Journal of Engineering and Applied Sciences 14 (24): 9538-9543, 2019
ISSN: 1816-949X

Solving Index-1 Semi Explicit System of Differential Algebraic Equations by Mix-Multistep Method

[1]Yong Faezah Rahim, [2]Mohamed Suleiman and [3]Zarina Bibi Ibrahim
[1]Center of Foundation Studies for Agriculture Science,
[2]Institute For Mathematical Research (INSPEM),
[3]Faculty of Science, University Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia
yfaezah@upm.edu.my

Abstract: Differential Algebraic Equations (DAEs) are regarded as stiff Ordinary Differential Equations (ODEs), therefore, they are solved using implicit method such as Backward Differential Formula (BDF) type of method and require the use of Newton iteration which usually requires a lot of computational effort. However, not all of the ODEs in the DAE system are stiff. In this study, we describe a new technique for solving index-1 semi explicit system of DAE where the ODEs are treated as non-stiff at the start of the integration and putting the non-stiff ODE's into the stiff subsystem should instability occurs. Adams type of method is used to solve the non-stiff part and BDF method for the stiff part. This strategy is shown to be competitive in terms of computational effort and accuracy. Some numerical experiments are presented to illustrate the effectiveness.

Key words: Differential algebraic equation, multistep-method, stiff ODE, numerical experiments, computational effort, accuracy

INTRODUCTION

The dynamical behaviour of physical process is usually modelled in several equations. But if the states of the physical system have constrain, for example by conservation laws such as Kirchoff's laws in electrical networks or by position constraints such as movements of mass points on a surface, then, the mathematical model also contains algebraic equations to describes these such constrains. Such systems, consisting of both differential and algebraic equations are called Differential Algebraic Equations (DAE).

In recent years, much research has been focused on the numerical solution of DAEs. Some numerical methods have been developed using BDF (Gear, 1971; Petzold, 1982), implicit Runge-Kutta method (Ascher and Petzold, 1991), Pade approximation method (Celik *et al*., 2006) and Adomain Decomposition method (Celik *et al*., 2006). Following the study by Gear (1971), several codes implementing the BDF methods were written. The most widely used production code is the code DASSL of Petzold (Brenan *et al*., 1989). The code is designed to be used for the solution of the implicit form F(t, y, y') = 0 with index zero and one. The code LSODI, developed by Hindmarsh (1998) is written for linearly implicit DAEs of the form A(t, y) y' = f (t, y). The SPRINT code developed by Berzins *et al*. (1989), also employs BDF method for the solution of DAEs. Still not all DAEs were solved successfully with these codes. Each code has its own restriction. For example, code DASSL has difficulties to distinguish between a failure due to inconsistent initial conditions and one due to a higher formulation. Code LSODI, the user must supply a subroutine for evaluating the matrix A times a vector. The general form of DAEs is given by:

$$F(t, y, y') = 0$$

With consistent initial values:

$$y(x_0) = y_0, \qquad y'(x_0) = y_1 \tag{1}$$

When ∂F/∂y' is non-singular (Eq. 1) is an ODEs. The existence of algebraic constraints on the variable is expressed by the singularity of the Jacobian matrix ∂F/∂y'. They are more difficult to handle that ODEs due to the existence of algebraic equations. The algebraic constraints may appear explicitly as in the system:

Corresponding Author: Yong Faezah Rahim, Center of Foundation Studies for Agriculture Science, University Putra Malaysia (UPM) 43400 Serdang, Selangor, Malaysia, yfaezah@upm.edu.my

$$\begin{aligned} y' &= f(t, y, z) \\ 0 &= l(t, y, z) \end{aligned} \quad (2)$$

The system, Eq. 2 is called a semi-explicit system of DAEs. A system of DAEs is characterized by its index which is the number of differentiations required to convert it into a system of ODEs. Here, the Jacobian matrix l_y is assumed to be non-singular for all t, therefore, the system has index one. Since, most DAEs arising in the applications are in semi-explicit form, then, the technique to be developed is for solving DAEs in this form.

In solution of ODEs, partitioning of the equations into stiff and non-stiff subsystems has been very successful in terms of computational effort (Suleiman, 1979; Suleiman and Baok, 1992). Some of the earlier works on partitioning are given by Enright and Kamel. Watkins and Hansonsmith developed a precise partitioning method proposed by Enright and Kamel but differ in the partition of the Jacobian matrix. Other research on partitioning are discussed by Hall *et al.* DAEs consist of algebraic equations which are treated as stiff (Wanner and Hairer, 1991). However, the same cannot be said of the ODEs counterpart which consist of non-stiff and stiff subsystems that can be treated by non-stiff and stiff methods, respectively. Consider the regularization of DAEs (Eq. 2) where it is replaced by the ODE:

$$ez' = l(x, y, z) \quad (3)$$

which depends on a small parameter $0 \le \varepsilon \le 1$. This means Eq. 3 is very stiff. It is natural to consider methods for stiff ODE for the solution of the DAEs in Eq. 2. If, we now let $\varepsilon \to 0$, we obtain the reduced (Eq. 2). For the ODEs Eq. 2, the systems can be partitioned into stiff and non-stiff parts to reduce the cost of iteration scheme. These reasons motivate us to look into the partitioning of Eq. 2. Therefore, in this study we look into partitioned system, so that, substantial saving can be gained if this part is done efficiently.

MATERIALS AND METHODS

The propose method for this solution of DAEs given in Eq. 2 requires a modification of the integration formulae by Suleiman (Brenan *et al*., 1989). The formulae are based on storing back values of $y^{(d-j)}$ where j takes one of the values 0, 1, ..., d and d is the order of the equations in this study we only discuss on equations with d = 1. The case of j = 0 will refer to generalisation of Adams method to solve non-stiff problems where a predictor-corrector scheme using the simple iteration is appropriate. The case j>0 is for stiff problems and the solution set at each integration step using Newton type of iteration is required.

Most partitioning strategies start by treating the ODEs system as non-stiff and solve using Adams method. Once there is an indication of instability due to stiffness, then, the whole system is treated as stiff and solve using BDF method. As we mentioned earlier, the solution of large stiff system requires expensive iteration process. Therefore, if the equations requires stiffness can be identified, there will be significant reduction in the computational cost where only the relevant equations that cause instability are placed in the stiff subsystem. At the first instance of instability which is due to the eigenvalues of the largest and almost equal in the magnitude of the Jacobian of the system, the appropriate equation is placed in the stiff subsystem and solved using BDF method.

In doing so, the effect of these eigenvalues is nullified and larger stepsizes are permissible until the effect of the next set of the largest and almost equal in magnitude of the eigenvalues cause instability. Hence, again the appropriate equations are placed in the stiff subsystem and the process continues. In general, problem may be stiff in some intervals and non-stiff in others. Therefore, this technique also allows us switching from stiff to non-stiff when necessary. This type of partitioning is called dynamic or componentwise partitioning. Consider a system of first order ODEs of the form:

$$y' = f(x, y) \quad y(a) = \eta \quad y \in \Re^s \quad (4)$$

Let, the first w equations be non-stiff and the next (s-w) equations be stiff. Then the iterative equations corresponding to the solution of Eq. 4 given by Suleiman and Baok (1992) are:

$$^{i+1}e_t = \alpha \sum_{j=1}^{w} \frac{\partial f_t}{\partial y_j} {}^{i}e_j + \sum_{j=w+1}^{s} \frac{\partial f_t}{\partial y_i} {}^{i+1}e_j \qquad t = 1, ..., w \quad (5)$$

$$\frac{y_r}{h} {}^{i+1}e_j - \sum_{j=w+1}^{N} \frac{\partial f_t}{\partial y_i} {}^{i+1}e_j = \alpha \sum_{j=1}^{w} \frac{\partial f_t}{\partial y_j} {}^{i}e_j \qquad t = 1, ..., s \quad (6)$$

where, ${}^{I+1}e_t = {}^{i+1}y_t - {}^{i}y_t$ is the increment at (i+1)th iteration of the tth equation. We define:

$$\alpha = \begin{cases} 1 & \text{If the component stiff} \\ h\beta_{k_p} & \text{If the component non-stiff} \end{cases}$$

$\alpha_j \frac{\partial f_t}{\partial y_j} {}^i e_j$ is referred to as the perturbation term in the tth equation due to jth the component of the system. It is the growth of this perturbation that causes instability and is the main identification for partitioning. If there is an indication of instability because of LTE>TOL or non-convergence, then, the presence of stiffness is suspected. Once stiffness is detected, the component that is associated with the largest perturbation which is the cause of stiffness is identified. Note that stiffness may occur in the equation already stiff due to coupling to a non-stiff component. Suppose the jth equation fails the error test, then (Algorithm 1):

Algorithm 1; Equation fails the error test

1. Let n = 0, m = j (variable n counts the number of equations, m identifies the questions)
2. Set n = n+1
3. Set A[n] = m
4. If (mth component is not stiff) then the most dominant term on the RHS is:

$${}^2e_m = \alpha_p \frac{\partial f_m}{\partial y_p} {}^1e_p$$

(if component p is non-stiff)

or

$${}^2e_m = \frac{\partial f_m}{\partial y_p} {}^1e_p$$

(if component p is stiff)

If ($|{}^2e_m| < |c_1 {}^1e_m|$), c_1 is a suitable constant, then

Goto 8 (exit)

else (mth component is stiff) then
if (n = 1) then
a) find the most dominant term on the LHS of Eq. 6:

$$\theta_{mq} = \left(\frac{\gamma_m}{h} - \frac{\partial f_m}{\partial y_m}\right) {}^2e_m \quad (\text{if } q = m)$$

$$\theta_{mq} = -\frac{\partial f_m}{\partial y_q} {}^2e_q \quad (\text{if } q \neq m)$$

b) set A [n] = q
c) find the most dominant term on the RHS of Eq. 6:

$$\alpha_m \frac{\partial f_m}{\partial y_p} {}^1e_p$$

d) if $|\theta_{mq}| < C\left|\alpha_q \frac{\partial f_q}{\partial y_p} {}^1e_p\right|$ then

goto 8 (exit)
else
a) find q such that the most dominant term on the LHS of Eq. 6:

i. $\theta_{qm} = \left(\frac{\gamma_q}{h} - \frac{\partial f_q}{\partial y_q}\right) {}^2e_q \quad (\text{if } q = m)$

ii. $\theta_{qm} = -\frac{\partial f_q}{\partial y_m} {}^2e_m \quad (\text{if } q \neq m)$

b) set A [n] = q
c) find the most dominant term on the RHS of Eq. 6:

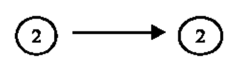
$$\alpha_q \frac{\partial f_q}{\partial y_p} {}^1e_p$$

d) if $|\theta_{qm}| < C\left|\alpha_q \frac{\partial f_q}{\partial y_p} {}^1e_p\right|$ then

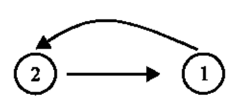
goto 8 (exit)
end if
end if

5. For s = 1 to n
If (A [s] = p) then
a) for d = s to n
if component *d* is not stiff, change to stiff
end if
b) go to 8 (exit)
6. m = p
7. goto 2
8. RETURN

Illustration of the partitioning process: Example: system of two equations (both are treated non-stiff initially):

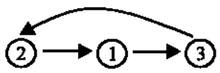


The first instability occur on Eq. 2. It shows that the term due to Eq. 2 dominates on the RHS of Eq. 5 meaning the instability in Eq. 2 was due to the term of Eq. 2 while due to Eq. 1 is almost negligible. Therefore, Eq. 2 is changed to stiff while Eq. 1 remain as non-stiff:

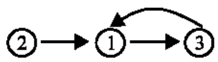


The term due to Eq. 1 dominates on RHS of Eq. 6, meaning the instability in Eq. 2 was due to Eq. 1. From Eq. 1, the dominant term is Eq. 2, meaning that the perturbation in Eq. 1 was due to the term from Eq. 2. Hence, there is coupling effect on Eq. 1 and 2. Therefore, both equation will change to stiff.

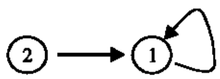
System of three equations (mix method): Examples: Eq. 1-3 become stiff:



Equation 1 and 3 become stiff:



Equation 1 become stiff:



We now consider the iteration process of solving the system (Eq. 2) using the Adams PECE and BDF method on non-stiff and stiff equations, respectively. Initially, the ODEs (Eq. 2) are treated as non-stiff and solved with a Variable Stepsize Variable Order Componentwise (VSVOC) Adam method. This is referred as case j = 0. The predictor:

$$^{p}y_{n+1} = y_n + \sum_{i=0}^{k-1} g_{i,1} f_{[n, n-1, \ldots, n-i]}$$

$$^{p}y'_{n+1} = y'_n + \sum_{i=0}^{k-1} g_{i,0} f_{[n, n-1, \ldots, n-i]}$$

And:

$$^{p}z_{n+1} = z_n + \sum_{i=0}^{k-1} g_{i,1} l_{[n, n-1, \ldots, n-i]}$$

Where:

$$g_{i,1} = \int_{x_n}^{x_{n+1}} (x-x_n), \ldots, (x-x_{n-i+1})dx$$

$f_{[n, n-1, \ldots, n-i]}$, $l_{[n, n-1, \ldots, n-i]}$-the ith divided difference

The corrector:

$$^{c}y_{n+1} = {}^{p}y_{n+1} + \frac{g_{k,1}}{g_{k,0}} e$$

where:

$$e = f\left({}^{p}y_{n+1}, {}^{p}z_{n+1}\right) - {}^{p}y'_{n+1}$$

Case j = 1 is introduced for stiff problems and the solution of a set of in general, nonlinear equations, a Newton type iteration is required at each integration step. For simplicity, we combine the solution for Eq. 2. The formulas are given below. The predictor:

$$^{p}y_{n+1} = \sum_{i=0}^{k-1} g_{i,1} F_{[n, n-1, \ldots, n-i]}$$

where:

$F_{[n, n-1, \ldots, n-i]}$-the ith divided difference

$F \equiv y(x)$

$$^{p}z_{n+1} = \sum_{i=0}^{k-1} g_{i,1} F_{[n, n-1, \ldots, n-i]}$$

where:

$F_{[n, n-1, \ldots, n-i]}$-the ith divided difference

$F \equiv l(x)$

$$^{p}z'_{n+1} = \sum_{i=0}^{k-1} d_{i,1} F_{[n, n-1, \ldots, n-i]}$$

where:

$$d_{i,1} = \frac{d}{dx}(x-x_n)(x-x_{n-1}), \ldots, (x-x_{n-i+1})\Big|_{x=x_{n+1}}$$

The corrector, for j = 1 (Eq. 2) is solved using modified Newton iteration where two corrector iterations are used. The notation i is introduced for specifying the iteration:

$$f\left({}^{i+1}y_{n+1}, {}^{i+1}z_{n+1}\right) = {}^{i}y'_{n+1} + \frac{d_{k-1,1}}{g_{k-1,0}} {}^{i+1}e_y$$

$$f\left({}^{i}y_{n+1} + {}^{i+1}e_y, {}^{i}z_{n+1} + {}^{i+1}e_z\right) = {}^{i}y'_{n+1} + \frac{d_{k-1,1}}{g_{k-1,0}} {}^{i+1}e_y$$

Expanding $f({}^{i}y_{n+1} + {}^{i+1}e_y, {}^{i}z_{n+1} + {}^{i+1}e_z)$ by Taylor expansion:

$$f\left({}^{i+1}y_{n+1}, {}^{i+1}z_{n+1}\right) + {}^{i+1}e_y \frac{\partial f}{\partial y} + {}^{i+1}e_z \frac{\partial f}{\partial z} = {}^{1}y'_{n+1} - f\left({}^{i}y_{n+1}, {}^{i}z_{n+1}\right)$$

$$^{i+1}e_y \frac{\partial f}{\partial y} - \frac{d_{k-1,1}}{g_{k-1,0}} {}^{i+1}e_y + {}^{i+1}e_z \frac{\partial f}{\partial z} = {}^{1}y'_{n+1} - f\left({}^{i}y_{n+1}, {}^{i}z_{n+1}\right) \quad (7)$$

$$l\left({}^{i+1}y_{n+1}, {}^{i+1}z_{n+1}\right) = 0$$

$$l\left({}^{i}y_{n+1} + {}^{i+1}e_y, {}^{i}z_{n+1} + {}^{i+1}e_z\right) = 0$$

Expanding $l({}^{i}y_{n+1} + {}^{i+1}e_y, {}^{i}z_{n+1} + {}^{i+1}e_z)$ by Taylor expansion:

$$l\left({}^{i}y_{n+1}, {}^{i}z_{n+1}\right) + {}^{i+1}e_y \frac{\partial l}{\partial y} + {}^{i+1}e_z \frac{\partial l}{\partial z} = 0$$

$$^{i+1}e_y \frac{\partial l}{\partial y} + {}^{i+1}e_z \frac{\partial l}{\partial z} = -l\left({}^{i}y_{n+1}, {}^{i}z_{n+1}\right) \quad (8)$$

From Eq. 7 and 8, this then allows us to construct an iterative scheme for the BDF method. At iteration I:

$$\begin{bmatrix} \frac{\partial f}{\partial y} - \frac{d_{k-1,1}}{g_{k-1,0}} & \frac{\partial f}{\partial z} \\ \frac{\partial l}{\partial y} & \frac{\partial l}{\partial z} \end{bmatrix} \begin{bmatrix} {}^{i+1}e_y \\ {}^{i+1}e_z \end{bmatrix} = \begin{bmatrix} {}^{i}y'_{n+1} - f\left({}^{i}y_{n+1}, {}^{i}z_{n+1}\right) \\ -l\left({}^{i}y_{n+1}, {}^{i}z_{n+1}\right) \end{bmatrix}$$

The two stage iteration process proceeds as follows. (A is the Jacobian Matrix) Solve for ${}^{1}e$ from:

$$A\,{}^{1}e_y = {}^{p}y'_{n+1} - f\left({}^{p}y_{n+1}, {}^{p}z_{n+1}\right)$$
$$A\,{}^{1}e_z = -l\left({}^{p}y_{n+1}, {}^{p}z_{n+1}\right)$$

Compute the corrected values:

$${}^{1}y_{n+1} = {}^{p}y_{n+1} + {}^{1}e_y$$
$${}^{1}z_{n+1} = {}^{p}z_{n+1} + {}^{1}e_z$$

Solve for e^2 from:

$$A\,{}^{2}e_y = {}^{1}y'_{n+1} + \frac{d_{k-1,1}}{g_{k-1,0}}\,{}^{1}e - f\left({}^{1}y_{n+1}, {}^{1}z_{n+1}\right)$$
$$A\,{}^{2}e_z = -l\left({}^{1}y_{n+1}, {}^{1}z_{n+1}\right)$$

Compute the final corrected values:

$${}^{c}y_{n+1} = {}^{1}y_{n+1} + {}^{2}e_y$$
$${}^{c}z_{n+1} = {}^{1}z_{n+1} + {}^{2}e_z$$

RESULTS AND DISCUSSION

Numerical: The test problem considers the following DAEs (Table 1 and 2):

$$\begin{aligned} y' &= t\cos t - y + (1+t)z \qquad & y(0) = 1, \quad z(0) = 0 \\ 0 &= \sin t - z & 0 \le t \le 10 \end{aligned}$$

$$\text{Exact solution:} \quad y(t) = \exp(-t) + t\sin t, \qquad z(t) = \sin(t)$$

$$\begin{aligned} y_1' &= -ty_2 - (1+t)z_1 \qquad & y_1(0) = 1, \quad y_2(0) = 1 \\ y_2' &= -ty_1 - (1+t)z_2 & z_1(0) = -1, \quad z_2(0) = 1 \\ 0 &= (y_1 - z_1)/5 - \cos(t^2/2) & 0 \le t \le 10 \\ 0 &= (y_1 + z_1)/5 - \sin(t^2/2) & 0 \le t \le 10 \end{aligned}$$

$$\text{Exact solution:} \quad \begin{aligned} y_1(t) &= \sin(t) + 5\cos(t^2/2) \\ y_2(t) &= \cos(t) + 5\sin(t^2/2) \\ z_1(t) &= -\cos(t) \\ z_2(t) &= \sin(t) \end{aligned}$$

Notations used in Table 1 and 2:

- TOL-Tolerance
- MAXERR-Maximum Error
- SS-Success Step
- FS: Failure Step
- TS: Total Step

Table 1: Notation (1)

	10 (-2)		10 (-3)		10 (-4)	
Tol	Partitioning	Without	Partitioning	Without	Partitioning	Without
MAXERR	1.55935(-2)	4.84960(-2)	1.70125(-3)	1.04648(-2)	6.24365(-5)	2.46260(-3)
SS	25	30	33	38	53	62
FS	3	7	5	3	6	6
TS	28	37	38	41	59	68

Table 2: Notation (2)

	10 (-2)		10 (-3)		10 (-4)	
Tol	Partitioning	Without	Partitioning	Without	Partitioning	Without
MAXERR	9.24758 (-3)	2.73121 (-1)	5.03043 (-4)	2.5821 (-2)	9.71525 (-5)	2.51579 (-3)
SS	156	102	161	152	159	231
FS	2	5	2	2	2	4
TS	158	107	163	154	161	235

CONCLUSION

We noted that with the same TOL, the global error is smaller and the number of steps also reduced for the partitioning case compared to non-partition case. For problem 2 while in lower TOL despite partitioning case needs more number of steps but its accuracy are much better compared to non-partition case and with higher TOL partitioning technique increase the performance by giving smaller number of steps and continue increasing the accuracy. In conclusion, we have demonstrated that it is favourable to partitioning DAE system into non-stiff and stiff subsystem rather to treat the system as a stiff system to all equations. The approach (using partitioning) is effective in term of computational effort, since, the Jacobian has smaller dimension, hence, requires less number of matrix operation in order to evaluate the Jacobian matrix and also increasing the accuracy.

ACKNOWLEDGEMENT

This research was supported by University Putra Malaysia under Fundamental Research Grant Scheme (FRGS).

REFERENCES

Ascher, U.M. and L.R. Petzold, 1991. Projected implicit runge-kutta methods for differential-algebraic equations. SIAM. J. Numer. Anal., 28: 1097-1120.

Berzins, M., P.M. Dew and R.M. Furzeland, 1989. Developing software for time-dependent problems using the method of lines and differential-algebraic integrators. Appl. Numer. Math., 5: 375-397.

Brenan, K.E., S.L. Campbell and L.R. Petzold, 1989. Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. Elsevier, Amsterdam, Netherlands, ISBN:9780444015112, Pages: 210.

Celik, E., M. Bayram and T. Yeloglu, 2006. Solution of Differential-Algebraic Equations (DAEs) by adomian decomposition method. Intl. J. Pure Appl. Math. Sci., 3: 93-100.

Gear, C., 1971. The simultaneous numerical solution of differential-algebraic equations. IEEE Trans. Circuit Theory, 18: 89-95.

Hindmarsh, A.C., 1998. A fortron code for the numerical dynamic simulation of solid/fluid reaction to non-isothermal multispecies porous spheres with Stefan-Maxwell diffusion. Lawrence Livermore National Laboratory, Livermore, California.

Suleiman, M. and S. Baok, 1992. Using nonconvergence of iteration to partition ODEs. Appl. Math. Comput., 49: 111-140.

Suleiman, M.B., 1979. Generalized multistep Adams and backward differentiation methods for the solution of stiff and non-stiff ordinary differential equations. Ph.D Thesis, University of Manchester, Manchester, England, UK.

Wanner, G. and E. Hairer, 1991. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. 2nd Edn., Springer, Berlin, Germany, ISBN:978-3-662-09947-6, Pages: 604.