

# AN ADAPTATION OF BICGSTAB FOR NONLINEAR BIOLOGICAL SYSTEMS

E. Venturino,<sup>1</sup> P. R. Graves-Morris,<sup>2</sup> and A. De Rossi<sup>1</sup>

<sup>1</sup>*Università di Torino, Dipartimento di Matematica, via Carlo Alberto 10, I-10123, Torino, Italy, {ezio.venturino,alessandra.derossi}@unito.it\**, <sup>2</sup>*University of Bradford, Department of Computing, Bradford BD7 1DP, UK, p.r.graves-morris@bradford.ac.uk*

**Abstract** Here we propose a new adaptation of Van der Vorst's BiCGStab to nonlinear systems, a method combining the iterative features of both sparse linear system solvers, such as BiCGStab, and of nonlinear systems, which in general are linearized by forming Jacobians, and whose resulting system usually involves the use of a linear solver. We consider the feasibility and efficiency of the proposed method in the context of a space-diffusive population model, the growth of which depends nonlinearly on the density itself.

**keywords:** BiCGStab, iterative methods, population models, sparse nonlinear systems.

## 1. Introduction

Many popular methods for the solution of a sparse system of linear equations such as BiCGStab are iterative. Methods for the solution of nonlinear systems are usually recursive and the recursion usually involves forming Jacobians for linearization of the nonlinear terms. The solution of a large sparse nonlinear system usually involves the use of a linear solver after the Jacobian has been formed.

The task of solving sparse systems of linear or nonlinear equations comes up in many large-scale problems of scientific computing. The iterative approach to the solution of large linear systems is preferable to the direct one in some situations, especially when we have to solve problems arising from applications in which the coefficient matrix is sparse. Among the many existing iterative methods, the Lanczos-Type Product Methods (LTPMs) are characterized by residual polynomials that are products of a Lanczos polynomial and another

\*Paper written with financial support of the Dipartimento di Matematica, Progetto per la ricerca locale "Modelli matematici avanzati per le applicazioni: studio analitico e numerico II" and of the M.I.U.R. of Italy (P.R.I.N. 2003).

Please use the following format when citing this chapter:

Author(s) [insert Last name, First-name initial(s)], 2006, in IFIP International Federation for Information Processing, Volume 199, System Modeling and Optimization, eds. Ceragioli F., Dontchev A., Furuta H., Marti K., Pandolfi L., (Boston: Springer), pp. [insert page numbers].

polynomial of the same degree. LTPMs enjoy some remarkable properties: in fact, they incorporate stabilization, so as to reduce and smooth the residuals as much as possible; they are transpose-free and gain one dimension of the Krylov space per matrix-vector multiplication. For a good survey of LTPMs, see for example [10].

A subset of the LTPMs is given by the BiCGStab family. The biconjugate gradient method (BiCG) has the property that the estimates of the solution of (1) below have residuals that are orthogonal to dual (or shadow) Krylov subspaces which increase in dimension with  $n$  and this feature is retained *implicitly* by LTPMs. In 1992, Van der Vorst proposed the use of the BiCGStab method so as to get smoother convergence of the estimates of the solution of (1), see [13]. In the BiCGStab algorithm, the stabilising polynomials are built up in factored form, with a new linear factor being included at each step in a way such that the residual undergoes a one-dimensional minimization process. This basic stabilisation was soon advanced to BiCGStab $l$  and BiCGStab( $l$ ) to help to avoid possible breakdowns, see [10].

In 1974, Gragg observed a connection between BiCG and vector-Padé approximation [5]. Vector-Padé methods in general construct vector-valued rational functions which approximate functions specified by their vector-valued power series. The idea is usually to accelerate the convergence of the given vector-valued power series (see [1]). A linear system of equations is often denoted by

$$Ax = b. \quad (1)$$

Many models arising in science and engineering require the solution of large sparse systems of non-linear equations. We consider the feasibility and efficiency of the methods proposed here in the context of models of a population whose growth depends non-linearly on the density of the population and we allow for diffusion of the population through space.

## 2. The method

A basic iterative method [11] for the solution of the nonlinear system of equations

$$F(x) = 0, \text{ with } F : \mathcal{R}^d \rightarrow \mathcal{R}^d. \quad (2)$$

may summarily be expressed by

$$z_{k+1} = S(z_k) = z_k + \Pi F(z_k), k = 0, 1, 2, \dots, \quad (3)$$

where  $S$  is the successor functional, and  $\Pi$  is an approximation to the Jacobian of the system

$$\Pi \approx -J_F^{-1}, \text{ with } J_F \equiv \frac{\partial F(z_k)}{\partial z}. \quad (4)$$

We develop acceleration methods for (3), not requiring explicit evaluation of the sequence  $\{z_k\}$ . A sequence  $\{x_k\}$  of estimates of the root will instead be formed using three-term inhomogeneous recursion relations and an optimal successor functional. To assess the accuracy of a successor functional, use its actual definition and let its residual be

$$R(z) = S(z) - z = \Pi F(z). \tag{5}$$

Thus  $R(z)$  is a residual *preconditioned* by  $\Pi$ , while  $F(z)$ , given algebraically by (2), denotes the vector residual of the original system.

The core three-term inhomogeneous recursion relations, designed originally for convergence acceleration of linear systems, are taken directly from VPAStab and here adapted to a nonlinear system [7]. Define preconditioned residuals of the estimates  $x_i$  by

$$r_i = R(x_i) = S(x_i) - x_i, \text{ for } i = 0, 1, 2, \dots \tag{6}$$

with initializations given by  $x_0 = z_0$  and  $x_1 = z_1 = S(x_0)$ .

The VPAStab recursion formulas are [7]

$$x_{2k} = (1 + \alpha_k)x_{2k-1} - \alpha_k x_{2k-2} + (1 - \theta_k)[(1 + \alpha_k)r_{2k-1} - \alpha_k r_{2k-2}], \tag{7}$$

$$x_{2k+1} = (1 + \beta_k)x_{2k} - \beta_k x_{2k-1} + (1 + \beta_k)r_{2k} - \beta_k(1 - \theta_k)r_{2k-1}, \tag{8}$$

with coefficients  $\alpha_k, \beta_k$  defined by

$$\alpha_k = e_{2k-1}(e_{2k-2} - e_{2k-1})^{-1}, k = 1, 2, 3, \dots, \tag{9}$$

$$\beta_k = e_{2k}(e_{2k-1}(1 - \theta_k) - e_{2k})^{-1}, k = 1, 2, 3, \dots, \tag{10}$$

where

$$e_N := w^T r_N, N = 0, 1, 2, \dots \tag{11}$$

For the vector  $w$  we can take  $w = r_0$ .

For  $k = 1, 2, \dots$ , each parameter  $\theta_k$  is chosen to minimise an estimate of  $r_{2k}$ . Grouping the terms in equn (7)

$$\bar{x}_{2k} := (1 + \alpha_k)x_{2k-1} - \alpha_k x_{2k-2}, \tag{12}$$

$$\bar{r}_{2k} := (1 + \alpha_k)r_{2k-1} - \alpha_k r_{2k-2}. \tag{13}$$

and starting with the definition (6), we find that

$$r_{2k} \approx R(\bar{x}_{2k} + (1 - \theta_k)\bar{r}_{2k}) \tag{14}$$

We estimate  $r_{2k}$  using first order Taylor expansion of the right-hand side of (14), under the assumption that the residual terms in (7) are smaller in norm than the estimates of  $x$ . Thus

$$r_{2k} \approx R(\bar{x}_{2k}) + (1 - \theta_k) \frac{\partial R(\bar{x}_{2k})}{\partial z} \bar{r}_{2k}. \quad (15)$$

The matrix  $J_R(z) = \partial R / \partial z$  is the Jacobian of this preconditioned residual, and it is only the product  $J_R(\bar{x}_{2k}) \bar{r}_{2k}$  that is required in (15). The value of  $\theta_k$  is determined by minimising the right-hand side of (15) in norm. A suitable choice of the preconditioning operator  $\Pi$  still has to be made. Equation (3) allows dynamical updating during iteration. Then equations (5) - (15) complete the specification of the nonlinear algorithm.

### 3. An application

We consider the feasibility and efficiency of the method proposed in the context of models of a population diffusing through space whose growth depends nonlinearly on the density itself ([8] and [9]).

Let  $\rho(t, \mathbf{x})$  denote the density at time  $t$  of a population living in an environment, the spatial variable being represented by the vector  $\mathbf{x}$ . The density is assumed to reproduce and diffuse along the space direction. For simplicity here we first formulate the case of one spatial dimension and then outline the case of two spatial dimension  $x \in \mathcal{R}^2$ , although our framework is also designed for  $x \in \mathcal{R}^3$ .

#### 3.1 The uni-dimensional case

We have

$$\frac{\partial \rho}{\partial t} = f(\rho) + \epsilon \frac{\partial^2 \rho}{\partial x^2}, \quad (16)$$

where  $\epsilon$  denotes the diffusion coefficient and the function  $f$  expresses the reproduction mechanism. One of the most frequently used forms is the logistic law, which gives a nonlinear model:

$$f(\rho) \equiv r\rho(t, x) \left[ 1 - \frac{\rho(t, x)}{K} \right], \quad (17)$$

where  $K$  is the carrying capacity and  $r$  is the reproduction rate. Thus the complete one dimensional model reads

$$\frac{\partial \rho}{\partial t} = r\rho(t, x) \left[ 1 - \frac{\rho(t, x)}{K} \right] + \epsilon \frac{\partial^2 \rho}{\partial x^2}, \quad (18)$$

where  $x \in [a, b]$ . The discretization method used is the Crank-Nicholson formula, which evaluates the equation at a suitable point  $P$  (see [12]). Letting

the grid  $(t_n, x_i) \equiv (nk, ih)$  where  $k$  and  $h$  are respectively the time and space discretization stepsizes,  $i = 1, 2, \dots, N-1$ , and the evaluation point  $P$  is taken as the time midpoint in the mesh,  $(\frac{1}{2}(t_n + t_{n+1}), x_i)$ . Letting  $\rho_{n,i} \equiv \rho(t_n, x_i)$ , the equation gets discretized as follows

$$\frac{1}{k}[\rho_{n+1,i} - \rho_{n,i}] = \frac{\epsilon}{2} \left[ \frac{1}{h_x^2}(\rho_{n+1,i-1} - 2\rho_{n+1,i} + \rho_{n+1,i+1}) + \frac{1}{h_x^2}(\rho_{n,i-1} - 2\rho_{n,i} + \rho_{n,i+1}) \right] + \frac{1}{2}[f(\rho_{n+1,i}) + f(\rho_{n,i})] \quad (19)$$

so that, introducing as a shorthand  $q_x \equiv \frac{k}{h_x^2}$ , the system reads finally

$$-\epsilon \frac{q_x}{2} \rho_{n+1,i-1} + \rho_{n+1,i} [1 + \epsilon q_x - \frac{rk}{2}] - \epsilon \frac{q_x}{2} \rho_{n+1,i+1} + \frac{rk}{2K} (\rho_{n+1,i})^2 \quad (20)$$

$$= \epsilon \frac{q_x}{2} \rho_{n,i-1} + \rho_{n,i} [1 - \epsilon q_x + \frac{rk}{2}] + \epsilon \frac{q_x}{2} \rho_{n,i+1} - \frac{rk}{2K} (\rho_{n,i})^2 \quad (21)$$

and in matrix form

$$A\rho_{(n+1)} + \lambda[\rho_{(n+1)}]^2 = B\rho_{(n)} + \lambda[\rho_{(n)}]^2, \quad \forall n = 0, 1, 2, \dots, \quad (22)$$

where

$$\lambda = \frac{rk}{2K}, \quad \rho_{(n)} = (\rho_{n,1}, \rho_{n,2}, \dots, \rho_{n,N-1})^t, \quad (23)$$

$$A = \begin{bmatrix} 1 + \epsilon q - \frac{rk}{2} & -\epsilon \frac{q}{2} & 0 & \dots & 0 \\ -\epsilon \frac{q}{2} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\epsilon \frac{q}{2} \\ 0 & \dots & 0 & -\epsilon \frac{q}{2} & 1 + \epsilon q - \frac{rk}{2} \end{bmatrix}, \quad (24)$$

and

$$B = \begin{bmatrix} 1 - \epsilon q + \frac{rk}{2} & \epsilon \frac{q}{2} & 0 & \dots & 0 \\ \epsilon \frac{q}{2} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \epsilon \frac{q}{2} \\ 0 & \dots & 0 & \epsilon \frac{q}{2} & 1 - \epsilon q + \frac{rk}{2} \end{bmatrix}. \quad (25)$$

The matrices  $A$  and  $B$  have dimension  $N \times N$ . Suitable boundary conditions are used.

For simplicity, we can also write

$$A\rho + \lambda\rho^2 = b \quad (26)$$

Introducing an extra index  $m$  to count the iterations we then define the iteration scheme

$$A\rho_{(n+1)}^{(m+1)} = B\rho_{(n)} + \lambda[\rho_{(n)}]^2 - \lambda[\rho_{(n+1)}^{(m)}]^2, \quad m \geq 1. \quad (27)$$

Now, we consider the splitting  $A = L + U$  and the iteration then becomes

$$L\rho_{(n+1)}^{(m+1)} = B\rho_{(n)} + \lambda[\rho_{(n)}]^2 - U[\rho_{(n+1)}^{(m)}]^2 - \lambda[\rho_{(n+1)}^{(m)}]^2, \quad m \geq 1. \quad (28)$$

The solution is obviously obtained by operating on the right hand side by  $L^{-1}$  in the usual sequential way.

Another iteration scheme which we consider is given by

$$(LU)\rho_{(n+1)}^{(m+1)} = (LU - A)\rho_{(n)}^{(m)} - \lambda\rho_{(n)}^{(m)}, \quad (29)$$

where one could also use an incomplete  $LU$  decomposition of the Jacobian (incomplete  $LU$  preconditioning)

$$(LU)_{\text{inc}} \approx A + 2\lambda \text{diag} \rho. \quad (30)$$

Then, we obviously obtain the solution as if by using  $U_{\text{inc}}^{-1}L_{\text{inc}}^{-1}$ , but without explicitly forming the inverse.

### 3.2 The two-dimensional case

The two-dimensional case is a simple extension of the former one. We consider the solution of the problem in the square  $[0, 1] \times [0, 1]$ . Letting the grid  $(t_n, x_i, y_j) \equiv (nk, ih_x, jh_y)$ , where  $k$  is the time stepsize and  $h_x, h_y$  are respectively the step-sizes along the  $x$  and  $y$  axes, the evaluation point  $P$  will be the average of the Laplacian, discretized via the five point formula at time  $n$  and  $n + 1$ . Letting  $\rho_{n,i,j} \equiv \rho(t_n, x_i, y_j)$ , and then  $q_x \equiv \frac{k}{h_x^2}$  and  $q_y \equiv \frac{k}{h_y^2}$ , the discretized equation gives the nonlinear system

$$\begin{aligned} & -\epsilon \frac{q_y}{2} \rho_{n+1,i,j-1} - \epsilon \frac{q_x}{2} \rho_{n+1,i-1,j} + \rho_{n+1,i,j} \left[ 1 + \epsilon q_x + \epsilon q_y - \frac{rk}{2} \right] \\ & \quad - \epsilon \frac{q_x}{2} \rho_{n+1,i+1,j} - \epsilon \frac{q_y}{2} \rho_{n+1,i,j+1} + \frac{rk}{2K} (\rho_{n+1,i,j})^2 \\ & = \epsilon \frac{q_y}{2} \rho_{n,i,j-1} + \epsilon \frac{q_x}{2} \rho_{n,i-1,j} + \rho_{n,i,j} \left[ 1 - \epsilon q_x - \epsilon q_y + \frac{rk}{2} \right] \\ & \quad + \epsilon \frac{q_x}{2} \rho_{n,i+1,j} + \epsilon \frac{q_y}{2} \rho_{n,i,j+1} - \frac{rk}{2K} (\rho_{n,i,j})^2 \end{aligned} \quad (31)$$

Notice that the matrices (24) and (25) are here replaced by band matrices.

### 4. Numerical results

We conducted experiments to compare the following iteration schemes:

- $(L + U)$  not accelerated,
- $(L + U) + \text{Nonlinear VPAStab}$ ,
- $(LU) + \text{Nonlinear VPAStab}$ ,

and to compare how different choices of the biological parameters affect the various schemes. After performing an extensive experimentation in dimension one, we concentrated our attention on the two-dimensional case, the results of which are here presented.

We also considered different boundary conditions: *Dirichlet BCs* and *Neumann BCs*. The computations were done on a workstation running an AMD Athlon XP 1500+ with 576 MB of RAM. The iteration is initialized with

$$\rho_{0,i,j} = p_0(1 - y_j), \quad i, j = 1, 2, \dots, N,$$

where  $p_0 = 275$ , and the iterations were stopped when  $\|r^{(k)}\|_2 < \text{tol}$ , where  $\text{tol}$  is a given tolerance. As a failure criterion we used either too slow convergence, or a maximum number of nonlinear iterations per timestep (250). In the experiments the numerical solutions were computed with timestep  $k = 10^{-2}$ .

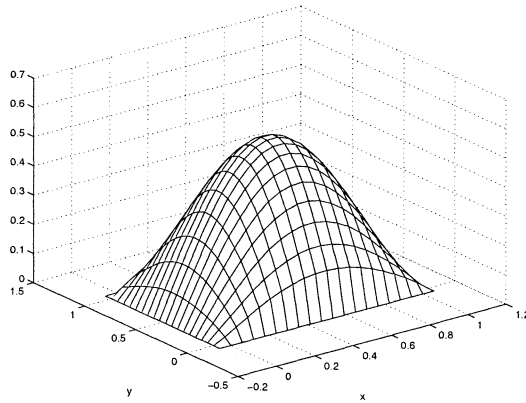


Figure 1. Sample of a numerical solution with Dirichlet BCs,  $r = 10$ ,  $K = 100$ ,  $\epsilon = 1$  and  $N = 17$ .

Since we are interested in evaluation and comparison of the computational cost of the various methods, the following counters are given in tables and figures:

- NI, the number of nonlinear iterations;
- FC, the number of function calls to the residual;
- ET, the execution time.

If we look at the number of nonlinear iterations, we note that acceleration reduces this number. Consequently, the execution times are also greatly reduced in all cases to which we apply acceleration.

In Figures 2 and 3, we respectively show how the number of nonlinear iterations and the execution time vary versus to the required accuracy (specified tolerance) expressed as  $\log_{10}(tol)$ .

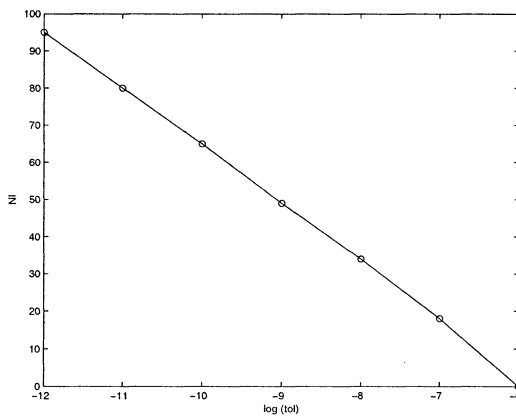


Figure 2. Plot of the number of nonlinear iterations against  $\log_{10}(tol)$  for the accelerated LU scheme.

In Figure 3 (left), execution times for the various methods are shown, versus the number  $N$  of discretization points in the mesh. We point out that the  $LU$  iteration scheme requires a higher execution time than the  $L + U$  one, while, in general, it needs fewer nonlinear iterations.

We compare the number of nonlinear iterations for different chosen values of the biological parameters, the reproduction rate  $r$  and the carrying capacity  $K$ . Results are shown in Figure 4, in Table 1 for different reproduction rates ( $r$ ) and in Table 2 for different carrying capacities ( $K$ ).

## References

- [1] G. A. Baker, P. R. Graves Morris. *Padé Approximants*. Cambridge Univ. Press, Cambridge, 1997.
- [2] S. C. Eisenstat, H. F. Walker. Globally convergent inexact Newton methods. *SIAM J. Optimization* 4:393-422, 1994.



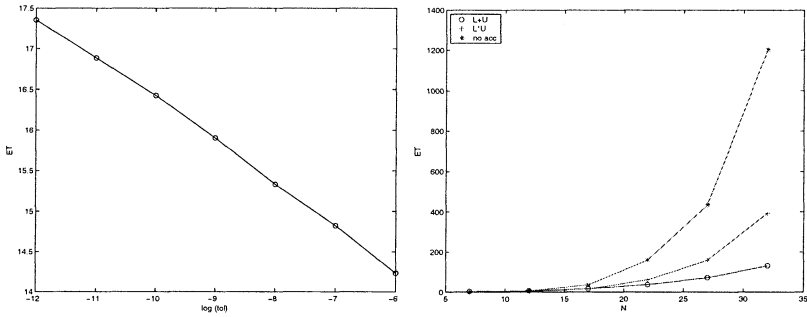


Figure 3. Execution times in seconds against  $\log_{10}(\text{tol})$  for the accelerated LU scheme (left) and for the various methods as functions of  $N$  (right).

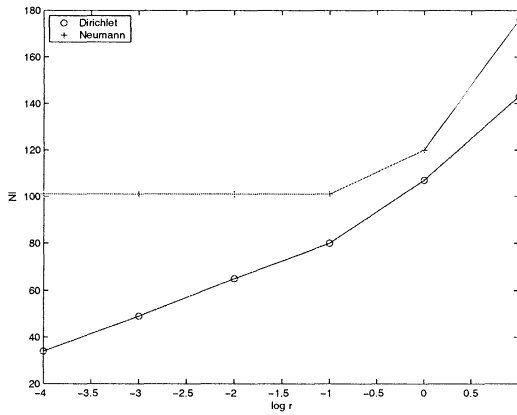


Figure 4. Nonlinear iterations for different values of  $\log_{10} r$  for the accelerated LU scheme with  $\text{tol} = 10^{-8}$ .

- [3] D. R. Fokkema, G. L. J. Sleijpen, H. A. Van der Vorst. Accelerated inexact Newton schemes for large systems of nonlinear equations. *SIAM J. Sci. Computing* 19:657-674, 1998.
- [4] R. Glowinski, H. B. Keller, L. Reinhart. Continuation-conjugate gradient methods for the least squares solution of nonlinear boundary value problems. *SIAM J. Sci. Stat. Comput.* 4:793-832, 1985.
- [5] W. B. Gragg. Matrix interpretations and applications of the continued fraction algorithm. *Rocky Mountain J. Math.* 4:213-225, 1974.
- [6] P. R. Graves-Morris. VPAStab: stabilized vector-Padé approximation with application to linear systems. *Numer. Algorithms* 33:293-304, 2003.
- [7] P. R. Graves-Morris. BiCGStab, VPAStab and an adaptation to mildly nonlinear systems. *Submitted*.

	$N \setminus r$	1.	0.1	0.01	0.001	0.0001
NI	11	122	107	95	76	58
FC		616	549	496	430	376
ET		1.4020	1.2820	1.2820	1.1320	1.2310
NI	21	122	94	74	58	44
FC		616	510	436	376	334
ET		16.153	15.232	14.321	13.940	13.439
NI	31	109	81	64	52	38
FC		571	469	404	358	316
ET		161.11	154.50	150.85	150.80	145.40

Table 1. Accelerated  $LU$  iteration scheme - Dirichlet BCs -  $\epsilon = 1$ ,  $K = 10$ ,  $\text{tol} = 10^{-8}$

	$N \setminus K$	1	10	100	1000	10000
NI	11	121	107	95	77	59
FC		610	549	496	433	379
ET		1.8730	1.2010	1.1620	1.1220	1.0610
NI	21	119	94	74	60	44
FC		603	510	436	382	334
ET		16.153	15.082	14.300	13.820	13.419
NI	31	102	81	64	52	39
FC		549	469	404	358	319
ET		160.97	154.15	149.95	146.83	144.51

Table 2. Accelerated  $LU$  iteration scheme - Dirichlet BCs -  $\epsilon = 1$ ,  $r = 0.1$ ,  $\text{tol} = 10^{-8}$

- [8] M. E. Gurtin, R. C. MacCamy. On the diffusion of biological populations. *Math. Biosci.* 33:35-49, 1977.
- [9] M. E. Gurtin, R. C. MacCamy. Diffusion models for age-structured populations. *Math. Biosci.* 54:49-59, 1981.
- [10] M. H. Gutknecht. Lanczos-type solvers for non-symmetric linear systems of equations. *Acta Numerica.* 6:271-397, 1997.
- [11] C. T. Kelley. *Iterative methods for linear and nonlinear equations.* Frontiers in Applied Mathematics 16, SIAM, Philadelphia, PA, 1995.
- [12] G. D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, 2nd Ed., Clarendon Press, Oxford, 1979.
- [13] H. A. Van der Vorst. BiCGStab: A fast and smoothly convergent variant of BiCG for the solution of non-symmetric linear systems. *SIAM J. Sci. Statist. Comput.* 13:631-644, 1992.