Neural, Parallel, and Scientific Computations 20 (2012) 399-414

SMART ALTERNATING GROUP EXPLICIT METHOD (SMAGE) FOR THE CUBIC SPLINE SOLUTION OF NON-LINEAR TWO POINT BOUNDARY VALUE PROBLEMS

JYOTI TALWAR AND R.K. MOHANTY

Department of Mathematics
Faculty of Mathematical Sciences
University of Delhi, Delhi-110 007, India
E-mail: chhabrajyoti@gmail.com; rmohanty@maths.du.ac.in

Abstract: In this paper, we discuss the application of smart alternating group explicit (SMAGE) iteration and Newton-SMAGE iteration methods for the cubic spline solution of non-linear differential equation $u'' = f(x,u,u')$ subject to given natural boundary conditions. We compared the results of proposed SMAGE iteration method with the results of corresponding two parameter alternating group explicit (TAGE) iteration methods to demonstrate computationally the efficiency of the proposed method.

Keywords: Two point boundary value problems; Nonlinear first derivative terms; Cubic spline approximations; SMAGE method; Newton-SMAGE method; Burgers' equation.

1. INTRODUCTION

Consider the two point boundary value problem

$$L[u(x)] \equiv -u''(x) + f(x,u,u') = 0, \quad 0 < x < 1 \tag{1}$$

with natural boundary conditions

$$u(0) = A, \quad u(1) = B \tag{2}$$

where A and B are constants. We assume that for $0 < x < 1$ and $-\infty < u, v < \infty$

(i) $f(x,u,v)$ is continuous,

Received May 2, 2011

(ii) $\frac{\partial f}{\partial u}$ and $\frac{\partial f}{\partial v}$ exist and are continuous, and

(iii) $\frac{\partial f}{\partial u} > 0$ and $\left|\frac{\partial f}{\partial v}\right| \le W$ for some positive constant W.

These conditions assure that the boundary value problem (1)-(2) has a unique solution (Keller, 1992).

(Chawla & Subramanian, 1988) constructed a fourth order cubic spline method for second-order mildly nonlinear two point boundary value problems. In the recent past, many authors have suggested various numerical methods based on cubic spline approximations for the solution of linear singular two point boundary value problems (Al-Said, 2001; Ravi Kanth & Reddy, 2005; Al-Said & Noor, 2006). (Evans, 1985) developed the group explicit method for solving large linear systems arising from the discretization of differential equations. (Sukon & Evans, 1996) introduced two parameter alternating group explicit (TAGE) iterative methods for the solution of tri-diagonal linear system of equations. Later, (Mohanty & Evans, 2003; Mohanty *et al*, 2004) discussed the application of TAGE iterative method to fourth order accurate cubic spline approximation for the solution of non-linear singular two point boundary problems. In this paper, we discuss the smart alternating group explicit (SMAGE) and Newton-SMAGE iteration methods, and fourth order cubic spline finite difference approximation and their application to linear and nonlinear differential equations.

2. CUBIC SPLINE APPROXIMATION AND APPLICATION

To obtain a cubic spline solution of the boundary value problem (1) and (2), we choose a uniform mesh spacing $h > 0$ along the x-direction. The interval [0, 1] is divided into a set of points with interval spacing of $h = 1/(N+1)$, N being a positive integer. The cubic spline approximation to equation (1) is obtained on [0,1] which consists of the central point $x_k = kh$ and the two neighboring points $x_{k+1} = x_k + h$ and $x_{k-1} = x_k - h$, $k = 1(1)N$, where $x_0 = 0$ and $x_{N+1} = 1$. Let $U_k = u(x_k)$ be the exact solution of u at the grid point x_k and is approximated by u_k.

At each internal mesh point x_k, we denote:

$$M_k = u''(x_k) = f(x_k, u(x_k), u'(x_k)), \quad k = 0(1)N+1.$$

Given the values $u_0, u_1, \ldots, u_{N+1}$ of the function $u(x)$ at the mesh points $x_0, x_1, \ldots, x_{N+1}$ and the values of the second derivatives of u at the end points u''_0 and u''_{N+1}, there exists a unique interpolating cubic spline function $S(x)$ with the following properties:

(i) $S(x)$ coincides with a polynomial of degree three on each $[x_{k-1}, x_k]$, $k = 1(1)N+1$

(ii) $S(x) \in C^2[0,1]$ and

(iii) $S(x_k) = u_k$, $k = 0(1)N+1$

The interpolating cubic spline polynomial may be written as:

$$\begin{aligned} S(x) &= \frac{(x_k - x)^3}{6h} M_{k-1} + \frac{(x - x_{k-1})^3}{6h} M_k + \left(u_{k-1} - \frac{h^2}{6} M_{k-1}\right)\frac{(x_k - x)}{h} \\ &+ \left(u_k - \frac{h^2}{6} M_k\right)\frac{(x - x_{k-1})}{h}, \quad x_{k-1} \le x \le x_k, \quad k = 1(1)N+1 \end{aligned} \tag{3}$$

We consider the following approximations:

$$x_{k\pm\eta} = x_k \pm \eta h, \quad 0 < \eta \le 1, \tag{4.1}$$

$$\bar{m}_k = \bar{u}'_k = \frac{(u_{k+1} - u_{k-1})}{2h}, \tag{4.2}$$

$$\bar{m}_{k\pm1} = \frac{(\pm 3u_{k\pm1} \mp 4u_k \pm u_{k\mp1})}{2h}, \tag{4.3}$$

$$\bar{f}_k = f(x_k, u_k, \bar{m}_k), \tag{4.4}$$

$$\bar{f}_{k\pm1} = f(x_{k\pm1}, u_{k\pm1}, \bar{m}_{k\pm1}), \tag{4.5}$$

$$\bar{\bar{u}}_{k\pm\eta} = \eta u_{k\pm1} + (1-\eta)u_k + h^2(p\bar{f}_{k\pm1} + q\bar{f}_k), \tag{4.6}$$

$$\bar{\bar{m}}_{k\pm\eta} = \pm\frac{1}{h}(u_{k\pm1} - u_k) \pm h(p^*\bar{f}_{k\pm1} + q^*\bar{f}_k), \tag{4.7}$$

$$\hat{m}_k = \bar{m}_k - \frac{h}{12}(\bar{f}_{k+1} - \bar{f}_{k-1}), \tag{4.8}$$

$$\bar{\bar{f}}_{k\pm\eta} = f(x_{k\pm\eta}, \bar{\bar{u}}_{k\pm\eta}, \bar{\bar{m}}_{k\pm\eta}), \tag{4.9}$$

$$\hat{f}_k = f(x_k, u_k, \hat{m}_k), \tag{4.10}$$

where $p = \dfrac{\eta(\eta^2 - 1)}{6}$, $p^* = \dfrac{dp}{d\eta} = \dfrac{1}{2}\left(\eta^2 - \dfrac{1}{3}\right)$,

$q = \dfrac{(1-\eta)\left[(1-\eta)^2 - 1\right]}{6}$, $q^* = \dfrac{dq}{d\eta} = \dfrac{1}{2}\left[\dfrac{1}{3} - (1-\eta)^2\right]$.

Then the cubic spline method with order of accuracy four for the differential equation (1) may be written as:

$$U_{k+1} - 2U_k + U_{k-1} = \frac{h^2}{12\eta^2}\left[\bar{\bar{f}}_{k+\eta} + \bar{\bar{f}}_{k-\eta} + (12\eta^2 - 2)\,\hat{f}_k\right] + T_k, \qquad 0 < \eta \le 1, \quad k = 1(1)N \tag{5}$$

where $T_k = O(h^6)$ (Jain & Aziz, 1983) with $u_0 = A$ and $u_{N+1} = B$.

Let us discuss the application of the difference formula (5) to the following singular problems

$$u'' = D(x)u' + E(x)u + f(x), \quad 0 < x < 1 \tag{6}$$

and

$$vu'' = B(x)u' + uu' + C(x)u + g(x), \quad 0 < x < 1 \tag{7}$$

where $v = R_e^{-1} > 0$ is a constant and

$D(x) = -\alpha/x$ and $E(x) = \alpha/x^2$, $B(x) = -\alpha v/x$ and $E(x) = \alpha v/x^2$.

For $\alpha = 0$, the non-linear singular problem (7) represents steady-state Burger's equation in Cartesian coordinates.

Now applying the difference formula (5) to the singular equations (6) and (7) and using the technique discussed by (Mohanty *et al.*, 2003), we may obtain the following fourth order difference scheme

$$a_k u_{k-1} + 2b_k u_k + c_k u_{k+1} = d_k, \quad 0 < \eta \le 1, \; k = 1(1)N, \tag{8}$$

for the numerical solution of the differential equation (6), where

$$a_k = -1 + \frac{\alpha}{12\eta}\left(\frac{6\eta}{k} + \frac{1}{2k^3}(6\eta + \alpha\eta - 2\alpha) + \frac{(2 - \alpha\eta)}{k^2}\right),$$

$$b_k = 1 + \frac{\alpha}{12\eta}\left(\frac{1}{k^2}(6\eta + \alpha\eta - 2) + \frac{1}{2k^4}(6\eta + \alpha\eta - 2\alpha)\right),$$

$$c_k = -1 + \frac{\alpha}{12\eta}\left(\frac{-6\eta}{k} - \frac{1}{2k^3}(6\eta + \alpha\eta - 2\alpha) + \frac{(2-\alpha\eta)}{k^2}\right),$$

$$d_k = \frac{-h^2}{12\eta}\left[\eta\left(12 f_k + \frac{h\alpha}{k} f_k' + h^2 f_k''\right) + \frac{\alpha}{k^2} f_k\left(3\eta^2 - 2\eta\right)\right].$$

and the following fourth order difference scheme

$$\begin{aligned}\phi(u_{k-1}, u_k, u_{k+1}) \equiv & -v\left[u_{k+1} - 2u_k + u_{k-1}\right] + \frac{h^2}{12}\Big[I_1 u_k + I_2\left(u_{k+1} - u_{k-1}\right) \\ & + I_3\left(u_{k+1} - 2u_k + u_{k-1}\right) + I_4 u_k^2 + I_5 u_k\left(u_{k+1} - u_{k-1}\right) + I_6 u_k\left(u_{k+1} - 2u_k + u_{k-1}\right) \\ & + I_7\left(u_{k+1}^2 - u_{k-1}^2\right) + I_8\left(u_{k+1}^2 - u_{k-1}^2\right)\left(u_{k+1} - u_{k-1}\right) + I_9 u_k^2\left(u_{k+1} - 2u_k + u_{k-1}\right) \\ & + I_{10} u_k\left(u_{k+1} - u_{k-1}\right)^2 + \Sigma P\Big] = 0, \quad k = 1(1)N,\end{aligned} \tag{9}$$

for the numerical solution of the differential equation (7), where

$$I_1 = \frac{12\alpha v}{(x_k)^2} + \frac{\alpha v h^2 (6-\alpha)}{(x_k)^4} - v^{-1} h^2 f_k', \qquad I_2 = \frac{-6\alpha v}{h x_k} + \frac{\alpha v h(\alpha - 6)}{2(x_k)^3} - \frac{v^{-1} h}{2} f_k,$$

$$I_3 = \frac{\alpha v(2-\alpha)}{(x_k)^2}, \quad I_4 = \frac{2\alpha h^2}{(x_k)^3}, \quad I_5 = \frac{4}{h} - \frac{2\alpha h}{3(x_k)^2}, \quad I_6 = \frac{2\alpha}{(x_k)},$$

$$I_7 = \frac{1}{h} + \frac{\alpha h}{3(x_k)^2}, \quad I_8 = \frac{v^{-1}}{6}, \quad I_9 = -v^{-1}, \quad I_{10} = -\frac{v^{-1}}{3},$$

and

$$\Sigma P = 12 f_k + h^2 f_k'' + \frac{\alpha h^2}{(x_k)^2}\left(f_k + x_k f_k'\right).$$

In order to avoid the numerical complexity, we consider $\eta = 1$.

If the differential equation is linear, we can apply the two parameter SMAGE iterative method and in the non-linear case, we can use the Newton-SMAGE iterative method to obtain the solution.

3 SMAGE AND NEWTON-SMAGE ALGORITHMS

The linear system (8) in matrix form may be written as:

$$\boldsymbol{Ay} = \boldsymbol{RH} \tag{10}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 2b_1 & c_1 & & & \boldsymbol{0} \\ a_2 & 2b_2 & c_2 & & \\ & & \ddots & & \\ & & a_{N-1} & 2b_{N-1} & c_{N-1} \\ \boldsymbol{0} & & & a_N & 2b_N \end{bmatrix}_{N\times N}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_N \end{bmatrix}_{N\times 1} \text{ and } \boldsymbol{RH} = \begin{bmatrix} \sum f_1 - a_1 y_0 \\ \sum f_2 \\ \vdots \\ \vdots \\ \sum f_N - c_N y_{N+1} \end{bmatrix}_{N\times 1} = \begin{bmatrix} RH_1 \\ RH_2 \\ \vdots \\ \vdots \\ RH_N \end{bmatrix}_{N\times 1} \text{ (say).}$$

To implement the SMAGE iterative method, we split the coefficient matrix $\boldsymbol{A}$ into two sub-matrices $\boldsymbol{A} = \boldsymbol{G}_1 + \boldsymbol{G}_2$, where $\boldsymbol{G}_1 + \omega \boldsymbol{I}$ and $\boldsymbol{G}_2 + \omega \boldsymbol{I}$ are non-singular for any $\omega > 0$. Now we discuss the case when N is odd (with $x_0 = 0, x_{N+1} = 1$).

Let $$\boldsymbol{G}_1 = \begin{bmatrix} \boxed{b_1} & & & & \boldsymbol{0} \\ & \boxed{\begin{matrix} b_2 & c_2 \\ a_3 & b_3 \end{matrix}} & & & \\ & & \ddots & & \\ \boldsymbol{0} & & & \boxed{\begin{matrix} b_{N-1} & c_{N-1} \\ a_N & b_N \end{matrix}} \end{bmatrix}_{N\times N}, \text{and} \quad \boldsymbol{G}_2 = \begin{bmatrix} \boxed{\begin{matrix} b_1 & c_1 \\ a_2 & b_2 \end{matrix}} & & & & \boldsymbol{0} \\ & \ddots & & & \\ & & \boxed{\begin{matrix} b_{N-1} & c_{N-1} \\ a_N & b_N \end{matrix}} & & \\ & \boldsymbol{0} & & & \boxed{b_N} \end{bmatrix}_{N\times N}$$

So that the system (10) can be re-written as

$$(\boldsymbol{G}_1 + \boldsymbol{G}_2)\boldsymbol{y} = \boldsymbol{RH} \tag{11}$$

Then a SMAGE method for solving the above system may be written as

$$\boldsymbol{z}^{(s)} = (\boldsymbol{G}_2 - \omega \boldsymbol{I})\boldsymbol{y}^{(s)}, \quad s = 0,1,2,... \tag{12}$$

$$(\boldsymbol{G}_1 + \omega \boldsymbol{I})\boldsymbol{y}^{(s+1/2)} = \boldsymbol{RH} - \boldsymbol{z}^{(s)}, \quad s = 0,1,2,... \tag{13}$$

$$(\boldsymbol{G}_2 + \omega \boldsymbol{I})\boldsymbol{y}^{(s+1)} = 2\omega \boldsymbol{y}^{(s+1/2)} + \boldsymbol{z}^{(s)}, \quad s = 0,1,2,... \tag{14}$$

where $\boldsymbol{z}^{(s)} = [z_1, z_2, \ldots, z_N]^T$ and $\boldsymbol{y}^{(s+1/2)}$ is an intermediate vector. The SMAGE iterative method saves time because of the single evaluation of the common term $(\boldsymbol{G}_2 - \omega \boldsymbol{I})\boldsymbol{y}^{(s)}$ on the right-hand side of the iterative method (12)-(14).

The algorithm for the method (12)-(14) is as follows:

For simplicity, let us denote $p_k = b_k + \omega, q_k = b_k - \omega$, then the SMAGE method in the matrix form may be written as:

$$\mathbf{z}^{(s)} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{N-1} \\ z_N \end{bmatrix}^{(s)} = \begin{bmatrix} q_1 y_1 + c_1 y_2 \\ a_2 y_1 + q_2 y_2 \\ \vdots \\ a_{N-1} y_{N-2} + q_{N-1} y_{N-1} \\ q_N y_N \end{bmatrix}^{(s)}$$

$$\begin{bmatrix} \boxed{p_1} & & & \boldsymbol{0} \\ & \boxed{\begin{matrix} p_2 & c_2 \\ a_3 & p_3 \end{matrix}} & & \\ & & \ddots & \\ \boldsymbol{0} & & & \boxed{\begin{matrix} p_{N-1} & c_{N-1} \\ a_N & p_N \end{matrix}} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-2} \\ y_{N-1} \\ y_N \end{bmatrix}^{(s+1/2)} = \begin{bmatrix} RH_1 - z_1 \\ RH_2 - z_2 \\ RH_3 - z_3 \\ \vdots \\ RH_{N-2} - z_{N-2} \\ RH_{N-1} - z_{N-1} \\ RH_N - z_N \end{bmatrix}^{(s)},$$

$$\begin{bmatrix} \boxed{\begin{matrix} p_1 & c_1 \\ a_2 & p_2 \end{matrix}} & & & \boldsymbol{0} \\ & \ddots & & \\ & & \boxed{\begin{matrix} p_{N-2} & c_{N-2} \\ a_{N-1} & p_{N-1} \end{matrix}} & \\ \boldsymbol{0} & & & \boxed{p_N} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-2} \\ y_{N-1} \\ y_N \end{bmatrix}^{(s+1)} = 2\omega \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-2} \\ y_{N-1} \\ y_N \end{bmatrix}^{(s+1/2)} + \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{N-2} \\ z_{N-1} \\ z_N \end{bmatrix}^{(s)}$$

By carrying out the necessary algebra (12)-(14) can be written in the explicit form. We obtain the following SMAGE algorithms:

Step I For $k = 1(2)N-1$, we have

$$z_k^{(s)} = q_k y_k^{(s)} + c_k y_{k+1}^{(s)}, \quad s = 0,1,2,\ldots$$
$$z_{k+1}^{(s)} = a_{k+1} y_k^{(s)} + q_{k+1} y_{k+1}^{(s)}, \quad s = 0,1,2,\ldots$$

and $$z_N^{(s)} = q_N y_N^{(s)}, \quad s = 0,1,2,\ldots$$

Step II For k = 1, we have

$$y_1^{(s+1/2)} = \frac{\left(RH_1^{(s)} - z_1^{(s)}\right)}{p_1}, \quad s = 0,1,2,\ldots$$

For $k = 2(2)N-1$, let $\Delta = p_k p_{k+1} - c_k a_{k+1} \neq 0$,

$$R_1 = RH_k^{(s)} - z_k^{(s)}, \quad R_2 = RH_{k+1}^{(s)} - z_{k+1}^{(s)}$$

Then, $$y_k^{(s+1/2)} = \frac{\left(R_1 p_{k+1} - R_2 c_k\right)}{\Delta}, \quad s = 0,1,2,\ldots$$

$$y_{k+1}^{(s+1/2)} = \frac{\left(R_2 p_k - R_1 a_{k+1}\right)}{\Delta}, \quad s = 0,1,2,\ldots$$

Step III For $k = 1(2)N-2$, let $\Delta = p_k p_{k+1} - c_k a_{k+1} \neq 0$,

$$R_3 = 2\omega y_k^{(s+1/2)} + z_k^{(s)}, \quad R_4 = 2\omega y_{k+1}^{(s+1/2)} + z_{k+1}^{(s)}$$

Then, $$y_k^{(s+1)} = \frac{\left(R_3 p_{k+1} - R_4 c_k\right)}{\Delta}, \quad s = 0,1,2,\ldots$$

$$y_{k+1}^{(s+1)} = \frac{\left(R_4 p_k - R_3 a_{k+1}\right)}{\Delta}, \quad s = 0,1,2,\ldots$$

Finally, for k = N,

$$y_N^{(s)} = \frac{\left(2\omega y_N^{(s+1/2)} + z_N^{(s)}\right)}{p_N}, \quad s = 0,1,2,\ldots$$

In a similar manner, we can write the SMAGE algorithm when N is even.

Now, we discuss the Newton-SMAGE algorithm. We follow the technique used by (Evans, 1985).

Let us define

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}_{N\times 1} , \qquad \boldsymbol{\varphi}(\boldsymbol{y}) = \begin{bmatrix} \phi_1(\boldsymbol{y}) \\ \phi_2(\boldsymbol{y}) \\ \vdots \\ \phi_N(\boldsymbol{y}) \end{bmatrix}_{N\times 1}$$

and

$$a_k(\boldsymbol{y}) = \frac{\partial \phi_k}{\partial y_{k-1}}, \quad k = 2(1)N,$$

$$2b_k(\boldsymbol{y}) = \frac{\partial \phi_k}{\partial y_k}, \quad k = 1(1)N,$$

$$c_k(\boldsymbol{y}) = \frac{\partial \phi_k}{\partial y_{k+1}}, \quad k = 1(1)N-1.$$

Then the Jacobian of $\boldsymbol{\varphi}(\boldsymbol{y})$ can be written as the Nth-order tri-diagonal matrix

$$T = \frac{\partial \varphi(y)}{\partial y} = \begin{bmatrix} 2b_1(y) & c_1(y) & & & \mathbf{0} \\ a_2(y) & 2b_2(y) & c_2(y) & & \\ & & \ddots & & \\ & & & \ddots & \\ \mathbf{0} & & & a_N(y) & 2b_N(y) \end{bmatrix}_{N\times N} \tag{15}$$

Now with any initial vector $y^{(0)}$, we define

$$y^{(s+1)} = y^{(s)} + \Delta y^{(s)}, \quad s = 0,1,2,... \tag{16}$$

where $\Delta y^{(s)}$ is the solution of the nonlinear system

$$T\,\Delta y^{(s)} = -\varphi(y^{(s)}), \quad s = 0,1,2,... \tag{17}$$

For the Newton-SMAGE method, we consider the case when N is odd. We split the matrix T as $T = T_1 + T_2$, where

$$T_1 = \begin{bmatrix} \boxed{b_1} & & & & \mathbf{0} \\ & \boxed{\begin{matrix} b_2 & c_2 \\ a_3 & b_3 \end{matrix}} & & & \\ & & \ddots & & \\ \mathbf{0} & & & & \begin{matrix} b_{N-1} & c_{N-1} \\ a_N & b_N \end{matrix} \end{bmatrix}_{N\times N}, \quad T_2 = \begin{bmatrix} \boxed{\begin{matrix} b_1 & c_1 \\ a_2 & b_2 \end{matrix}} & & & \mathbf{0} \\ & \ddots & & \\ & & \boxed{\begin{matrix} b_{N-1} & c_{N-1} \\ a_N & b_N \end{matrix}} & \\ \mathbf{0} & & & b_N \end{bmatrix}_{N\times N} \tag{18}$$

then we write Newton-SMAGE method as:

$$z^{(r)} = (T_2 - \omega I)\Delta y^{(r)}, \quad r = 0(1)5 \tag{19}$$

$$(T_1 + \omega I)\Delta y^{(r+1/2)} = -\psi\left(y^{(r)}\right) - z^{(r)}, \quad r = 0(1)5 \tag{20}$$

$$(T_2 + \omega I)\Delta y^{(r+1)} = 2\omega y^{(r+1/2)} + z^{(r)}, \quad r = 0(1)5 \tag{21}$$

where $\omega > 0$ are relaxation parameters and $(T_1 + \omega I)$ and $(T_2 + \omega I)$ are non-singular. Since $(T_1 + \omega I)$ and $(T_2 + \omega I)$ consists of (2×2) sub-matrices, they can be easily inverted. In order for this Newton-SMAGE method to converge it is sufficient that the initial vector $u^{(0)}$ be close to the solution.

In a similar manner, we can write the Newton-SMAGE algorithm when N is even.

4 CONVERGENCE OF SMAGE METHOD

The SMAGE iteration method is given by:

$$\left(\boldsymbol{G}_2+\omega_2\boldsymbol{I}\right)\boldsymbol{u}^{(s+1)}=\left[\boldsymbol{I}-\left(\omega_1+\omega_2\right)\left(\boldsymbol{G}_1+\omega_1\boldsymbol{I}\right)^{-1}\right]\left(\boldsymbol{G}_2-\omega_1\boldsymbol{I}\right)\boldsymbol{u}^{(s)}$$
$$+\left(\omega_1+\omega_2\right)\left(\boldsymbol{G}_1+\omega_1\boldsymbol{I}\right)^{-1}\boldsymbol{RH},\quad s=0,1,2,\ldots$$

or, $$\boldsymbol{u}^{(s+1)}=\boldsymbol{T}_w\boldsymbol{u}^{(s)}+\boldsymbol{RH}_w,\quad s=0,1,2,\ldots \tag{22}$$

where

$$\boldsymbol{T}_w=\left(\boldsymbol{G}_2+\omega_2\boldsymbol{I}\right)^{-1}\left[\left(\boldsymbol{G}_2-\omega_1\boldsymbol{I}\right)-\left(\omega_1+\omega_2\right)\left(\boldsymbol{G}_1+\omega_1\boldsymbol{I}\right)^{-1}\left(\boldsymbol{G}_2-\omega_1\boldsymbol{I}\right)\right]$$

and

$$\boldsymbol{RH}_w=\left(\omega_1+\omega_2\right)\left(\boldsymbol{G}_2+\omega_2\boldsymbol{I}\right)^{-1}\left(\boldsymbol{G}_1+\omega_1\boldsymbol{I}\right)^{-1}\boldsymbol{RH}$$

The matrix $\boldsymbol{T}_w$ is called the SMAGE iteration matrix.

To prove the convergence of the method, we need to prove that $S\left(\boldsymbol{T}_w\right)\le 1$, where $S\left(\boldsymbol{T}_w\right)$ denotes the spectral radius of $\boldsymbol{T}_w$.

Lemma 1. For $\alpha=1$, the eigenvalues of $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are all real, provided $\frac{2}{7}<\eta\le 1$, and for $\alpha=2$, the eigenvalues of $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are all real, provided $\frac{1}{3}<\eta\le 1$.

Proof: Consider $\alpha=1$,

$$\begin{aligned}
a_k &= -1+\frac{1}{12\eta}\left(\frac{6\eta}{k}+\frac{1}{2k^3}\left(7\eta-2\right)+\frac{\left(2-\eta\right)}{k^2}\right),\quad k=1(1)N\\
&\le -1+\frac{1}{12\eta}\left(\frac{6\eta}{k}+\frac{1}{2k}\left(7\eta-2\right)+\frac{\left(2-\eta\right)}{k}\right)\\
&= -1+\frac{1}{12\eta}\left(\frac{1}{2k}\left(17\eta+2\right)\right)\\
&\le -1+\frac{1}{12\eta}\left(\frac{1}{2}\left(17\eta+2\right)\right)=\frac{1}{24\eta}\left(-7\eta+2\right)<0,\quad \text{for}\quad \frac{2}{7}<\eta\le 1
\end{aligned}$$

$$c_k=-1+\frac{1}{12\eta}\left(\frac{-6\eta}{k}-\frac{1}{2k^3}\left(7\eta-2\right)+\frac{\left(2-\eta\right)}{k^2}\right),\quad \text{for}\quad k=1(1)N$$

$$<-1+\frac{1}{12\eta}\left(\frac{\left(2-\eta\right)}{k^2}\right)\le -1+\frac{\left(2-\eta\right)}{12\eta}=\frac{\left(2-13\eta\right)}{12\eta}<0,\quad \text{for}\quad \frac{2}{7}<\eta\le 1.$$

Therefore, for $\alpha = 1$, we have $a_{k+1}c_k > 0, \quad \text{for } k = 1(2)N-1.$

Similarly, for $\alpha = 2$, we can show $a_{k+1}c_k > 0, \quad \text{for } k = 1(2)N-1.$

Let λ_i, $i = 1(1)N$, be the eigenvalues of $\boldsymbol{G}_1$. Then λ_i's are the roots of the quadratic equation

$$\lambda_i^2 - \left(b_i + b_{i+1}\right)\lambda_i + \left(b_i b_{i+1} - a_{i+1}c_i\right) = 0 \tag{23}$$

Simplifying, we get

$$\lambda_i = \frac{1}{2}\left[\left(b_i + b_{i+1}\right) \pm \sqrt{\left(b_i - b_{i+1}\right)^2 + 4a_{i+1}c_i}\,\right] \tag{24}$$

The discriminants of the quadratic equations are

$$\left(b_i - b_{i+1}\right)^2 + 4a_{i+1}c_i > 0, \quad k = 1(2)N-1$$

Hence the eigenvalues of G_1 are real.In a similar manner we can show that the eigenvalues of G_2 are real.

Now we give the sufficient condition for the convergence of the SMAGE method.

Theorem 1: Let λ_i and μ_i, $i = 1(1)N$, be the eigenvalues of $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$, respectively. If

$$\omega_1 > \max\{0, -\lambda_1, \ldots, -\lambda_N\} \tag{25}$$

$$\omega_2 > \max\{0, -\mu_1, \ldots, -\mu_N\} \tag{26}$$

$$\omega_2 - 2\min_i \lambda_i < \omega_1 < \omega_2 + 2\min_i \mu_i \tag{27}$$

then the SMAGE iterative method is convergent for the system (10).

Proof:

Let $\boldsymbol{D} = diag\left(1, \frac{c_1}{a_2}, \frac{c_1 c_2}{a_2 a_3}, \ldots \frac{c_1 c_2 \ldots c_{N-1}}{a_2 a_3 \ldots a_N}\right) \equiv diag\left(d_1, d_2, d_3, \ldots, d_N\right)$

Since the off diagonal entries of $\boldsymbol{A}$ are negative. Therefore $a_{k+1}c_k > 0, k = 1, \ldots, N-1$. Therefore the diagonal entries of $\boldsymbol{D}$ are positive.

The SMAGE iteration matrix is given by:

$$\begin{aligned}\boldsymbol{T}_w &= (\boldsymbol{G}_2+\omega_2\boldsymbol{I})^{-1}\left[(\boldsymbol{G}_2-\omega_1\boldsymbol{I})-(\omega_1+\omega_2)(\boldsymbol{G}_1+\omega_1\boldsymbol{I})^{-1}(\boldsymbol{G}_2-\omega_1\boldsymbol{I})\right]\\ &= (\boldsymbol{G}_2+\omega_2\boldsymbol{I})^{-1}\left[\boldsymbol{I}-(\omega_1+\omega_2)(\boldsymbol{G}_1+\omega_1\boldsymbol{I})^{-1}\right](\boldsymbol{G}_2-\omega_1\boldsymbol{I})\end{aligned}$$

Define

$$\begin{aligned}\boldsymbol{T}_w^* &= (\boldsymbol{G}_2+\omega_2\boldsymbol{I})\boldsymbol{T}_w(\boldsymbol{G}_2+\omega_2\boldsymbol{I})^{-1}\\ &= \left[\boldsymbol{I}-(\omega_1+\omega_2)(\boldsymbol{G}_1+\omega_1\boldsymbol{I})^{-1}\right](\boldsymbol{G}_2-\omega_1\boldsymbol{I})(\boldsymbol{G}_2+\omega_2\boldsymbol{I})^{-1}\end{aligned}$$

$$\boldsymbol{S}(\boldsymbol{T}_w)=\boldsymbol{S}(\boldsymbol{T}_w^*)=\boldsymbol{S}\left(\boldsymbol{D}^{1/2}\boldsymbol{T}_w^*\boldsymbol{D}^{-1/2}\right)$$

$$\boldsymbol{D}^{1/2}\boldsymbol{T}_w^*\boldsymbol{D}^{-1/2}=\left[\boldsymbol{I}-(\omega_1+\omega_2)(\bar{\boldsymbol{G}}_1+\omega_1\boldsymbol{I})^{-1}\right](\bar{\boldsymbol{G}}_2-\omega_1\boldsymbol{I})(\bar{\boldsymbol{G}}_2+\omega_2\boldsymbol{I})^{-1}$$

where $\bar{\boldsymbol{G}}_1=\boldsymbol{D}^{1/2}\boldsymbol{G}_1\boldsymbol{D}^{-1/2}$ and $\bar{\boldsymbol{G}}_2=\boldsymbol{D}^{1/2}\boldsymbol{G}_2\boldsymbol{D}^{-1/2}$.

$$\begin{aligned}\boldsymbol{S}(\boldsymbol{T}_w)=\boldsymbol{S}(\boldsymbol{T}_w^*)=\boldsymbol{S}\left(\boldsymbol{D}^{1/2}\boldsymbol{T}_w^*\boldsymbol{D}^{-1/2}\right)&\le\left\|\boldsymbol{D}^{1/2}\boldsymbol{T}_w^*\boldsymbol{D}^{-1/2}\right\|_2\\ &\le\left\|\left[\boldsymbol{I}-(\omega_1+\omega_2)(\bar{\boldsymbol{G}}_1+\omega_1\boldsymbol{I})^{-1}\right]\right\|_2\left\|(\bar{\boldsymbol{G}}_2-\omega_1\boldsymbol{I})(\bar{\boldsymbol{G}}_2+\omega_2\boldsymbol{I})^{-1}\right\|_2\end{aligned}$$

It is easy to verify that $\bar{\boldsymbol{G}}_1$ and $\bar{\boldsymbol{G}}_2$ are symmetric. Therefore, the matrices $(\bar{\boldsymbol{G}}_2-\omega_1\boldsymbol{I})(\bar{\boldsymbol{G}}_2+\omega_2\boldsymbol{I})^{-1}$ and $\left[\boldsymbol{I}-(\omega_1+\omega_2)(\bar{\boldsymbol{G}}_1+\omega_1\boldsymbol{I})^{-1}\right]$ are also symmetric.

Hence,

$$\left\|(\bar{\boldsymbol{G}}_2-\omega_1\boldsymbol{I})(\bar{\boldsymbol{G}}_2+\omega_2\boldsymbol{I})^{-1}\right\|_2=\max_{\mu_i\in\sigma(\bar{\boldsymbol{G}}_2)}\left|\frac{(\mu_i-\omega_1)}{(\mu_i+\omega_2)}\right|$$

$$\begin{aligned}\left\|\left[\boldsymbol{I}-(\omega_1+\omega_2)(\bar{\boldsymbol{G}}_1+\omega_1\boldsymbol{I})^{-1}\right]\right\|_2&=\left\|\left[(\bar{\boldsymbol{G}}_1+\omega_1\boldsymbol{I})(\bar{\boldsymbol{G}}_1+\omega_1\boldsymbol{I})^{-1}-(\omega_1+\omega_2)(\bar{\boldsymbol{G}}_1+\omega_1\boldsymbol{I})^{-1}\right]\right\|_2\\ &=\left\|\left[(\bar{\boldsymbol{G}}_1-\omega_2\boldsymbol{I})\right]\left[(\bar{\boldsymbol{G}}_1+\omega_1\boldsymbol{I})^{-1}\right]\right\|_2\end{aligned}$$

$(\bar{\boldsymbol{G}}_1-\omega_2\boldsymbol{I})(\bar{\boldsymbol{G}}_1+\omega_1\boldsymbol{I})^{-1}$ is symmetric, therefore

$$\left\|\left[(\bar{\boldsymbol{G}}_1-\omega_2\boldsymbol{I})\right]\left[(\bar{\boldsymbol{G}}_1+\omega_1\boldsymbol{I})^{-1}\right]\right\|_2=\max_{\lambda_i\in\sigma(\bar{\boldsymbol{G}}_1)}\left|\frac{(\lambda_i-\omega_2)}{(\lambda_i+\omega_1)}\right|$$

Therefore, we have

$$S\left(T_w\right) \le \max_{\lambda_i \in \sigma\left(\bar{G}_1\right)} \left| \frac{\left(\lambda_i - \omega_2\right)}{\left(\lambda_i + \omega_1\right)} \right| \max_{\mu_i \in \sigma\left(\bar{G}_2\right)} \left| \frac{\left(\mu_i - \omega_1\right)}{\left(\mu_i + \omega_2\right)} \right|$$

From equations (25) and (26) we have: $\omega_1, \omega_2 > 0$ and $\lambda_i + \omega_1 > 0$ for $k = 1,\ldots,N$, hence

$$\frac{\lambda_i - \omega_2}{\lambda_i + \omega_1} < 1, \quad k = 1(1)N$$

Also, from (27) we have:

$$\omega_2 < \omega_1 + 2\min_i \lambda_i < \omega_1 + 2\lambda_i, \quad i = 1(1)N$$

$$-1 < \frac{\lambda_i - \omega_2}{\lambda_i + \omega_1}, \quad k = 1(1)N$$

Hence, we conclude that $\left| \frac{\lambda_i - \omega_2}{\lambda_i + \omega_1} \right| < 1, \quad k = 1(1)N$

Thus, $\max_{\lambda_i \in \sigma\left(\bar{G}_1\right)} \left| \frac{\left(\lambda_i - \omega_2\right)}{\left(\lambda_i + \omega_1\right)} \right| < 1$

Similarly, we can prove that $\max_{\mu_i \in \sigma\left(\bar{G}_2\right)} \left| \frac{\left(\mu_i - \omega_1\right)}{\left(\mu_i + \omega_2\right)} \right| < 1$

Hence, $S\left(T_w\right) < 1$.

Hence, the convergence of the SMAGE method (22) follows.

5 NUMERICAL ILLUSTRATIONS

We have solved the following two problems to illustrate the proposed SMAGE iterative method, whose exact solutions are known. We have also compared the proposed SMAGE iterative methods with the corresponding TAGE iterative methods. The right-hand side functions and boundary conditions can be obtained by using the exact solutions. The initial vector $\boldsymbol{0}$ is used in all iterative methods, and iterations were stopped when $\left|\boldsymbol{u}^{(s+1)} - \boldsymbol{u}^{(s)}\right| \le 10^{-10}$ was achieved. While solving non-linear difference equations, we have considered five inner iterations only.

Problem 1 $\quad u'' = \beta u', \quad 0 < x < 1$ (Convection-Diffusion equation) (28)

The exact solution is $u(x)=\left(1-e^{-\beta(1-x)}\right)\Big/\left(1-e^{-\beta}\right)$.The root mean square (RMS) errors and the number of iterations both for SMAGE and TAGE methods are tabulated in table 1 for various values of β.

Problem 2 $\quad \nu u''=(u-\beta)u', \quad 0<x<1$ (Burgers' equation) (29)

The exact solution is $u(x)=\beta\left[1-\tanh\left(\beta x/2\nu\right)\right]$.The root mean square (RMS) errors and the number of iterations both for both Newton-SMAGE and Newton-TAGE methods are tabulated in table 2 for $\beta=1/2$ and various values of $R_e=\nu^{-1}$.

Table 1: Problem 1: the RMS errors

	TAGE method			SMAGE method			
N	$\omega_{1opt}=\omega_{2opt}=\omega_{opt}$	Iter	cpu time (in sec)	$\omega_{1opt}=\omega_{2opt}=\omega_{opt}$	Iter	cpu time (in sec)	RMS errors (for both TAGE and CAGE method)
$\beta=10$							
10	0.725	24	0.0016	0.72	24	0.0013	0.1619(-03)
20	0.41	48	0.0034	0.4	42	0.0021	0.1169(-04)
30	0.28	70	0.0062	0.27	66	0.0038	0.2428(-05)
40	0.21	100	0.0108	0.21	85	0.0059	0.7884(-06)
60	0.15	150	0.0228	0.137	125	0.0116	0.1599(-06)
80	0.11	200	0.0398	0.103	163	0.0193	0.5131(-07)
$\beta=100$							
10	6.0	18	0.0014	5.98	18	0.00118	0.8820(-01)
20	2.4	17	0.0019	2.38	17	0.00145	0.1977(-01)
30	1.61	20	0.0024	1.61	21	0.00182	0.6125(-02)
40	1.22	26	0.0035	1.22	27	0.00249	0.2331(-02)
60	0.82	38	0.0065	0.835	39	0.00423	0.5187(-03)
80	0.62	50	0.0105	0.617	52	0.00669	0.1684(-03)

Table 2: Problem 2: the RMS errors

	Newton-TAGE method			Newton-SMAGE method			
N	$\omega_{1opt} = \omega_{2opt} = \omega_{opt}$	Iter	cpu time (in sec)	$\omega_{1opt} = \omega_{2opt} = \omega_{opt}$	iter	cpu time (in sec)	RMS errors (for both Newton-TAGE and Newton-CAGE method)
$R=10, \beta=1/2$							
20	0.0270	15	0.0190	0.0270	15	0.0173	0.6970(-06)
30	0.0202	21	0.0212	0.0196	21	0.0182	0.1452(-06)
40	0.0154	28	0.0248	0.0150	28	0.0194	0.4719(-07)
60	0.0105	43	0.0353	0.0105	41	0.0228	0.9581(-08)
80	0.0093	57	0.0505	0.0077	55	0.0273	0.3081(-08)
$R=50, \beta=1/2$							
20	0.0100	06	0.0176	0.012	05	0.0168	0.1992(-03)
30	0.0080	06	0.0180	0.0079	06	0.0169	0.4113(-04)
40	0.0054	08	0.0191	0.006	07	0.0172	0.1295(-04)
60	0.0041	09	0.0208	0.0041	09	0.0179	0.2571(-05)
80	0.0029	12	0.0246	0.0031	11	0.0189	0.8188(-06)
$R=100, \beta=1/2$							
20	0.0070	05	0.0175	0.007	05	0.0164	0.1016(-02)
30	0.0078	05	0.0182	0.0061	05	0.0169	0.3038(-03)
40	0.0044	06	0.0183	0.0059	05	0.0171	0.1518(-03)
60	0.0041	06	0.0195	0.004	06	0.0176	0.3085(-04)
80	0.00303	07	0.0212	0.003	07	0.0181	0.9571(-05)

6 FINAL REMARKS

The TAGE method requires two sweeps to solve a problem and also, it requires a lot of algebra for computational work. In SMAGE method the amount of computational work is comparatively reduced because of the evaluation of the common term. Experimentally, although both TAGE and SMAGE method require approximately the same number of iterations, but as compared to the TAGE method the corresponding SMAGE method requires less time. We have solved two problems and numerical results shows the efficiency of the proposed SMAGE method.

ACKNOWLEDGEMENT

This research was supported by 'The Council of Scientific and Industrial Research' under research grant No. 09/045(0836)2009-EMR-I.

REFERENCES

1. Keller, H.B. (1992). Numerical Methods for Two Point Boundary Value Problem, New York: Dover Publication.

2. Chawla, M.M., & Subramanian, R. (1988). A new fourth order cubic spline method for second-order nonlinear two point boundary value problems, J. of Comp. and Appl. Math., v. 23, pp. 1-10.

3. Al-Said, E. A. (2001). The Use of Cubic Splines in the Numerical Solution of Systems of Second-Order Boundary Value Problems, Comput. Math. Applics., v. 42, pp. 861-869.

4. Ravi Kanth, A.S.V., & Reddy, Y.N. (2005). Cubic spline for a class of singular two-point boundary value problems, Appl. Math. Comput., v. 170, pp. 733–740.

5. Al-Said, E. A., & Noor, M.A. (2006). Cubic spline method for solving fourth order obstacle problems, Applied Math. Comput., v. 174, pp. 180-187.

6. Evans, D.J. (1985). Group explicit methods for solving large linear systems, Int. J. Comput. Math., v. 17, pp. 81-108.

7. Evans, D.J., & Mohanty, R.K. (2005). On the application of the SMAGE parallel algorithms on a non-uniform mesh for the solution of non-linear two-point boundary value problems with singularity, Int. J. Comput. Math., v. 82, pp. 341-353.

8. Mohanty, R.K., & Evans, D.J. (2003). A fourth order accurate cubic spline alternating group explicit method for non-linear singular two point boundary value problems, Int. J. Comput. Math., v. 80, pp. 479-492.

9. Mohanty, R.K., Sachdev, P.L., & Jha, N. (2004). An $O(h^4)$ accurate cubic spline TAGE method for non-linear singular two point boundary value problems, Appl. Math. Comput., v. 158, pp. 853-868.

10. Sukon, K.S., & Evans, D.J. (1996). Two Parameter AGE Method for the Solution of Tridiagonal Linear System of Equations , Int. J. Comput. Math., v. 60, pp. 265-278.

11. Jain, M.K., & Aziz, T. (1983). Cubic spline solution of two point boundary value problems with significant first derivatives, Comput. Meth. Appl. Mech. Eng., v. 39, pp. 83-91.