newton iteration is identical to that defined by equation (13) except that the residual on the righthand side is now defined by

$$\hat{\underline{r}}\,(t^p_{n+1}) \;\; = \;\; -\,\hat{\underline{y}}^p_{n+1} + \underline{z}_n + \gamma k \underline{F}^f(t_{n+1}, \underline{y}^*_{n+1}) + \gamma k \underline{F}^s(t_{n+1}, \hat{\underline{y}}^p_{n+1}) \; .$$

In this case a crude approximation to the norm of the inverse iteration matrix, [12], is provided by the observed rate of convergence r_c^{imax}

$$|| \ [I - k\gamma J_s]^{-1} \ || \quad \approx \ r_c^{imex} \qquad where \ \ r_c^{imex} < 1. \tag{17}$$

It is worth remarking that the cost of the first IMEX iteration is identical to that of one iteration of the splitting method in Section 3, but thereafter the term $\underline{F}^f(t_{n+1}, \underline{y}^*_{n+1})$ does not have to be evaluated.

5.1 The Test Equation of Ascher et al. [1]

Ascher et al. [1] consider the test equation

$$\dot{y} = (\alpha + \ i\beta)y \ , \quad \alpha, \beta \ \ real, \tag{18}$$

which is derived from Fourier analysis of the advection diffusion equation. In this equation β models the advective terms and α the diffusive terms. In this case the BDF2 characteristic polynomial ($\hat{\alpha} = 0$) is

$$\phi(z) = (\frac{3}{2} - \alpha k)z^2 - (2 + 2i\beta k)z + (\frac{1}{2} + i\beta k).$$

while the NDF2 characteristic polynomial is

$$\phi(z) = (\frac{10}{6} - \alpha k)z^3 - (\frac{5}{2} + 2 \ i\beta k)z^2 + (1 + i\beta k)z - \frac{1}{6}. \tag{19}$$

The stability contours of this polynomial are shown in Figure 2 in which the horizontal axis is α and the vertical axis β . A comparison between Figure 2 and Figure 5 in [1] shows that the NDF2 method is stable for purely imaginary eigenvalues, unlike BDF2.

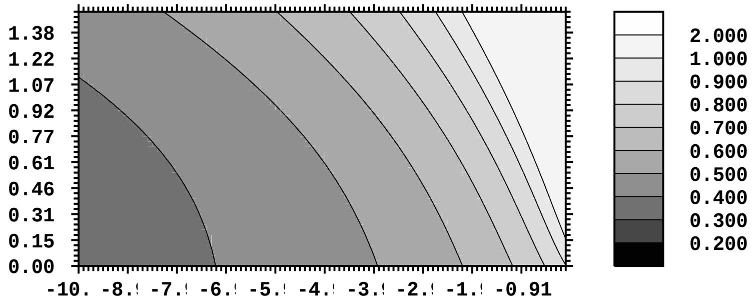

Fig. 2. IMEX Stability Region

In the case when the nonlinear equations splitting method of Section 3 is applied to the same model equation (18) used by Ascher , then from equation (14), the convergence condition is

$$| \ [I - k\gamma\alpha]^{-1} \ k \ \gamma i\beta \ | \quad < \ r_c \ .$$

and so the method is stable for purely imaginary $k\beta$ in the range $[0, r_c/(\gamma k)]$.

5.2 The Extended Test Equation of Frank et al. [9]

Frank et al. [9] point out that the stability decomposition used by Ascher needs to be generalised to model some aspects of the atmospheric diffusion equation. This may be achieved by allowing α and β in equation (18) to be both complex. There are then two situations considered by Frank et al. The first is to consider values of β for which the method is A-stable with respect to alpha. The second option , [9], is to recognise that while A-stability is valuable; it is, in many practical situations, possible to settle for A(α) stability. With this in mind the latter case is considered here in that β is forced to lie in the stability region of the explicit NDF2 method given by:

$$\underline{F}^f(t_{n+1}, \underline{y}^*_{n+1}) \ - \ \frac{1}{6k}(10\underline{\hat{y}}_{j+1} - 15\underline{y}_n + 6\underline{y}_{n-1} - \underline{y}_{n-2}) \ = \ 0 \qquad (20)$$

and a similarly modified BDF2 method. In these two cases the stability regions are the interiors of the semi-circular domains shown in Figure 3 and the maximum possible values of $|\beta k| \leq 1.775$ for NDF2 and ≤ 1.3 for BDF2. The NDF2 stability polynomial in this case is given by extending α and $i\beta$ in equation (18) to have both real and imaginary parts. Figure 4 shows the boundaries of the stability region for NDF2 ($\hat{\alpha} = -1/9$) and for BDF2 ($\hat{\alpha} = 0$) in the case when β may take any of the stable values in Figure 3.

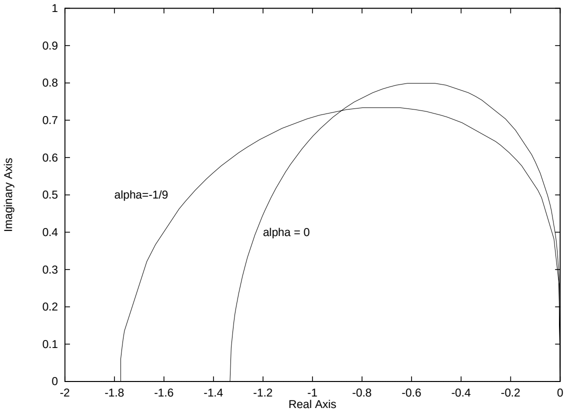

Fig. 3. Explicit Stability Region

The stability regions in Figure 4 are the exteriors of the semi-circular regions. Figure 4 shows that NDF2 has a desirably smaller stability region than BDF2 in the right half plane but a less desirable slightly smaller one in the left half plane than BDF2. Both methods are A(α) stable however.

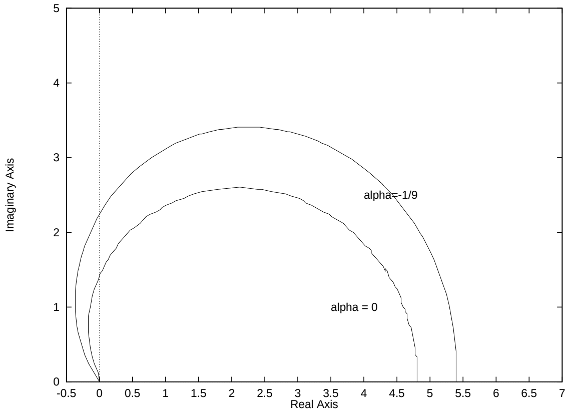

Fig. 4. Modified Stability Region

The convergence condition of the nonlinear equations splitting method of Section 3 for the model equation (18) used by [9] in which both α and β are complex is, from equation (14),

$$\mid [I - k\gamma\alpha]^{-1} \mid \; < \; \frac{r_c}{\gamma \; k|\beta|} \; .$$

The maximum values for $k|\beta|$ to remain in the stability region are 1.3 and 1.8 for BDF2 and NDF2 respectively and $\gamma = 2/3$ and $\gamma^* = 0.6$. Hence if $k\beta$ lies on the edge of the explict stability region then $\gamma \; k|\beta| \; \approx \; 1$ and this requirement is very similar to the IMEX convergence requirement, which from equation (17) is,

$$\mid [I - k\gamma\alpha]^{-1} \mid \; < \; r_c^{imex} \; . \tag{21}$$

Hence in both cases αk must satisfy a very similar condition if the iterations are to be stable and to converge at the same rate.

5.3 Estimating the local splitting error.

In order to understand the form of error introduced by using the IMEX method let $\underline{e\hat{y}}_{n+1} = \left[\underline{y}_{n+1} - \underline{\hat{y}}_{n+1}\right]$ be the local IMEX splitting error where $\underline{\hat{y}}_{n+1}$ is the value computed by the IMEX method defined by equation (16) and let $\underline{y}_{n+1}$ be the approximation to the solution at t_{n+1} defined as in equation (9). Assuming that the the past values $\underline{y}_j, j = n, n-1, n-2$ for both methods are identical, subtracting equation (16) from equation (9) , multiplying by γk and linearising gives

$$[I - k\gamma J_s] \ \underline{e\hat{y}}_{n+1} \ = \ \gamma k \left[\underline{F}^f(t_n, \underline{y}_{n+1}) \ - \ \underline{F}^f(t_{n-1}, \underline{y}^*_{n+1}) \right] . \qquad (22)$$

Addition and subtraction of the term $\underline{F}^f(t_{n+1}, \underline{\hat{y}}_{n+1})$ and further linearisation gives (after ignoring the higher order terms):

$$[I - k\gamma J_s] \ \underline{e\hat{y}}_{n+1} = \gamma k \left[J_f(\underline{e\hat{y}}_{n+1}) + \underline{F}^f(t_{n+1}, \underline{\hat{y}}_{n+1}) \ - \ \underline{F}^f(t_{n+1}, \underline{y}^*_{n+1}) \right] (23)$$

Applying the inverse matrix $[I - k\gamma J_s]^{-1}$ and assuming that equation (14) holds gives:

$$||\underline{e\hat{y}}_{n+1}|| \ \leq \ r_c ||\underline{e\hat{y}}_{n+1}|| \ + \\ ||[I - k\gamma J_s]^{-1} \ \gamma k \left[\underline{F}^f(t_{n+1}, \underline{\hat{y}}_{n+1}) - \underline{F}^f(t_{n+1}, \underline{y}^*_{n+1}) \right] || \ .$$

On collecting terms together this may be written as:

$$||\underline{e\hat{y}}_{n+1}|| \ \leq \ \frac{\gamma k}{1 - r_c} ||[I - k\gamma J_s]^{-1} \left[\underline{F}^f(t_{n+1}, \underline{\hat{y}}_{n+1}) - \underline{F}^f(t_{n+1}, \underline{y}^*_{n+1}) \right] || \ .$$

The righthand term $||.||$ may be calculated using one extra evaluation of $\underline{F}^f(t_{n+1}, \underline{\hat{y}}_{n+1})$ and a backsolve, (the value of r_c is not known however). Equation (17) may be used to bound the norm of the inverse Jacobian $||[I - k\gamma J_s]^{-1}||$ by the rate of convergence r_c^{imex} to get

$$||\underline{y}_{n+1} - \underline{\hat{y}}_{n+1}|| \ \leq \ \frac{r_c^{imex} \gamma k}{1 - r_c} \ || \left[\underline{F}^f(t_{n+1}, \underline{\hat{y}}_{n+1}) \ - \ \underline{F}^f(t_{n+1}, \underline{y}^*_{n+1}) \right] || \ . \quad (24)$$

The righthand term $\gamma k||.||$ is related to the extra local truncation error due to splitting, see [8], and may be written as:

$$-k \left[\ \underline{F}^f(t_{n+1}, \underline{\hat{y}}_{n+1}) - \underline{F}^f(t_{n+1}, \underline{y}^*_{n+1}) \right] \ = \ - k J_f(\underline{\hat{y}}_{n+1} - \underline{y}^*_n) \ + \ h.o.t.$$

where $\underline{\hat{y}}_{n+1} - \underline{y}^*_{n+1} \ = \ (\underline{\hat{y}}_{n+1} - 2\underline{y}_n + \underline{y}_{n-1})$ may be interpreted as an $O(k^2)$ error. Alternatively by using equation (14) again it is possible to derive the expression

$$||\underline{e\hat{y}}_{n+1}|| \ \leq \ \frac{r_c}{1 - r_c} ||\underline{\hat{y}}_{n+1} - \underline{y}^*_{n+1}|| \ .$$

In both cases the quantity r_c plays an important role in the relation ship between the IMEX splitting error and the quantity $\underline{\hat{y}}_{n+1} - \underline{y}^*_{n+1}$. As the restriction on r_c is also required for convergence of the nonlinear splitting method, this restriction thus appears to be important to both methods for different reasons.

6 Conclusions

This paper has shown that the NDF2 method has slightly superior accuracy and stability properties to the more widely used BDF2 method for the type of o.d.e. systems considered here. A comparison between the IMEX and nonlinear splitting approaches has shown some interesting similarities and has indicated a way of measuring the IMEX splitting error. The advantage of the splitting approach -that it more closely couples the flow and chemistry- is balanced by its greater cost after the first two iterations, (if the local splitting error estimation is included). The close coupling of flow and chemistry may be more important for combustion type problems e.g. [4] than for atmospheric problems. The numerical experiments have shown that the new NDF2 code works well but have also indicated that some tuning of the stepsize strategy and Jacobian evaluation criteria may be needed. At present the approach described here is already being used successfully in large scale experiments in computational atmospheric modelling.

Acknowledgement. The authors would like to thank their many collaborators on the atmospheric modelling project for their help, notably A.Tomlin and G.Hart. W. Hunsdorfer and L.F. Shampine are thanked for supplying pre-publication versions of their papers.

References

[1] U.M.Ascher, S.J.Ruuth and B.T.R.Wetton. Implicit-explicit methods for time-dependent p.d.e.s. *SIAM J.Numer. Anal.*, Vol. 32, No 3 pp797-823, June 1995.

[2] M. Berzins, A C^1 Interpolant for codes based on Backward Differentiation Formulae, *Appl.Numer. Math.* 2 (1986) 109-118.

[3] M. Berzins and R.M. Furzeland. An adaptive theta method for the solution of stiff and non-stiff differential equations. *Appl. Numer. Math.* **9** (1992), 1–19.

[4] M Berzins and J M Ware. Solving convection and convection reaction problems using the M.O.L. *Appl. Numer. Math.* , 20:83–99, 1996.

[5] M Berzins, P H Gaskell, A Sleigh, W Speares, A Tomlin, and J M Ware. An adaptive CFD solver for time dependent environmental flow problems.

pp.311-318 in *Numerical Methods for Fluid Dynamics V*, Eds K.W.Morton and M.J.Baines, Clarendon Press, Oxford 1995.

[6] K. Brenan, S. Campbell and L.R. Petzold, *Numerical Solution of Initial Value Problems in Differential-Algebraic Equations* ,SIAM , Philadelphia 1996.

[7] A.C. Heard, M.J. Pilling, A.S. Tomlin Mechanism reduction techniques applied to tropospheric chemistry. in press *Atmospheric Environment* (1997)

[8] W.Hunsdorfer and J.G. Verwer. A note on splitting errors for advection reaction equations. *Appl. Numer. Math.* 18:191-199 (1995),

[9] J.Frank, W.Hunsdorfer and J.G. Verwer. Stability of implicit-explicit linear multistep methods. submitted to *Appl. Numer. Math.* (1997),

[10] C.T. Kelly, *Iterative Methods for Linear and Nonlinear Equations* . SIAM, Philadelphia, 1995.

[11] R. W. Klopfenstein. Numerical Differentiation Formulae for stiff systems of o. d. e.s , *RCA Review* Vol. 32, pp447-462, September 1971.

[12] L. F. Shampine, Type-insensitive ode codes based on implicit A-stable formulas, *Math. of Comp.* , Vol. 36, April 1981, 499-510.

[13] L. F. Shampine and M.W. Reichelt, The Matlab ODE Suite, *SIAM J. on Sci. Comput* To appear .

[14] J.G. Verwer, Gauss-Seidel iteration for stiff odes from chemical kinetics, *SIAM J. on Sci. Comput* Vol.15 1243-1250, 1994.