International Journal of Bifurcation and Chaos, Vol. 14, No. 10 (2004) 3587–3605

MULTIGRID-CONJUGATE GRADIENT TYPE METHODS FOR REACTION–DIFFUSION SYSTEMS*

S.-L. CHANG
Center for General Education,
Southern Taiwan University of Technology,
Tainan 710, Taiwan

H.-S. CHEN and C.-S. CHIEN†
Department of Applied Mathematics,
National Chung-Hsing University,
Taichung 402, Taiwan
†*cschien@amath.nchu.edu.tw*

Received August 7, 2002; Revised September 30, 2003

We study multigrid methods in the context of continuation methods for reaction–diffusion systems, where the Bi-CGSTAB and GMRES methods are used as the relaxation scheme for the V-cycle, W-cycle and full approximation schemes, respectively. In particular, we apply the results of Brown and Walker [1997] to investigate how the GMRES method can be used to solve nearly singular systems that occur in continuation problems. We show that for the sake of switching branches safely, one would rather solve a perturbed problem near bifurcation points. We propose several multigrid-continuation algorithms for curve-tracking in nonlinear elliptic eigenvalue problems. Our numerical results show that the algorithms proposed have the advantage of being robust and easy to implement.

Keywords: Reaction–diffusion systems; continuation methods; bifurcation; finite differences; multigrid methods.

1. Introduction

In this paper we are concerned with numerical methods for solving parameter-dependent problems of the following form

$$G(u, \lambda) = 0\,, \tag{1}$$

where u represents the solution (i.e. flow fields, displacements, concentrations of some intermediate chemicals in a reaction–diffusion system, etc.), and λ is a vector of physical parameters (i.e. Reynolds number, load, initial or final products, temperature, etc.). Equation (1) arises, for instance, in homotopy continuation methods and nonlinear eigenvalue problems. In this paper we will concentrate on the latter. We wish to solve Eq. (1) numerically by the continuation methods based on parametrizing the solution branch by arc-length, say $[u(s), \lambda(s)]$. First, it is necessary to discretize Eq. (1), for example, by finite difference methods or finite element methods. In both cases, Eq. (1) is approximated by a finite-dimensional problem

$$H(x, \lambda) = 0\,, \tag{2}$$

where $H : \mathbf{R}^N \times \mathbf{R}^k \to \mathbf{R}^N$ is a smooth mapping with $x \in \mathbf{R}^N$ and $\lambda \in \mathbf{R}^k$, $k \geq 1$. Viewing

*Supported by the National Science Council of R.O.C. (TAIWAN) through Project NSC 91-2115-M005-003.
†Author for correspondence.

some component of λ as the continuation parameter, the continuation algorithm can be implemented to follow solution curves of Eq. (2). Specifically, in the context of predictor–corrector continuation methods [Allgower & Georg, 1997; Keller, 1987] one needs to solve bordered linear systems of the following form

$$By = b\,, \tag{3}$$

where $B \in \mathbf{R}^{(N+1)\times(N+1)}$ is the augmented matrix of the Jacobian $DH = [D_x H,\, D_\lambda H] \in \mathbf{R}^{N\times(N+1)}$, and $b \in \mathbf{R}^{N+1}$. In this paper we assume that the matrix $D_x H \in \mathbf{R}^{N\times N}$ is nonsymmetric, and $x = 0$ is a trivial solution of (2).

If a solution curve c of (2) is not regular, that is, rank $DH(x^*,\, \lambda^*) < N$ for some $(x^*,\, \lambda^*) \in c$, then c might contain singular points such as turning points and bifurcation points. Recently some conjugate gradient type methods were proposed to solve Eq. (3) and to detect bifurcation points along the solution curve c. See e.g. [Chang & Chien, 2003b; Desa *et al.*, 1992] and the further references cited therein. A well-known technique for switching branches at a bifurcation point is to solve the perturbed problem [Keller, 1987, pp. 112–113]

$$H(x,\, \lambda) = \tau q\,, \quad q \in \mathbf{R}^N\,, \quad \|q\|_2 = 1\,, \quad \tau \neq 0\,. \tag{4}$$

Assuming τq is a regular value, then Eq. (4) has no bifurcation point, and the corresponding linear systems (3) are nonsingular.

During the past two decades various numerical methods have been proposed for solving Eqs. (1)–(3). Perhaps AUTO97 [Doedel *et al.*, 1997; Doedel *et al.*, 2000] is one of the well-known software packages using continuation methods to treat bifurcation problems in ordinary differential equations and dynamical systems. A comprehensive report of numerical methods for bifurcation problems in dynamical systems was given in [Govaerts, 2000].

Multigrid methods have been regarded as efficient solvers for elliptic boundary value problems [Brandt, 1977, 1982]. Application of the multigrid methods in the context of continuation methods for tracing solution branches of nonlinear elliptic eigenvalue problems can be found in [Chan & Keller, 1982; Mittelmann & Weber, 1985; Weber, 1985; Bolstad & Keller, 1986]. Recently Chang and Chien [2003a] proposed a multigrid-Lanczos algorithm for the numerical solutions of nonlinear elliptic eigenvalue problems, where the matrix $D_x H$ is symmetric, and the Lanczos method is used as a relaxation scheme. The algorithm they proposed has the advantage of being robust and that can be easily implemented. They also showed that straightforward implementation of the algorithm is able to trace solution branches of bifurcation problems. We will assume that $D_x H$ is nonsymmetric in this paper.

Brown and Walker [1997] investigated, the behavior of the GMRES method [Saad & Schultz, 1986] for solving a (nearly) nonsymmetric singular system $Ax = b$. They showed that the (near) singularity may or may not affect the performance of GMRES, depending on the nature of the system and the initial approximate solution. For singular A, they gave conditions under which the GMRES iterates converge safely to a least-squares solution or to the pseudo-inverse solution. In nonlinear elliptic eigenvalue problems, one always has to solve nearly singular systems if a bifurcation point is approached in numerical curve-tracking. Then one may ask: How close can we approach a bifurcation point? In other words, can we determine a minimum value of τ in Eq. (4) so that the perturbed area is as small as possible? Moreover, can we replace the perturbed solution branches of Eq. (4) by the least-squares solutions or the pseudo-inverse solutions of nearly singular systems? In this paper we try to give some possible solutions to these two questions.

Our aim here is to design some multigrid-continuation algorithms for solving reaction–diffusion systems in partial differential equations of the following form:

$$\begin{aligned} \frac{\partial u}{\partial t} &= d_1 \triangle u + f(u, v, \lambda)\,, \\ \frac{\partial v}{\partial t} &= d_2 \triangle v + g(u, v, \lambda), \end{aligned} \qquad \text{in} \quad \Omega = [0, l] \times [0, 1]\,, \tag{5}$$

for all $t \geq 0$, subject to the homogeneous Dirichlet boundary conditions

$$u(x, y, t) = u_0\,, \quad v(x, y, t) = v_0\,, \quad (x, y) \in \partial\Omega\,. \tag{6}$$

We will show how the V-cycle, the W-cycle, and the full approximation scheme (FMG) multigrid algorithms [Briggs *et al.*, 2000] can be incorporated in the context of continuation methods for numerical curve-tracking. Both the GMRES and the Bi-CGSTAB [Van Der Vorst, 1992] algorithms will be used as the relaxation schemes in our multigrid-continuation algorithms. Our test problems is the well-known Brusselator equations [Dangelmayr, 1987; Nicolis & Prigogine, 1977; Schaeffer & Golubitsky, 1981]:

$$\begin{aligned} \frac{\partial u}{\partial t} &= d_1 \triangle u + u^2 v - (\lambda + 1)u + \alpha \,, \\ \frac{\partial v}{\partial t} &= d_2 \triangle v - u^2 v + \lambda u \,, \end{aligned} \qquad \text{in} \quad \Omega = [0, l] \times [0, 1] \,, \tag{7}$$

for all $t \geq 0$, subject to the boundary conditions

$$u(x, y, t) = \alpha \,, \quad v(x, y, t) = \frac{\lambda}{\alpha} \,, \qquad (x, y) \in \partial\Omega \,. \tag{8}$$

Here the unknowns u, v are state variables which represent concentrations of some intermediate chemicals in the reaction, d_1 and d_2 are diffusion rates, while λ is one of the control parameters in the system, e.g. initial or final products, catalysts, temperature, etc. Equation (7) with boundary conditions (8) were numerically investigated in [Chien & Chen, 1998], where a continuation-BCG algorithm was proposed to trace solution curves branching from various types of bifurcation points, such as transcritical and pitchfork bifurcation points. We believe that the multigrid-continuation algorithms we propose in this paper are more efficient than the algorithm described there.

This paper is organized as follows. In Sec. 2 we give a brief review of the GMRES and the Bi-CGSTAB algorithms. In Sec. 3 we discuss how the GMRES method can be used to solve nearly singular linear systems that occur in continuation problems. We show that for the sake of numerical stability, it would be better to solve the perturbed problem (4), even if we can choose a small parameter τ. We discuss various multigrid-continuation algorithms for tracing solution branches of nonlinear elliptic eigenvalue problems in Sec. 4, where the GMRES and the Bi-CGSTAB algorithms are used as the relaxation schemes. The numerical algorithms described in Sec. 4 were numerically implemented for our test problem. Our numerical results are reported in Sec. 5. Finally, some concluding remarks are given in Sec. 6.

2. A Brief Review of the GMRES and the Bi-CGSTAB Algorithms

2.1. *The GMRES algorithm*

We consider the following linear system

$$Ax = b \,, \tag{9}$$

where $A \in \mathbf{R}^{N \times N}$ is nonsymmetric and $b \in \mathbf{R}^N$. Let $x_0 \in \mathbf{R}^N$ be an initial guess to the solution of (9), and $r_0 = b - Ax_0$ the corresponding residual. Let $v_1 = r_0/\beta_1$ with $\beta_1 = \|r_0\|_2$. The GMRES generates a sequence of orthonormal vectors $v_1, \ldots, v_k$ for the Krylov subspace $\mathcal{K}(A, v_1, k)$ such that

$$\begin{aligned} \text{span}\{v_1, \ldots, v_k\} &= \text{span}\{v_1, Av_1, \ldots, A^{k-1}v_1\} \\ &\equiv \mathcal{K}(A, v_1, k) \\ &\equiv \mathcal{K}_k \,. \end{aligned}$$

Let $V_k = [v_1, v_2, \ldots, v_k]$ be the $N \times k$ matrix. The GMRES algorithm is described as follows:

Algorithm 2.1. *GMRES*

1. **Start:** *Set* $r_0 := b - Ax_0$ *and* $v_1 := r_0/\beta_1$ *with* $\beta_1 := \|r_0\|_2$.
2. **Iterate:**
   For $j = 1, 2, \ldots, k, \ldots$ *until satisfied do*:

$$\begin{aligned} h_{i,j} &:= (Av_j, v_i) \,, \quad i = 1, 2, \ldots, j \\ \hat{v}_{j+1} &:= Av_j - \sum_{i=1}^{j} h_{i,j} v_i \,, \\ h_{j+1,j} &:= \|\hat{v}_{j+1}\|_2 \,, \\ v_{j+1} &:= \hat{v}_{j+1}/h_{j+1,j} \,. \end{aligned}$$

3. **Form the approximate solution:**

$$x_k := x_0 + V_k y_k \,,$$

where y_k solves the minimization problem

$$\min_{y\in\mathbf{R}^k} \|\beta_1 e_1 - \overline{H}_k y\|_2.$$

Here $e_1 = (1,0,\ldots,0)^T \in \mathbf{R}^{k+1}$, and $\overline{H}_k \in \mathbf{R}^{(k+1)\times k}$ is the matrix whose nonzero entries are the same as those of the block upper Hessenberg matrix obtained from Step 2 of Algorithm 2.1 except for additional $k+1$ rows whose only nonzero entry is $h_{k+1,k}$ in the last k columns position. Therefore, we have

$$AV_k = V_{k+1}\overline{H}_k\,.$$

In Step 3 of Algorithm 2.1, one can easily check that the approximate solution x_k satisfies the following minimization problem

$$\min_{x\in x_0+\mathcal{K}_k} \|b - Ax\|_2 = \min_{y\in\mathbf{R}^k} \|\beta_1 e_1 - \overline{H}_k y\|_2\,.$$

Let P_i denote the set of polynomials q_i of degree less than or equal to i such that $q_i(0) = 1$. Eisenstat *et al.* [1983] derived some error bounds and convergence results of the generalized conjugate gradient algorithm (GCG) for solving nonsymmetric linear systems. These results also hold for GMRES. We state one of them as follows:

Theorem 2.2. *Let $\{r_i\}$ be the sequence of residual vectors generated by GMRES and let $M := (A + A^T)/2$ be positive-definite matrix, then*

$$\|r_i\|_2 \le \min_{q_i\in P_i} \|q_i(A)\|_2 \|r_0\|_2$$

$$\le \left[1 - \frac{\lambda_{\min}(M)}{\lambda_{\max}(A^TA)}\right]^{i/2} \|r_0\|_2\,.$$

Hence, GMRES converges.

In order to accelerate the rate of convergence of the GMRES method, we need to impose preconditioning techniques on the linear system Eq. (9). Assume that M is a preconditioner of A. Then we can transform Eq. (9) into the following preconditioned linear system

$$\overline{A}\overline{x} = \overline{b}\,, \tag{10}$$

where $\overline{A} = M^{-1}AM^{-T}$, $\overline{x} = M^Tx$, and $\overline{b} = M^{-1}b$. Note that Eq. (10) is equivalent to

$$M^{-1}Ax = M^{-1}b\,. \tag{11}$$

The preconditioned GMRES algorithm is given as follows:

Algorithm 2.3. *The preconditioned GMRES algorithm*

1. **Start:** *Choose $x_0 \in \mathbf{R}^N$, compute $\overline{r}_0 := M^{-1}r_0$, and $v_1 := \overline{r}_0/\beta_1$ with $\beta_1 := \|\overline{r}_0\|_2$.*
2. **Iterate:**
   For $j = 1,2,\ldots,m$ Do:
   Compute $w_j := M^{-1}Av_j$
   For $i = 1,2,\ldots,j$ Do:
   $h_{i,j} := (w_j, v_i)$
   $w_j := w_j - h_{i,j}v_i$
   End Do
   $h_{j+1,j} := \|w_j\|_2$
   $v_{j+1} := w_j/h_{j+1,j}$
   End Do
3. **Form the approximate solution:**
   Compute y_m the minimizer of $\|\beta_1 e_1 - \overline{H}_m y\|_2$ and $x_m := x_0 + V_m y_m$, where $\overline{H}_m = \{h_{i,j}\}_{1\le i\le m+1, 1\le j\le m}$ is the upper Hessenberg matrix.

2.2. *The Bi-CGSTAB algorithm*

Van Der Vorst [1992] proposed a variant of the bi-conjugate gradient (Bi-CG) method, called the Bi-CGSTAB method for the solutions of nonsymmetric linear systems. They showed that the Bi-CGSTAB algorithm converges more smoothly than the conjugate gradient squared (CGS) method [Sonneveld, 1989]. In order to accelerate the rate of convergence, the preconditioning techniques can be incorporated in the context of Bi-CGSTAB methods. More precisely, let K be any suitable preconditioning matrix, i.e. $K \approx A$, then we write $K = K_1K_2$ and apply the iteration scheme given above to the explicitly preconditioned system

$$\tilde{A}\tilde{x} = \tilde{b}\,,$$

with $\tilde{A} = K_1^{-1}AK_2^{-1}$, $x = K_2^{-1}\tilde{x}$, and $b = K_1^{-1}b$. The following preconditioned Bi-CGSTAB algorithm can be found in [Van Der Vorst, 1992].

Algorithm 2.4. *The preconditioned Bi-CGSTAB algorithm*

1. **Start:** *Choose $x_0 \in \mathbf{R}^N$, compute $r_0 = b - Ax_0$. Let $\overline{r}_0 \in \mathbf{R}^N$ be arbitrary such that $(\overline{r}_0, r_0) \neq 0$. Set $\rho_0 = \alpha = w_0 = 1$; $v_0 = p_0 = 0$.*
2. **Iterate:**
   For $i = 1,2,\ldots,m$ Do:
   $\rho_i = (\overline{r}_0, r_{i-1})$;
   $\beta = (\rho_i/\rho_{i-1})(\alpha/w_{i-1})$;
   $p_i = r_{i-1} + \beta(p_{i-1} - w_{i-1}v_{i-1})$;

Compute $y = K^{-1}p_i$
$v_i = Ay$;
$\alpha = \rho_i/(\overline{r}_0, v_i)$;
$s = r_{i-1} - \alpha v_i$;
Compute $z = K^{-1}s$
$t = Az$;
$w_i = (K_1^{-1}t, K_1^{-1}s)/(K_1^{-1}t, K_1^{-1}t)$
$x_i = x_{i-1} + \alpha y + w_i z$;
End Do

3. **Form the approximate solution:**
Compute x_m, *the minimizer of* $r_i = s - w_i t$.

3. Solving Nearly Singular Linear Systems

We consider parameter-dependent problems of the form

$$H(x, \lambda) = 0\,, \tag{12}$$

where $H : \mathbf{R}^N \times \mathbf{R} \to \mathbf{R}^N$ is a continuously differentiable function of $x \in \mathbf{R}^N$ and $\lambda \in \mathbf{R}$. We denote the Jacobian of H by $DH = [D_x H, D_\lambda H]$, and the solution curve c of (12) by

$$c = \{y(s) = (x(s), \lambda(s)) | H(y(s)) = 0, s \in I\}\,.$$

Here I is any interval in $\mathbf{R}$. Assume that a parameterization via arc length is available on c. We will trace the solution curve c by the predictor–corrector continuation methods. Let $y_i = (x_i, \lambda_i) \in \mathbf{R}^{N+1}$ be a point which has been accepted as an approximating point for the solution curve c. Suppose that the Euler predictor is used to predict a new point $z_{i+1,1}$. That is,

$$z_{i+1,1} = y_i + \delta_i u_i\,. \tag{13}$$

Here $\delta_i > 0$ is the step length, and u_i is the unit tangent vector at y_i, which is obtained by solving

$$\begin{bmatrix} D_x H(y_i) & D_\lambda H(y_i) \\ r_i^T & \end{bmatrix} \cdot u_i = \begin{bmatrix} \overline{0} \\ 1 \end{bmatrix}, \tag{14}$$

for some constraint vector $r_i \in \mathbf{R}^{N+1}$. The accuracy of approximation to the solution curve must be improved via a corrector process. In practice, Newton's method with constraint

$$\begin{bmatrix} D_x H(z_{i+1,j}) & D_\lambda H(z_{i+1,j}) \\ u_i^T & \end{bmatrix} \cdot w_j = \begin{bmatrix} -H(z_{i+1,j}) \\ 0 \end{bmatrix}, \quad j = 1, 2, \ldots \tag{15}$$

is solved, and we set $z_{i+1,j+1} = z_{i+1,j} + w_j$, $j = 1, 2, \ldots$. If y_i lies sufficiently near c, then the modified Newton's method will converge if the step size δ_i is small enough. We rewrite Eq. (14) or (15) as

$$\begin{bmatrix} A & p \\ q^T & \gamma \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{16}$$

which can be simplified as

$$By = b\,, \tag{17}$$

where p, q, $f \in \mathbf{R}^N$ and γ, $g \in \mathbf{R}$. We can use the block elimination algorithm to solve (16). Some iterative methods have been proposed to solve (17). See e.g. [Chien & Chang, 2003b; Desa *et al.*, 1992] and the further references cited therein.

If the assumption 0 a regular value of H is not satisfied, then $H^{-1}(0)$ may contain bifurcation points and the matrix B becomes singular there. On the other hand, if the solution curve c contains a simple fold, then the matrix A becomes singular there. In either case we have to solve nearly singular systems in numerical curve-tracking if we are close to a fold or a bifurcation point.

Brown and Walker [1997] studied the behavior of the GMRES method for solving (nearly) singular linear systems. They show that the (near) singularity of A may or may not affect the performance of GMRES, depending on the nature of the system and the initial approximate solution. If A is singular, they give conditions under which the GMRES iterates converge safely to a least-squares solution or to the pseudo-inverse solution.

Now, suppose that y^* is a primary bifurcation point on the trivial solution curve $c \equiv 0$. Assume that we are approaching y^* in our predictor–corrector continuation algorithm. We wish to switch from the trivial solution branch to the primary solution branch. Then we have to solve nearly singular linear systems. In this case the right-hand side vector $b = 0$. Therefore we have to solve the perturbed problem Eq. (4) otherwise we would jump over the bifurcation point and still trace along the trivial solution branch. However, the results given above show that we can use GMRES to solve nearly singular systems. Note that the coefficient matrix is still nonsingular. Thus we will obtain the unique solution to the linear system but not the least-squares solution or the pseudo-inverse solution. This means that we can approach the bifurcation point as close as we can by choosing a relatively small parameter τ in (4). Furthermore, the step length in the predictor–corrector continuation algorithms should be properly controlled. Let δ be the current step

length in the predictor step. Our numerical experiments show that we can choose τ so that $\tau \approx 2\delta$. It could happen that we might jump over the bifurcation point and trace the trivial solution branch again if $\tau < 2\delta$. In our numerical experiments we chose $\tau \approx 0.1$ and $\delta = 0.05$.

4. Some Multigrid-GMRES (Bi-CGSTAB) Algorithms

It is well-known that multigrid methods are efficient solvers for elliptic boundary value problems [Brandt, 1977; Hackbusch, 1985; Briggs *et al.*, 2000], where the classical Gauss–Seidel method or the SOR is used as a relaxation scheme. During the eighties several researchers have investigated multigrid methods for nonlinear elliptic eigenvalue problems. On the one hand, Chan and Keller [1982] proposed some multigrid-continuation algorithms for nonlinear elliptic eigenvalue problems with folds. Similar problems were also studied by Bolstad and Keller [1986], using full approximation scheme multigrid-continuation methods. On the other hand, Weber [1985] proposed to use singular multigrid iterations to solve singular linear systems of equations which arise from the discretization of a bifurcation problem, where one has to compute the Moore–Penrose inverse of a (nearly) singular coefficient matrix at each iteration. Mittelmann and Weber [1985] also studied multigrid methods for the computation of solution branches in bifurcation problems. Actually, the pioneer work on this research area may go back to [Hackbusch, 1979]. See also [Hackbusch, 1985, Chap. 13] and further references cited therein for details.

Recently Chang and Chien [2003a] proposed a multigrid-Lanczos algorithm for solving symmetric linear systems that arise from the discretization of continuation problems. The algorithm they proposed is efficient and robust, and can be easily implemented. From the results of Brown and Walker [1997] described in Sec. 3, and our numerical experiments, we notice that as far as the coefficient matrix is ill-conditioned but remains to be nonsingular, the GMRES method will find the unique solution of the nearly singular linear system. The results of Chang and Chien [2003a] give us a heuristic idea that a straightforward implementation of the multigrid-GMRES algorithm given below can be used to find solution branches of a bifurcation problem. In other words, it is unnecessary to find the Moore–Penrose inverse of a (nearly) singular matrix and perform the singular multigrid iteration as was done in [Hackbusch, 1985, Chap. 13; Weber, 1985].

In this section we will give a more comprehensive study of multigrid methods for solving nonsymmetric linear systems in nonlinear elliptic eigenvalue problems. We will discuss how the GMRES and the Bi-CGSTAB methods can be used as relaxation schemes in the context of V-cycle, W-cycle and full approximation schemes.

First, we adopt the idea of the V-cycle scheme multigrid method for solving the associated symmetric linear systems. Suppose that there is a hierarchy of grids Ω^h, $\Omega^{2h}, \ldots, \Omega^{Mh}$ defined on a domain Ω, where ih denotes the mesh size on Ω. To start with, we relax on the finest grid Ω^h using the Bi-CGSTAB or the GMRES method until convergence deteriorates. Then we compute the finest grid residual and restrict it to the next finer grid Ω^{2h}, which is to be used as the right-hand side for the linear system on Ω^{2h}. We repeat this process until the coarsest grid Ω^{Mh} is reached. Next, we perform the correction scheme by injecting the solution of the linear system obtained on Ω^{Mh} to $\Omega^{\frac{M}{2}h}$, and adding it to the solution obtained in $\Omega^{\frac{M}{2}h}$, which is to be used as the initial guess for the Bi-CGSTAB or the GMRES method. We relax the same linear system on $\Omega^{\frac{M}{2}h}$ using the BI-CGSTAB or GMRES method again.

This process is repeated until the approximate solution obtained in the finest grid is accurate enough. Our numerical results show that the multigrid-Bi-CGSTAB(GMRES) algorithm proposed here is more efficient than the continuation-Bi-CGSTAB(GMRES) algorithm. Furthermore, it can be easily modified to detect singularity of coefficient matrices. All we need to do is to implement the Bi-CGSTAB method and GMRES on the finest grid if necessary. We refer to [Chien *et al.*, 1997] for details.

One may use either a fixed strategy or an adaptive one to transfer between grids. In a fixed strategy one performs p relaxation sweeps on each grid Ω^{kh} before transferring to a coarser grid $\Omega^{(2k)h}$, and perform q relaxation sweeps before interpolating back to a finer grid $\Omega^{(2k)h}$. In an adaptive strategy one transfers to a coarser grid when the ratio of the residual norm of current iterate to the residual norm a sweep earlier is greater than some tolerance η, and transfer to a finer grid when the ratio of the residual norm of current iterate to the residual norm on the next finer grid is less than another tolerance δ.

The strategy we propose here is based on the convergence theory of the Bi-CGSTAB method and is described as follows. Let x_0^{kh} be an initial guess to the linear system

$$A^{kh}x = f^{kh} \tag{18}$$

on the grid Ω^{kh}, and $r_0^{kh} = f^{kh} - A^{kh}x_0^{kh}$ the corresponding residual. In the V-cycle scheme we relax on Eq. (18) a certain number of sweeps by using the Bi-CGSTAB method, and transfer to a coarser grid $\Omega^{(k+1)h}$ if

$$\|r_0^{kh}\|_2 < \delta_k \cdot \|r_i^{kh}\|_2 \tag{19}$$

is satisfied. Here $r_i^{kh} = f^{kh} - A^{kh}x_i^{kh}$ is the residual vector of the ith sweep x_i^{kh} and $\delta_k \in (0,1)$.

Similarly, in the correction process we also relax on the associated linear system of Eq. (18) a certain number of sweeps and transfer to a finer grid $\Omega^{(k-1)h}$ if the initial residual vector r_0^{kh} and the final residual vector r_j^{kh} satisfy

$$\|r_0^{kh}\|_2 < \eta_k \cdot \|r_j^{kh}\|_2 \tag{20}$$

for some $\eta_k \in (0,1)$.

Our numerical experiments show that the convergence behavior of the multigrid-Bi-CGSTAB algorithm depends on the choices of δ_k and η_k. The strategy of when to transfer between grids is based on the assumption that the Bi-CGSTAB method converges smoothly for nonsymmetric linear systems. In order to accelerate the rate of convergence of the GMRES (Bi-CGSTAB), we suggest solving preconditioned linear systems on the fine grid.

4.1. *A V-cycle scheme*

Let I_{ih}^{jh} stand for the restriction operator, while the bicubic interpolation operator is denoted by I_{jh}^{ih} where $j = 2i$. That is, $I_{ih}^{jh} : \Omega^{ih} \to \Omega^{jh}$ and $I_{jh}^{ih} : \Omega^{jh} \to \Omega^{ih}$.

Algorithm 4.1. *A V-cycle scheme GMRES (Bi-CGSTAB) algorithm with preconditioner on the fine grid*

Input: $\delta_k, \eta_k \in (0,1)$, $k = 1, 2, 3, \ldots$.

- *Relax on* $A^h u^h = f^h$ i_1 *times by the preconditioned GMRES (Bi-CGSTAB) method with initial guess* v^h *so that* $\|r_0^h\|_2 < \delta_1 \cdot \|r_{i_1}^h\|_2$.
- *Set* $r^h = r_{i_1}^h$.
- *Compute* $f^{2h} = I_h^{2h} r^h$.
  - *Relax on* $A^{2h}u^{2h} = f^{2h}$ i_2 *times by the GMRES (Bi-CGSTAB) method with initial guess* $v^{2h} = 0$ *so that* $\|r_0^{2h}\|_2 < \delta_2 \cdot \|r_{i_2}^{2h}\|_2$.
  - *Set* $r^{2h} = r_{i_2}^{2h}$.
  - *Compute* $f^{4h} = I_{2h}^{4h} r^{2h}$.

    ⋮

    - *Solve* $A^{Mh}u^{Mh} = f^{Mh}$ *and obtain* v^{Mh}.

    ⋮

  - *Correct* $v^{2h} \leftarrow v^{2h} + I_{4h}^{2h} v^{4h}$.
  - *Relax on* $A^{2h}u^{2h} = f^{2h}$ j_2 *times by the GMRES (Bi-CGSTAB) method with initial guess* v^{2h} *so that* $\|r_0^{2h}\|_2 < \eta_2 \cdot \|r_{j_2}^{2h}\|_2$.
- *Correct* $v^h \leftarrow v^h + I_{2h}^h v^{2h}$.
- *Relax on* $A^h u^h = f^h$ j_1 *times by the preconditioned GMRES (Bi-CGSTAB) method with initial guess* v^h *so that* $\|r_0^h\|_2 < \eta_1 \cdot \|r_{j_1}^h\|_2$.

By incorporating Algorithm 4.1 into the continuation method, we can obtain the multigrid-Bi-CGSTAB-continuation or the multigrid-GMRES-continuation algorithm for tracing solution branches of nonlinear elliptic eigenvalue problems. These algorithms will not be described here.

4.2. *A W-cycle scheme*

A W-cycle scheme multigrid-GMRES (Bi-CGSTAB) algorithms with preconditioner on the fine grid is given as follows.

Algorithm 4.2. *A W-cycle scheme GMRES (Bi-CGSTAB) algorithm with preconditioner on the fine grid*

Input: $\delta_k, \eta_k \in (0,1)$, $k = 1, 2, 3, \ldots$.

- *Relax on* $A^h u^h = f^h$ i_1 *times by the preconditioned GMRES (Bi-CGSTAB) method with initial guess* v^h *so that* $\|r_0^h\|_2 < \delta_1 \cdot \|r_{i_1}^h\|_2$.
- *Set* $r^h = r_{i_1}^h$.
- *Compute* $f^{2h} = I_h^{2h} r^h$.
  - *Relax on* $A^{2h}u^{2h} = f^{2h}$ i_2 *times by the GMRES (Bi-CGSTAB) method with initial guess* $v^{2h} = 0$ *so that* $\|r_0^{2h}\|_2 < \delta_2 \cdot \|r_{i_2}^{2h}\|_2$.
  - *Set* $r^{2h} = r_{i_2}^{2h}$.
  - *Compute* $f^{4h} = I_{2h}^{4h} r^{2h}$.

    ⋮

    - *Solve* $A^{Mh}u^{Mh} = f^{Mh}$ *and obtain* v^{Mh}.
  - *Correct* $v^{\frac{M}{2}h} \leftarrow v^{\frac{M}{2}h} + I_{Mh}^{\frac{M}{2}h} v^{Mh}$.
  - *Relax on* $A^{\frac{M}{2}h}u^{\frac{M}{2}h} = f^{\frac{M}{2}h}$ i_{m-1} *times by the GMRES (Bi-CGSTAB) method with*

initial guess $v^{\frac{M}{2}h}$ *so that* $\|r_0^{\frac{M}{2}h}\|_2 < \delta_{m-1} \cdot \|r_{i_{m-1}}^{\frac{M}{2}h}\|_2$.

- *Compute* $f^{Mh} = I_{\frac{M}{2}h}^{Mh} r^{\frac{M}{2}h}$.
  - *Solve* $A^{Mh}u^{Mh} = f^{Mh}$ *and obtain* v^{Mh}.
- *Correct* $v^{\frac{M}{2}h} \leftarrow v^{\frac{M}{2}h} + I_{Mh}^{\frac{M}{2}h} v^{Mh}$.
- *Relax on* $A^{\frac{M}{2}h}u^{\frac{M}{2}h} = f^{\frac{M}{2}h}$ j_{m-1} *times by the GMRES (Bi-CGSTAB) method with initial guess* $v^{\frac{M}{2}h}$ *so that* $\|r_0^{\frac{M}{2}h}\|_2 < \delta_{m-1} \cdot \|r_{j_{m-1}}^{\frac{M}{2}h}\|_2$.

- *Correct* $v^{\frac{M}{4}h} \leftarrow v^{\frac{M}{4}h} + I_{\frac{M}{2}h}^{\frac{M}{4}h} v^{\frac{M}{2}h}$.
- *Relax on* $A^{\frac{M}{4}h}u^{\frac{M}{4}h} = f^{\frac{M}{4}h}$ i_{m-2} *times by the GMRES (Bi-CGSTAB) method with initial guess* $v^{\frac{M}{4}h}$ *so that* $\|r_0^{\frac{M}{4}h}\|_2 < \delta_{m-2} \cdot \|r_{i_{m-2}}^{\frac{M}{4}h}\|_2$.
- *Compute* $f^{\frac{M}{2}h} = I_{\frac{M}{4}h}^{\frac{M}{2}h} r^{\frac{M}{4}h}$.
  - *Relax on* $A^{\frac{M}{2}h}u^{\frac{M}{2}h} = f^{\frac{M}{2}h}$ i_{m-1} *times by the GMRES (Bi-CGSTAB) method with initial guess* $v^{\frac{M}{2}h}$ *so that* $\|r_0^{\frac{M}{2}h}\|_2 < \delta_{m-1} \cdot \|r_{i_{m-1}}^{\frac{M}{2}h}\|_2$.
  - *Compute* $f^{Mh} = I_{\frac{M}{2}h}^{Mh} r^{\frac{M}{2}h}$.
    - *Solve* $A^{Mh}u^{Mh} = f^{Mh}$ *and obtain* v^{Mh}.

    ⋮
  - *Correct* $v^{2h} \leftarrow v^{2h} + I_{4h}^{2h} v^{4h}$.
  - *Relax on* $A^{2h}u^{2h} = f^{2h}$ j_2 *times by the GMRES (Bi-CGSTAB) method with initial guess* v^{2h} *so that* $\|r_0^{2h}\|_2 < \delta_2 \cdot \|r_{j_2}^{2h}\|_2$.
- *Correct* $v^h \leftarrow v^h + I_{2h}^h v^{2h}$.
- *Relax on* $A^h u^h = f^h$ j_1 *times by the preconditioned GMRES (Bi-CGSTAB) method with initial guess* v^h *so that* $\|r_0^h\|_2 < \delta_1 \cdot \|r_{j_1}^h\|_2$.

4.3. *A full approximation scheme GMRES (Bi-CGSTAB) algorithm*

Algorithm 4.3. *A full approximation scheme GMRES (Bi-CGSTAB) algorithm with preconditioner on the fine grid*

Input: $\delta_k, \eta_k \in (0,1)$, $k = 1, 2, 3, \ldots$.

- *Initialize* $f^{2h} \leftarrow I_h^{2h} f^h, f^{4h} \leftarrow I_{2h}^{4h} f^{2h}, \ldots$

  Solve or relax on the coarsest grid.

  ⋮
  - *Correct* $v^{4h} \leftarrow v^{4h} + I_{8h}^{4h} v^{8h}$.
  - *Relax on* $A^{4h}u^{4h} = f^{4h}$ j *times by the V-cycle scheme multigrid-GMRES (Bi-CGSTAB) method with initial guess* v^{4h} *so that* $\|r_0^{4h}\|_2 < \eta_3 \cdot \|r_j^{4h}\|_2$.
- *Correct* $v^{2h} \leftarrow v^{2h} + I_{4h}^{2h} v^{4h}$.
- *Relax on* $A^{2h}u^{2h} = f^{2h}$ j *times by the V-cycle scheme multigrid-GMRES (Bi-CGSTAB) method with initial guess* v^{2h} *so that* $\|r_0^{2h}\|_2 < \eta_2 \cdot \|r_j^{2h}\|_2$.
- *Correct* $v^h \leftarrow v^h + I_{2h}^h v^{2h}$.
- *Relax on* $A^h u^h = f^h$ j *times by the V-cycle scheme preconditioned multigrid-GMRES (Bi-CGSTAB) method with initial guess* v^h *so that* $\|r_0^h\|_2 < \eta_1 \cdot \|r_j^h\|_2$.

5. Numerical Results

In this section, we report some numerical results concerning the implementations of the multigrid continuation algorithms described in Sec. 4 with Bi-CGSTAB and GMRES as the relaxation schemes, respectively, on a reaction–diffusion system. Our model problem is the well-known Brusselator equations. All our computations were executed on an IBM Pentium 4 machine with double precision arithmetic at National Chung Hsing University.

NOMENCLATURE

NCS	:	ordering of the continuation steps.
ε	:	accuracy tolerance in Newton corrector.
tol	:	stopping criterion for the corresponsive method.
NCI	:	number of corrector iterations required at each continuation step.
MAXNORM	:	maximum norm of the approximating solution.
p-cycle	:	average number of cycles required in the *V*-cycle, *W*-cycle, and FMG to solve a single linear system in the predictor step.
c-cycle	:	average number of cycles required in the *V*-cycle, *W*-cycle, and FMG to solve a single linear system in the corrector step.
Time	:	total execution time (in seconds) for each method.

Example. The test problem was discretized by the centered difference approximations with $h = 1/64$ on the finest grid, and a total of six levels of grids, making the coarsest grid with $h = 1/2$. The discretization matrix on the finest grid is of order 7938×7938. In the following, the first bifurcation point of (22) is $(0, \lambda) = (0,\ 29.1444935)$. For $h = 1/16, 1/32$ and $1/64$, the first discrete bifurcation points are (0, 29.0824621), (0, 29.1289699), and (0, 29.1406116), respectively. The accuracy tolerance for solving linear systems was 10^{-10}, and the parameters in Algorithms 4.1–4.4 were chosen to be $\delta_1 = 0.0001$, $\eta_1 = 0.1$ and $\delta_k = \eta_k = 0.9$, $k = 2 : 5$.

To simplify the numerical computations, we shift the homogeneous states (u_0, v_0) to $(0, 0)$ by the transformation $(u, v) = (u_0 + \tilde{u}, v_0 + \tilde{v})$. Furthermore, we incorporate explicitly the length l into the equations by the transformation $x = l\tilde{x}$, which changes the domain $\Omega = [0, l] \times [0, 1]$ to the unit square $\tilde{\Omega} = [0, 1] \times [0, 1]$, and Eq. (5) can be expressed as

$$\begin{aligned} \frac{\partial \tilde{u}}{\partial t} &= d_1 \left(\frac{1}{l^2} \frac{\partial^2 \tilde{u}}{\partial \tilde{x}^2} + \frac{\partial^2 \tilde{v}}{\partial \tilde{y}^2} \right) + f(u_0 + \tilde{u}, v_0 + \tilde{v}, \lambda)\,, \\ &\qquad\qquad \text{in } \tilde{\Omega} = [0, 1] \times [0, 1]\,, \\ \frac{\partial \tilde{v}}{\partial t} &= d_2 \left(\frac{1}{l^2} \frac{\partial^2 \tilde{v}}{\partial \tilde{x}^2} + \frac{\partial^2 \tilde{v}}{\partial \tilde{y}^2} \right) + g(u_0 + \tilde{u}, v_0 + \tilde{v}, \lambda)\,, \end{aligned} \tag{21}$$

For simplicity, we denote $\tilde{\Omega}$, $\tilde{x}$, $f(u_0 + \tilde{u}, v_0 + \tilde{v}, \lambda)$, $g(u_0 + \tilde{u}, v_0 + \tilde{v}, \lambda)$, and $\tilde{u}, \tilde{v}$ again by Ω, x, $f(u, v, \lambda)$, $g(u, v, \lambda)$ and u, v, respectively, when no confusion arises. Then we have

$$\begin{aligned} \frac{\partial u}{\partial t} &= d_1 \left(\frac{1}{l^2} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + f(u, v, \lambda)\,, \\ &\qquad\qquad \text{in } \Omega = [0, 1] \times [0, 1]\,, \\ \frac{\partial v}{\partial t} &= d_2 \left(\frac{1}{l^2} \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + g(u, v, \lambda)\,, \end{aligned} \tag{22}$$

where

$$g(u, v, \lambda) = -\lambda u - \alpha^2 v - \left(\frac{\lambda}{\alpha} u^2 + 2\alpha u v + u^2 v \right),$$

with boundary conditions

$$u(x, y, t) = v(x, y, t) = 0 \quad \text{on} \quad \partial\Omega\,.$$

Table 1. Implementing Algorithm 4.1 for the lower solution branch with $h = 1/16$, $\varepsilon = 5 \cdot 10^{-4}$, tol $= 10^{-10}$.

NCI	λ	MAXNORM	p-cycle	c-cycle	Time
5	27.8997909	−5.1429243e-003	1	1	19.16
10	28.7416066	−1.6395584e-002	2	1	47.69
15	29.1856049	−8.0003178e-002	1	2	81.82
20	29.4121194	−1.5887541e-001	2	2	120.01
25	29.6237955	−2.3808050e-001	2	2	158.47
30	29.8384966	−3.1699605e-001	2	2	196.91
35	30.0601053	−3.9547588e-001	2	2	234.81
40	30.2900789	−4.7344731e-001	2	2	273.67
45	30.5291747	−5.5085757e-001	2	2	315.64
50	30.7778832	−6.2766079e-001	2	2	356.17
55	31.0365717	−7.0381374e-001	2	2	396.15

Table 2. Implementing Algorithm 4.1 for the lower solution branch with $h = 1/32$, $\varepsilon = 5 \cdot 10^{-4}$, tol $= 10^{-10}$.

NCI	λ	MAXNORM	p-cycle	c-cycle	Time
5	27.8997909	−2.0056665e-004	1	1	141.05
10	28.7499451	−6.5181164e-004	1	1	337.67
15	29.1676046	−2.2296146e-002	1	1	616.27
20	29.2685066	−6.3374358e-002	2	2	1000.77
25	29.3661716	−1.0443518e-001	2	1	1367.65
30	29.4653811	−1.4544123e-001	2	2	1740.27
35	29.5667482	−1.8638579e-001	2	2	2122.97
40	29.6704779	−2.2726501e-001	2	2	2503.39
45	29.7766796	−2.6807564e-001	2	2	2879.47
50	29.8854314	−3.0881461e-001	2	2	3244.07
55	29.9967988	−3.4947891e-001	2	2	3598.66

Table 3. Implementing Algorithm 4.1 for the lower solution branch with $h = 1/64$, $\varepsilon = 5 \cdot 10^{-4}$, tol $= 10^{-10}$.

NCI	λ	MAXNORM	p-cycle	c-cycle	Time
5	27.8999238	−7.9482473e-004	1	1	1360.47
10	28.7469491	−2.4884024e-003	1	1	3163.38
15	29.1285833	−1.8616773e-002	1	1	5342.06
20	29.2033518	−3.9246340e-002	1	2	8156.78
25	29.2594756	−5.9912773e-002	2	2	11227.63
30	29.3117353	−8.0576571e-002	2	2	14504.45
35	29.3628710	−1.0123185e-001	2	2	17687.19
40	29.4137705	−1.2187665e-001	2	2	20745.71
45	29.4648107	−1.4251001e-001	2	2	23867.58
50	29.5161806	−1.6313134e-001	2	2	26939.49
55	29.5679875	−1.8374016e-001	2	2	30244.28

We chose $\alpha = 4$, $d_1 = 1$ and $d_2 = 2$ in our numerical experiments. First, we implemented Algorithm 4.2 to trace the solution curves of Eq. (22) branching from the first bifurcation point with mesh size $h = 1/16, 1/32$ and $1/64$, where the preconditioned Bi-CGSTAB was used as the relaxation scheme. From [Chien & Chen, 1998] we know that the first bifurcation point is transcritical. Our simple numerical results were shown in Tables 1–3. Figure 1 shows how the upper and lower solution curves bifurcate from the transcritical bifurcation point. Figure 2 displays the convergence behavior of the Bi-CGSTAB method in the V-cycle scheme at $\lambda \approx 29.19$. Here and in what follows, the symbol "+" denotes the number of iterations performed in the finest grid, and the symbol "$*$" denotes the log of residual norm reached after one V-cycle is finished. Figure 3 depicts the number of iterations required in the V-cycle scheme during the continuation process. Figure 4 shows the convergence behavior of the Bi-CGSTAB method in Algorithm 4.3 at $\lambda \approx 29.12$. Figure 5 displays the number of iterations versus the continuation parameter λ in the W-cycle scheme, where the lower solution branch is traced. Figures 6 and 7 show similar results for the FAS scheme with Bi-CGSTAB as the relaxation scheme.

We also used Algorithm 4.2 with GMRES as the relaxation scheme to trace the upper solution branch. Table 4 shows the numerical result. Figure 8 displays the convergence behavior of the GMRES method in the V-cycle scheme. Figure 9 shows the

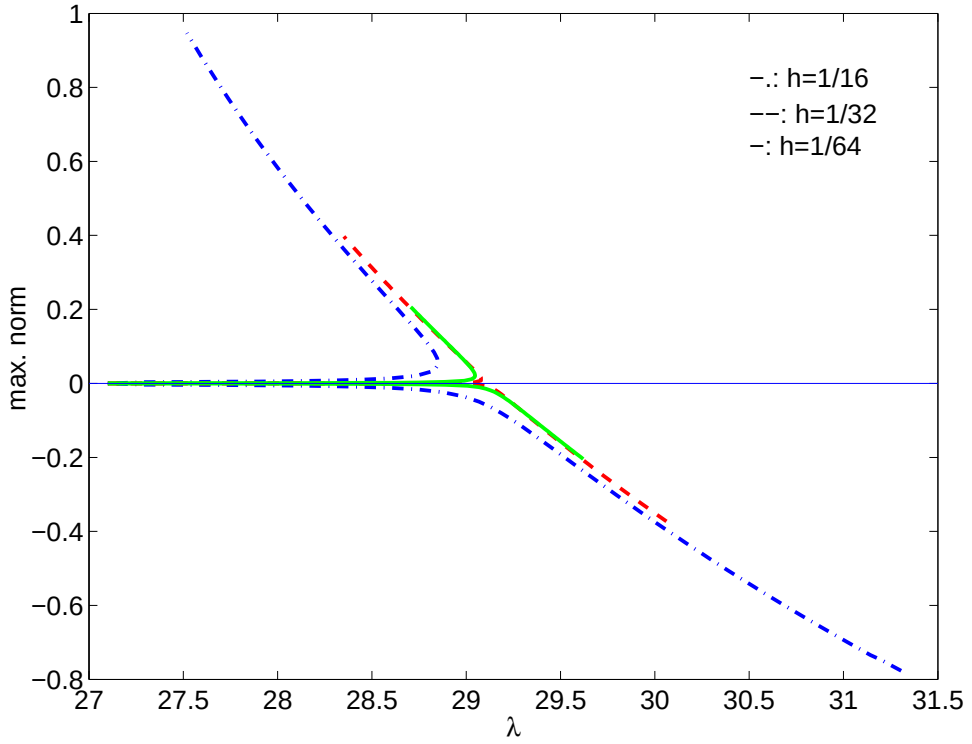

Fig. 1. Solution curves branching from the first bifurcation point of Eq. (22).

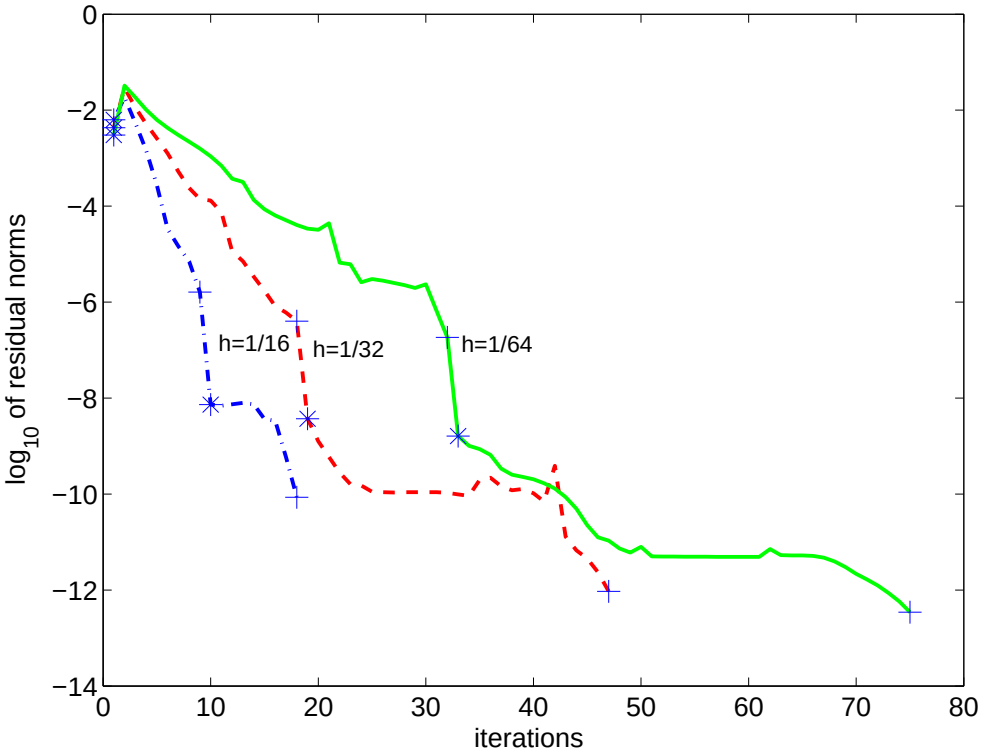

Fig. 2. Convergence behavior of the Bi-CGSTAB method in the V-cycle multigrid algorithm for the Brusselator equations with $\lambda \approx 29.19$.

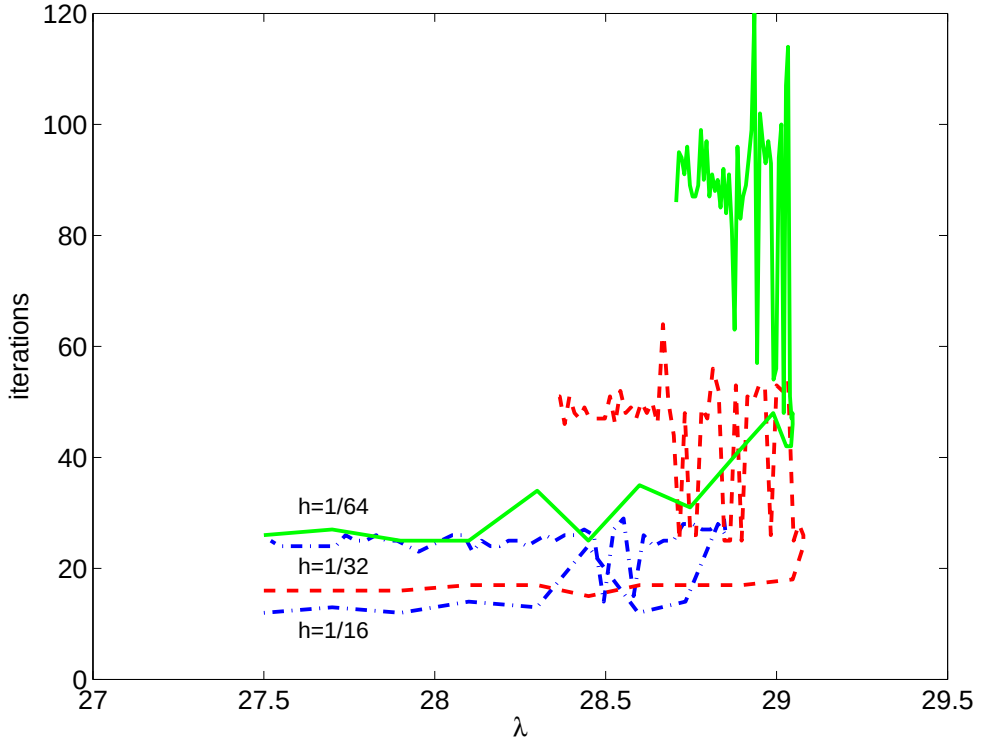

Fig. 3. Number of iterations of the Bi-CGSTAB algorithm versus the parameter λ in the V-cycle scheme, upper solution branch.

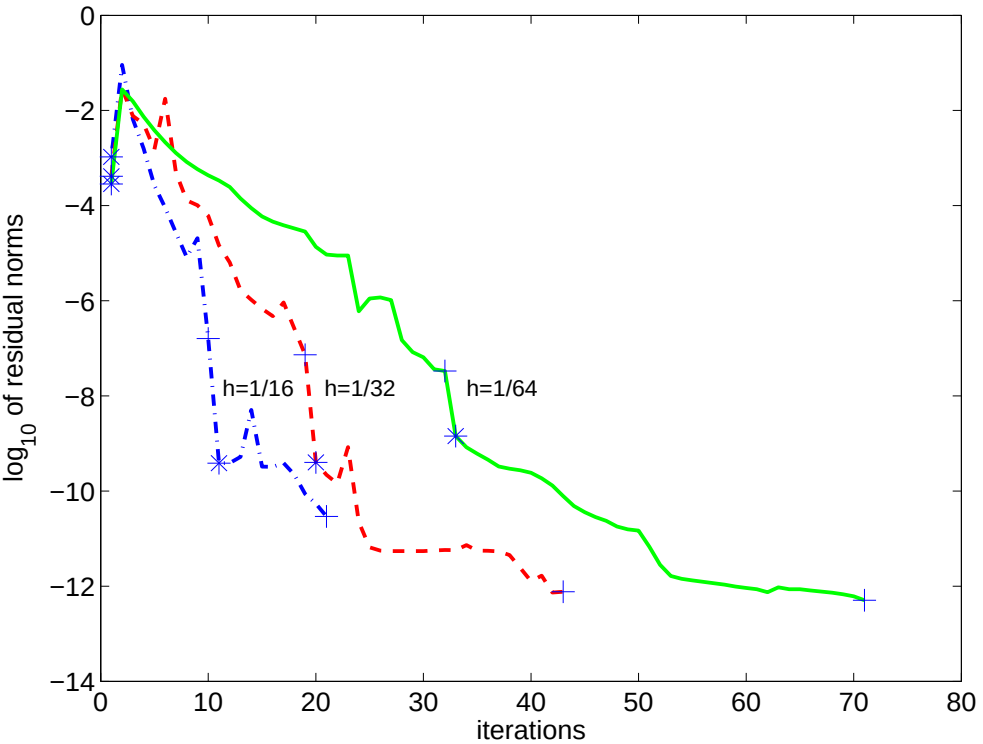

Fig. 4. Convergence behavior of the Bi-CGSTAB method in the W-cycle multigrid algorithm for the Brusselator equations with $\lambda \approx 29.12$.

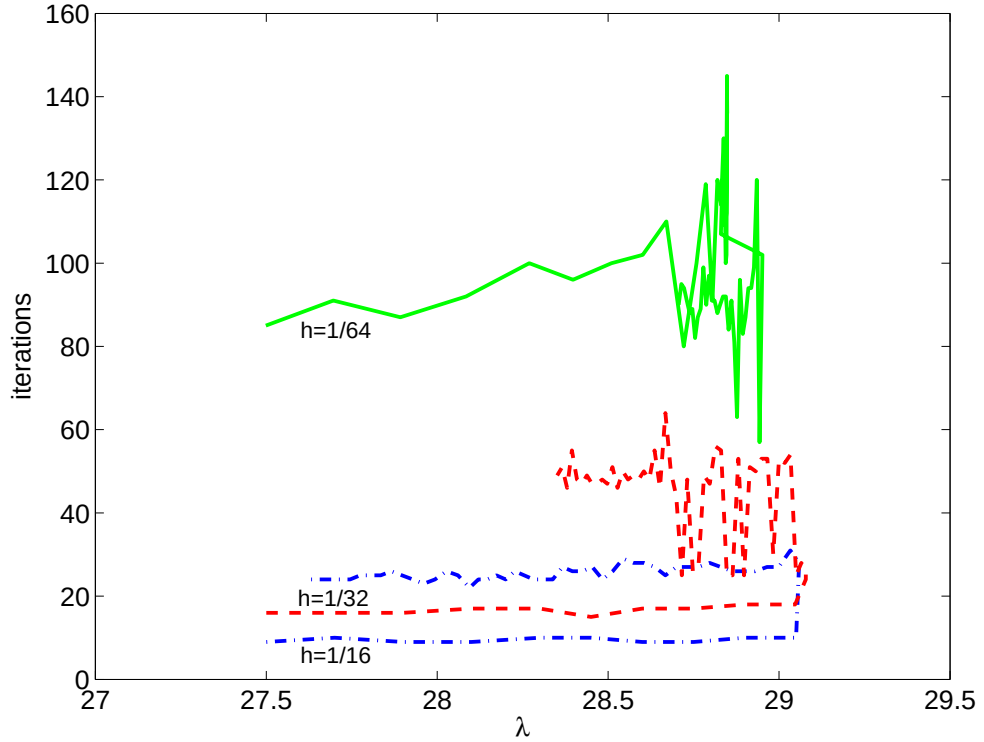

Fig. 5. Number of iterations of the Bi-CGSTAB algorithm versus the parameter λ in the W-cycle scheme, upper solution branch.

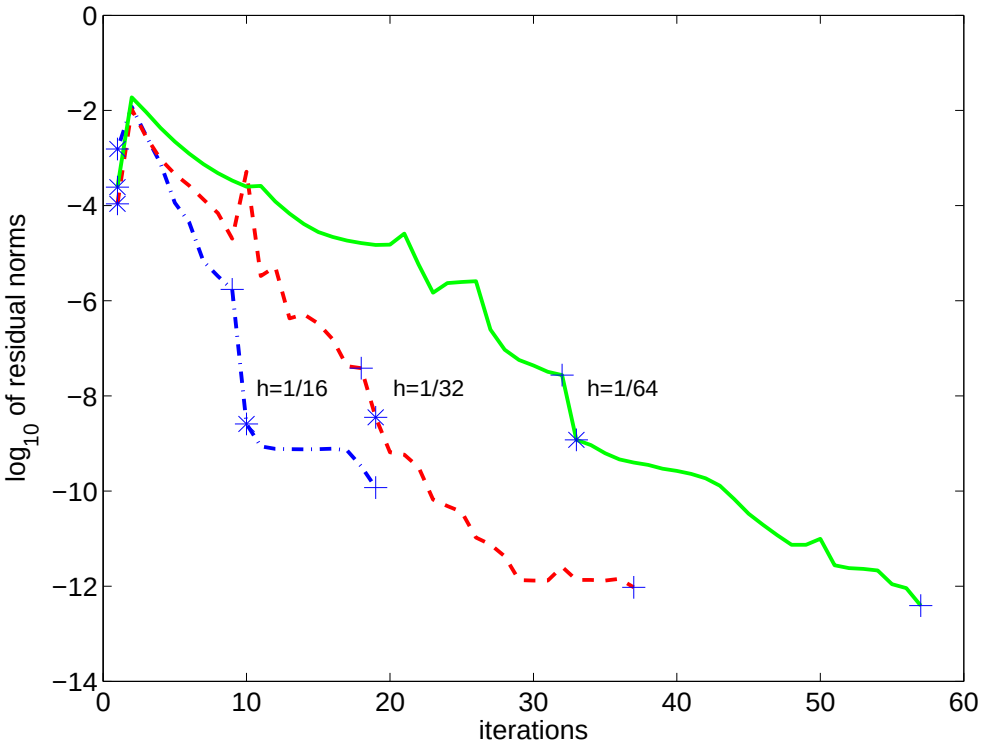

Fig. 6. Convergence behavior of the Bi-CGSTAB method in the FAS multigrid algorithm for the Brusselator equations with $\lambda \approx 29.15$.

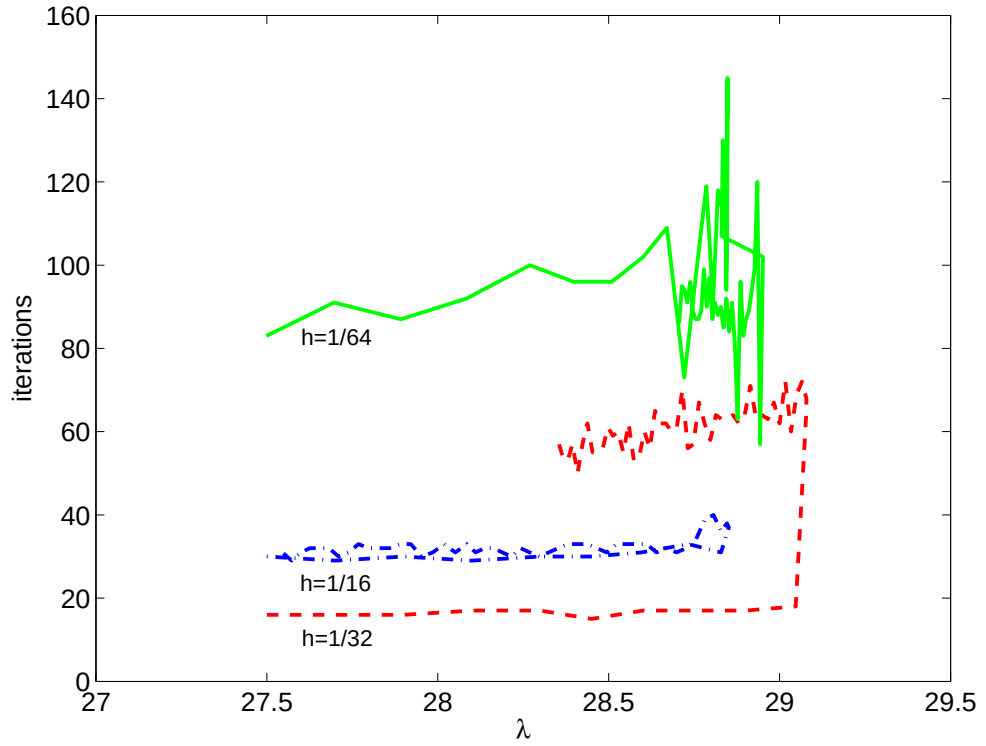

Fig. 7. Number of iterations of the Bi-CGSTAB algorithm versus the parameter λ in the FAS scheme, upper solution branch.

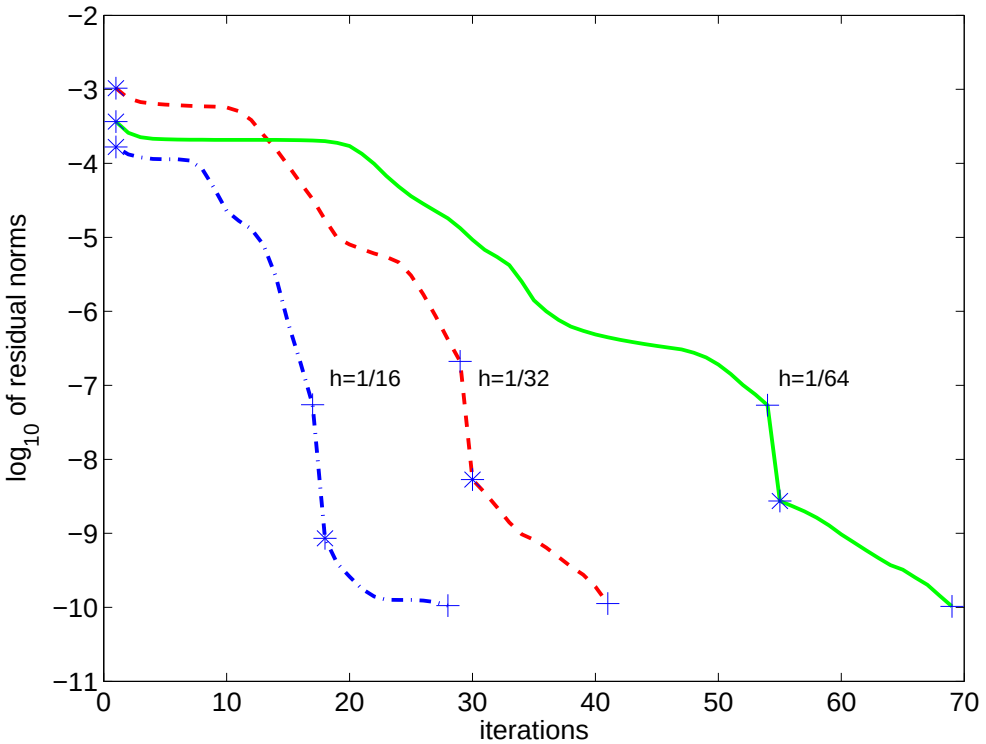

Fig. 8. Convergence behavior of the GMRES method in the V-cycle multigrid algorithm for the Brusselator equations with $\lambda \approx 29.05$.

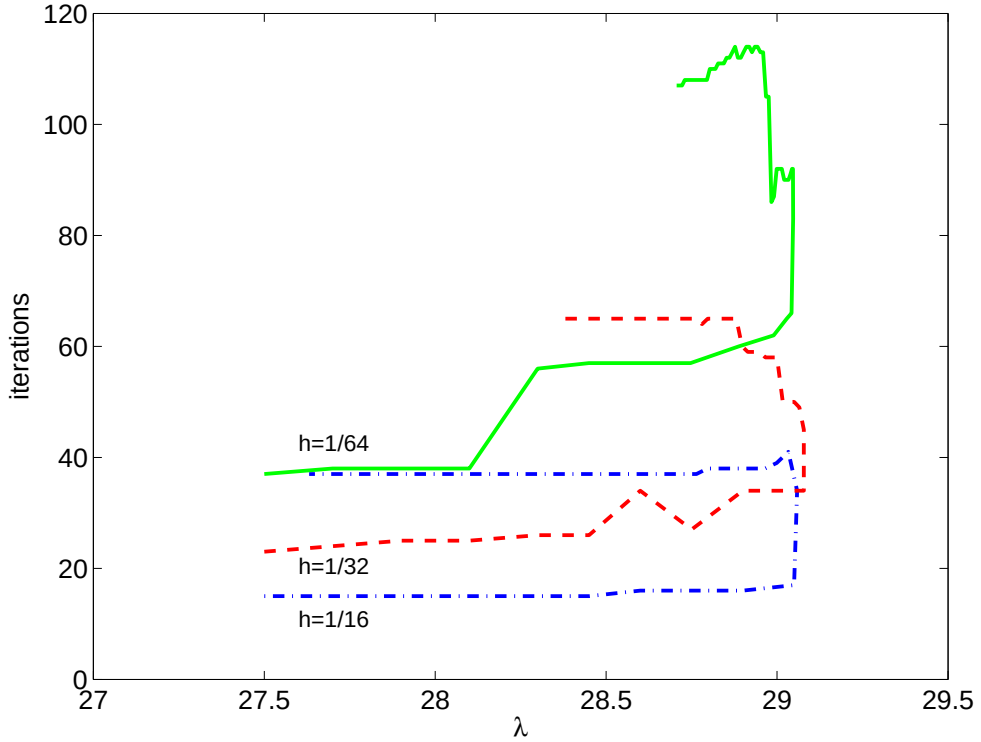

Fig. 9. Number of iterations of the GMRES algorithm versus the parameter λ in the V-cycle scheme, upper solution branch.

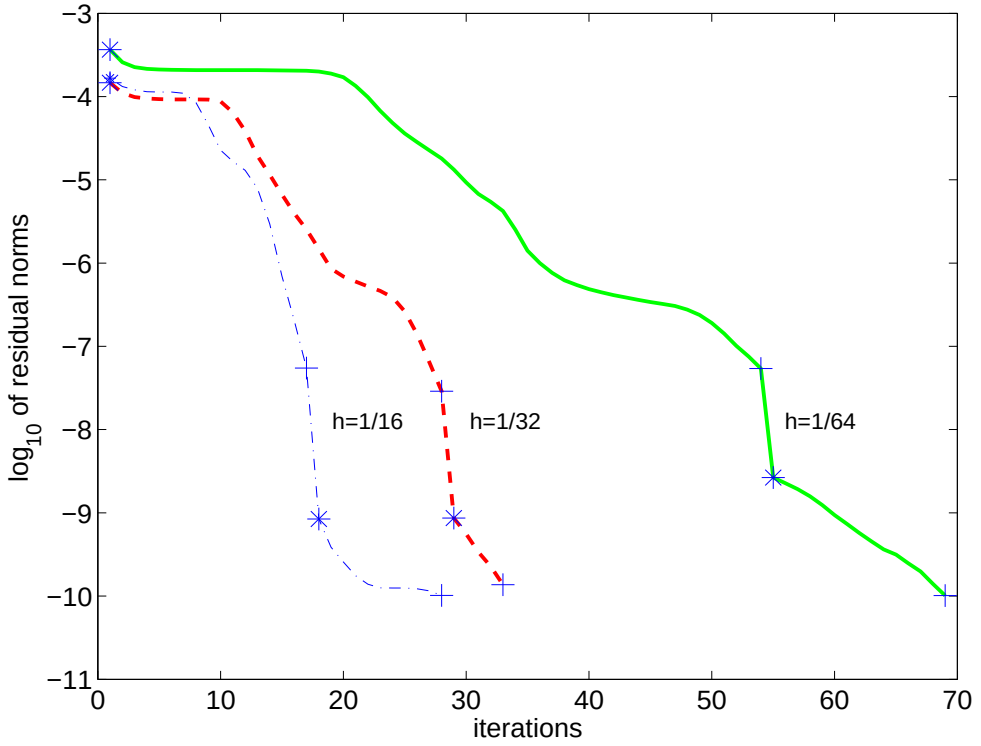

Fig. 10. Convergence behavior of the GMRES method in the W-cycle multigrid algorithm for the Brusselator equations with $\lambda \approx 29.05$.

Table 4. Implementing Algorithm 4.1 for the upper solution branch with $h = 1/64$, $\varepsilon = 5 \cdot 10^{-4}$, tol $= 10^{-10}$.

NCI	λ	MAXNORM	p-cycle	c-cycle	Time
5	27.8999229	7.9690101e-004	1	1	1018.27
10	28.7466659	2.5564784e-003	1	1	2518.10
15	29.0467808	1.9724339e-002	2	2	4507.83
20	29.0274336	4.0478430e-002	2	2	6475.32
25	28.9911630	6.1225915e-002	2	2	8404.26
30	28.9512569	8.1977374e-002	2	2	10503.22
35	28.9102569	1.0273601e-001	2	2	12678.72
40	28.8689922	1.2350283e-001	2	2	14823.27
45	28.8278118	1.4427812e-001	2	2	16959.86
50	28.7868863	1.6506193e-001	2	2	19068.77
55	28.7463076	1.8585415e-001	2	2	21179.05

Table 5. Implementing Algorithm 4.2 for the upper solution branch with $h = 1/64$, $\varepsilon = 5 \cdot 10^{-4}$, tol $= 10^{-10}$.

NCI	λ	MAXNORM	p-cycle	c-cycle	Time
5	27.8999229	7.9690101e-004	1	1	1030.97
10	28.7466659	2.5564784e-003	2	2	2642.03
15	29.0467808	1.9724339e-002	2	2	4852.05
20	29.0274336	4.0478430e-002	2	2	6852.83
25	28.9911630	6.1225915e-002	2	2	8843.50
30	28.9512569	8.1977374e-002	2	2	11015.27
35	28.9102569	1.0273601e-001	2	2	13231.91
40	28.8689922	1.2350283e-001	2	2	15429.16
45	28.8278118	1.4427812e-001	2	2	17612.05
50	28.7868863	1.6506193e-001	2	2	19774.07
55	28.7463076	1.8585415e-001	2	2	21918.50

Table 6. Implementing Algorithm 4.3 for the lower solution branch with $h = 1/64$, $\varepsilon = 5 \cdot 10^{-4}$, tol $= 10^{-10}$.

NCI	λ	MAXNORM	p-cycle	c-cycle	Time
5	27.8999992	−7.9580607e-005	1	1	1626.66
10	28.7499675	−2.5384382e-004	1	1	3535.50
15	29.1554338	−1.0795886e-002	2	2	7143.83
20	29.2080638	−3.1483184e-002	2	2	10296.83
25	29.2565028	−5.2168342e-002	2	2	13406.14
30	29.3048331	−7.2843353e-002	2	2	16501.94
35	29.3535314	−9.3507118e-002	2	2	19848.86
40	29.4027339	−1.1415914e-001	2	2	23273.00
45	29.4524969	−1.3479906e-001	2	2	26461.38
50	29.5028503	−1.5542657e-001	2	2	29445.95
55	29.5538130	−1.7604136e-001	2	2	32613.63

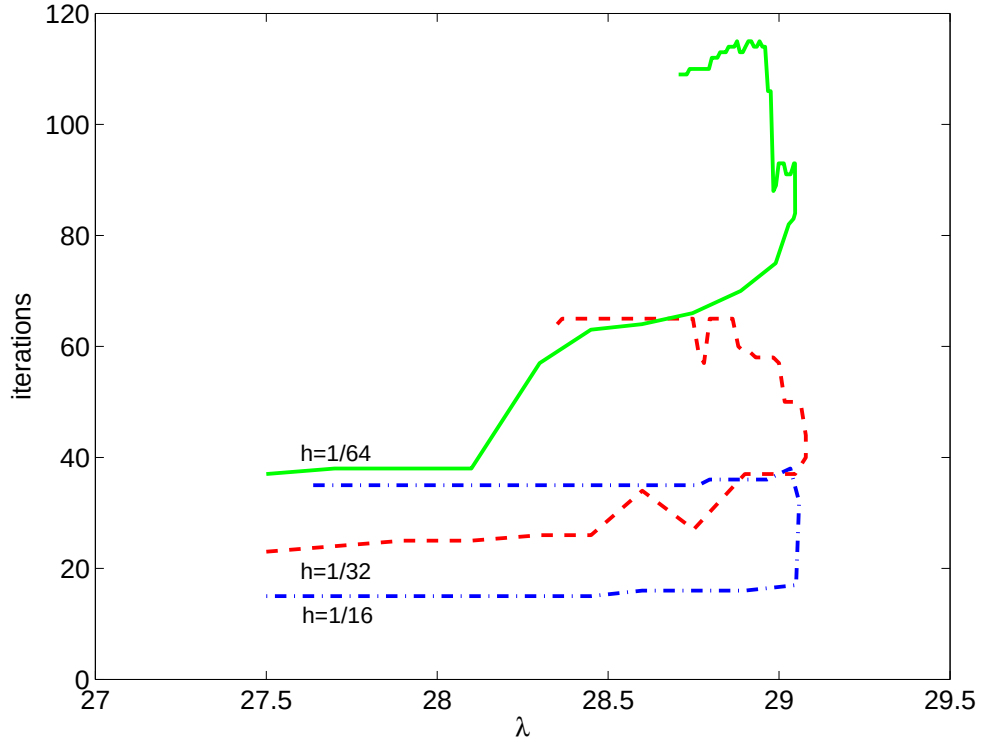

Fig. 11. Number of iterations of the GMRES algorithm versus the parameter λ in the W-cycle scheme, upper solution branch.

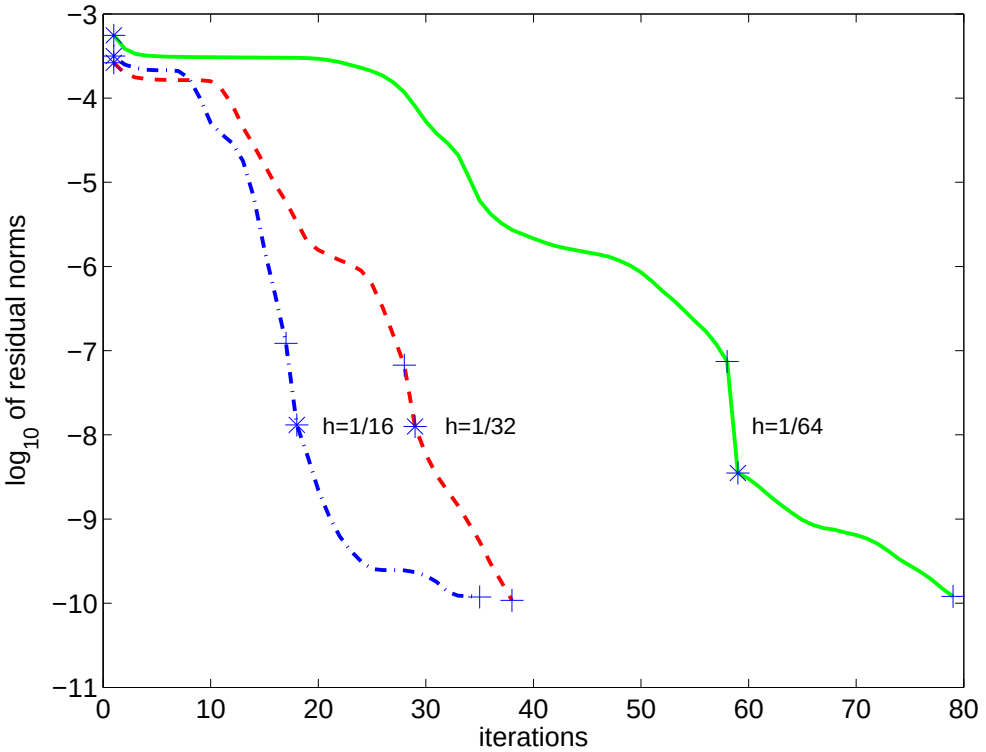

Fig. 12. Convergence behavior of the GMRES method in the FAS multigrid algorithm for the Brusselator equations with $\lambda \approx 29.19$.

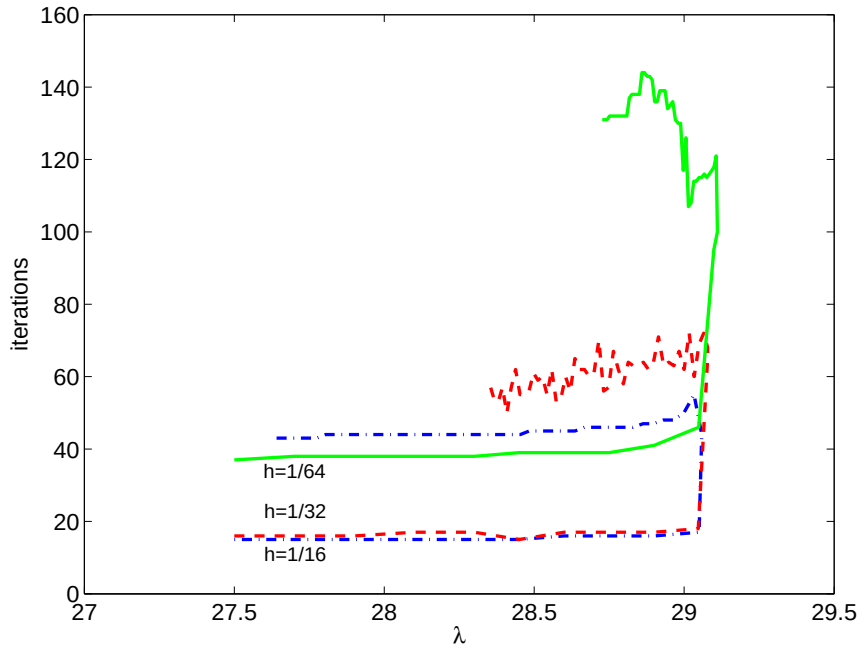

Fig. 13. Number of iterations of the GMRES algorithm versus the parameter λ in the FAS scheme, upper solution branch.

Table 7. Implementing the Bi-CGSTAB algorithm with preconditioner on the fine grid for various multigrid schemes.

Methods	V-cycle	W-cycle	FMG
Time	38500.8	205948.9	52995.7
p-cycle	1.6500	1.9333	1.9500
c-cycle	1.6441	1.9492	1.9661

Table 8. Implementing the GMRES algorithm with preconditioner on the fine grid for various multigrid schemes.

Methods	V-cycle	W-cycle	FMG
Time	24321.7	25075.2	37863.8
p-cycle	1.7833	1.8833	1.8167
c-cycle	1.7797	1.8983	1.8136

number of iterations versus the continuation parameter λ in the V-cycle scheme. Tables 5 and 6 show the numerical results of implementing Algorithms 4.3–4.4 with preconditioned GMRES as the relaxation method on the finest grid. Figure 10 shows the convergence behavior of the GMRES method in Algorithm 4.3 at $\lambda \approx 29.05$. Figure 11 displays the number of iterations versus the continuation parameter λ in the W-cycle scheme, where the upper solution branch is traced. Figures 12 and 13 show similar results for the FAS scheme with GMRES as the relaxation scheme. Tables 7 and 8 list the total execution time of implementing Algorithms 4.2–4.3 with mesh size $h = 1/64$ and 60 continuation steps for our test problem.

6. Conclusions

Based on the numerical results reported in Sec. 5, we wish to draw some concluding remarks concerning the performance of the numerical algorithms proposed in Sec. 4.

(i) If the mesh size is small enough, say $h = 1/64$, then the number of iterations of the Bi-CGSTAB and the GMRES methods increase sharply as we are close to the bifurcation point. Furthermore, the number of iterations decrease again as we are away from the bifurcation point. This means that we need more iterates to solve nearly singular linear systems. By comparing Figs. 3, 5, 7 and 9, 11, 13, we see that the GMRES method is more stable than the Bi-CGSTAB method in some neighborhood of the bifurcation point.

(ii) From Tables 7 and 8 we see that the V-cycle scheme is more efficient than the other two schemes for our test problem. Moreover, the V-cycle scheme

with the preconditioned GMRES method as the relaxation scheme is superior to the other algorithms described in this paper.

References

Allgower, E. L. & Georg, K. [1997] "Numerical path following," in *Handbook of Numerical Analysis*, Vol. 5, eds. Ciarlet, P. G. & Lions, J. L. (North-Holland), pp. 1–238.

Bolstad, J. H. & Keller, H. B. [1986] "A multigrid continuation method for elliptic problems with folds," *SIAM J. Sci. Stat. Comput.* **7**, 1081–1104.

Brandt, A. [1977] "Multi-level adaptive solutions to boundary value problems," *Math. Comput.* **31**, 333–390.

Brandt, A. [1982] *Guide to Multigrid Development, in Multigrid Methods*, Lecture Notes in Mathematics, Vol. 960, eds. Hackbusch, W. & Trottenberg, U. (Springer Verlag, NY), pp. 220–312.

Briggs, W. L., Henson, Van Emden, Henson & McCormic, S. F. [2000] *A Multigrid Tutorial*, 2nd edition (SIAM Publications, Philadelphia).

Brown, P. N. & Walker, H. F. [1997] "GMRES on (nearly) singular systems," *SIAM J. Matrix Anal. Appl.* **18**, 37–55.

Chan, T. F. & Keller, H. B. [1982] "Arc-length continuation and multi-grid techniques for nonlinear elliptic eigenvalue problems," *SIAM J. Sci. Stat. Comput.* **3**, 173–194.

Chang, S. L. & Chien, C. S. [2003a] "A multigrid-Lanczos algorithm for the numerical solutions of nonlinear eigenvalue problems," *Int. J. Bifurcation and Chaos* **13**, 1217–1228.

Chang, S.-L. & Chien, C.-S. [2003b] "Application of the Lanczos algorithm for solving the linear systems that occur in continuation problems," *Numer. Lin. Alg. Appl.* **10**, 335–355.

Chien, C. S. & Chen, M. H. [1998] "Multiple bifurcation in a reaction–diffusion problem," *Comput. Math. Appl.* **35**, 15–39.

Dangelmayr, G. [1987] "Degenerate bifurcations near a double eigenvalue in the Brusselator," *J. Austral. Math. Soc. Ser. B.* **28**, 486–535.

Desa, C., Irani, K. M., Ribbens, L. T., Watson, L. T. & Walker, H. F. [1992] "Preconditioned iterative methods for homotopy curve tracking," *SIAM J. Sci. Stat. Comput.* **13**, 30–46.

Doedel, E. J., Champneys, A. R., Fairgrieve, T. F., Kuznetsov, Y. A., Sandstede, B. & Wang, X. J. [1997] "AUTO97: Continuation and bifurcation software for ordinary differential equations (with HomCont)," User's Guide, Computer Science Department, Concordia University, Montreal, Canada.

Doedel, E. J., Paffenroth, R. C., Champneys, A. R., Fairgrieve, T. F., Kuznetsov, Y. A., Sandstede, B. & Wang, X. J. [2000] "AUTO2000: Continuation and bifurcation software for ordinary differential equations," Report, Applied Mathematics, Caltech, Pasadena.

Eisenstat, S. C., Elman, H. C. & Schultz, M. H. [1983] "Variational iterative method for solving consecutive symmetric positive definite linear systems," *SIAM J. Matrix Anal. Appl.* **21**, 1279–1299.

Govaerts, W. J. F. [2000] *Numerical Methods for Bifurcations of Dynamical Equilibria* (SIAM Publications, Philadelphia).

Hackbusch, W. [1979] "On the computation of approximate eigenvalues and eigenfunctions of elliptic operators by means of a multi-grid method," *SIAM J. Numer. Anal.* **16**, 201–215.

Hackbusch, W. [1985] *Multigrid Methods and Applications* (Springer-Verlag, Berlin).

Keller, H. B. [1987] *Lectures on Numerical Methods in Bifurcation Problems* (Springer, Berlin).

Mittelmann, H. D. & Weber, H. [1985] "Multi-grid solution of simple bifurcation problems," *SIAM J. Sci. Stat. Comput.* **6**, 49–60.

Nicolis, G. & Prigogine, I. [1977] *Self-Organization in Nonequilibrium Systems* (John Wiley and Sons, NY).

Saad, Y. & Schultz, M. H. [1986] "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.* **7**, 856–869.

Schaeffer, D. G. & Golubitsky, M. [1981] "Bifurcation analysis near a double eigenvalue of a model chemical reaction," *Arch. Rat. Mech. Anal.* **75**, 315–347.

Sonneveld, P. [1989] "CGS: A fast Lanczos-type solver for nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.* **10**, 36–52.

Van Der Vorst, H. A. [1992] "BI-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.* **13**, 631–644.

Weber, H. [1985] "Multigrid bifurcation iteration," *SIAM J. Numer. Anal.* **22**, 262–279.