Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

Lecturas en Ciencias de la Computación
ISSN 1316-6239

Large scale continuation using a block eigensolver

Z. Castillo

RT 2006-03

Centro de Cálculo Científico y Tecnológico de la UCV
CCCT-UCV
Caracas, Febrero, 2006.

LARGE SCALE CONTINUATION USING A BLOCK EIGENSOLVER

Zenaida Castillo
zenaida@kuaimare.ciens.ucv.ve
Centro de Cálculo Científico y Tecnológico (CCCT),
Dpto. de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002,
Caracas 1041-A, Venezuela.

Abstract: *This paper presents a new algorithm to find and follow particular solutions of parameterized nonlinear systems. A block eigenvalue solver is embedded in a continuation framework for the computation of some eigenvalues of large Jacobian matrices depending on one parameter. Experiments on several problems show the reliability of the new approach in the accurate detection of critical points. As a byproduct, one obtains information about the stability of the process with no additional cost.*

Keywords: Arnoldi Krylov methods, Path Following, Parameter Continuation, Stability Analysis.

1. INTRODUCTION

A significant number of problems arising in scientific areas can be modeled by systems of partial differential equations (PDE). Typically these models are discretized in space, leading to large scale discrete dynamical nonlinear systems. Very often these nonlinear systems depend on parameters, and it is of great interest to study the behavior of solutions as the parameters are varied.
This work presents a new algorithm to solve and to analyze parameter dependent nonlinear systems of the form:

$$G(x,\alpha) = 0, \tag{1}$$

where $x \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$, and $G : \mathbb{R}^{n+1} \to \mathbb{R}^n$. The aim here is to develop a block eigenvalue solver, and embed it in a predictor-corrector continuation approach in order to analyze stability of steady state solutions of (1). Information obtained with this eigensolver is also used to solve a linear system at the predictor step. We use the block eigensolver to propagate a basis of eigenvectors from one continuation step to the next. In this way, path following is combined with stability and bifurcation analysis in the same procedure.

2. ARCLENGTH CONTINUATION

The basic purpose of a continuation method is to find solutions of (1) corresponding to particular values of α. Continuation methods are also designed to detect and compute special solutions that could be a sign of significant changes in the behavior of the process modeled by the PDE, and this procedure is known as *bifurcation analysis*. The detection and computation of special points is important because they indicate changes in the system, as for example, instability, existence of two or more branches converging or emanating from a single point, and existence of periodic solutions. Indeed, for many processes, this information can be used to change operation parameters and thereby improve performance. However, when n is very large, the computational cost of this additional procedure is generally prohibitive. The approach proposed in this work overcomes this problem.

The output of a continuation algorithm is a set of solutions of different nonlinear systems of the form given in (1), each one defined by a particular value of α. This set of solutions is then used to construct one or several *branches*. A *branch* is a connected curve consisting of points $(\alpha,[x])$ which represent the solutions of these different systems. Here $[x]$ is a scalar measure of the solution, as for example, $[x]=\|x\|$ or $[x]=x(i)$, for some i. The graph constructed with the set of pairs $(\alpha,[x])$ is known as a *bifurcation diagram* and it provides a graphical interpretation of the behavior of the solutions.

Asuming that an initial solution (x_0,α_0) is known, algorithm (1) enumerates the steps of a pseudoarclength continuation method to follow regular points:

Algorithm 1 Arclength Continuation

```
Evaluate derivatives at $x_0$:   $J = G_x(x_0,\alpha_0)$ ; $L = G_\alpha(x_0,\alpha_0)$
Compute initial tangent:   $Jd_x = -L$ ; $d_\alpha = 1/\sqrt{1+\|d_x\|^2}$
for i=1,2,...,nsteps do
    Predictor step: Solve $\begin{pmatrix} J & L \\ t_{0x} & t_{0\alpha} \end{pmatrix} \begin{pmatrix} t_{1x} \\ t_{1\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
    $x_1^0 = x_{i-1} + \frac{\Delta s}{\|t_1\|} t_{1x}$
    $\alpha_1^0 = \alpha_{i-1} + \frac{\Delta s}{\|t_1\|} t_{1\alpha}$
    Corrector Step:
    for j=1,2,...,until convergence do
        Solve   $G_x(x_i^j,\alpha_i^j)\Delta x = -G(x_i^j,\alpha_i^j)$
        $x_i^{j+1} = x_1^j + \Delta x^x$ ; $\alpha_i^{j+1} = \alpha_i^j + \Delta x^\alpha$
    end for
    Prepare the next step: Adapt step-size $\Delta s$
    $t_0 = t_1$ ; $J = G_x(x_j)$ ; $L = G_\alpha(x_j)$
end for
```

3. NEW CONTINUATION METHOD

In this section we present a new approach to compute particular solutions of a nonlinear system depending on a parameter and, at the same time, to predict special points and their stability. A disadvantage of traditional approaches is that they implement the continuation method and the linear stability analysis as two independent procedures, and do not allow interaction between them, thereby discarding useful information obtained after the computation of the eigenvalues. In contrast, our method is based on Krylov approaches; thus, a small number ($<< n$) of eigenvalues is computed by a procedure that has low memory requirements.

In each continuation step, a single Arnoldi based method successfully computes the eigenvalues of the Jacobian matrix J, producing after m iterations the factorization:

$$JV_m = V_m H_m + f e_m^T, \tag{2}$$

where the columns of $V \in \mathbb{R}^{n \times m}$ are an orthogonal basis for the associated Krylov subspace, and H_m is a Hessenberg matrix whose eigenvalues approximate those of J when the residual $\|f e_m^T y\|$ is sufficiently small, even if $\|f\|$ is not small, where $Hy = \lambda y$. In this case, the mth Arnoldi factorization of J leads $JV_m \approx V_m H_m$, where $m << n$. Instead of discarding this information, it can be used to solve the linear system $Jx = b$, since this is approximately equivalent to solve the easier system $H_m y = V_m^T b$.

The previous analysis is also valid for a block Arnoldi based method. In this case, after m block Arnoldi steps

$$JV_{[m]} = V_{[m]} H_{[m]} + F E_{[m]}^T, \tag{3}$$

where $V_{[m]} = [V_1, V_2, ..., V_m]$ is an $n \times (m \times k)$ matrix, having m orthogonal blocks of dimension $n \times k$, $H_{[m]}$ is a block-Hessenberg matrix, F is a matrix of order $n \times k$, and $E = [Z, .., Z, I_k]$ with Z a zero matrix of order k, and I_k is the identity matrix of order k.

In particular, we implement a block eigensolver where $m = 1$ and use it as part of the predictor step of a continuation code. In this way, the large linear system of order n that needs to be solved is replaced by a small system of order k (the block size), where $k \ << \ n$. This block eigenvalue solver, called BLIRAM hereafter, is based on the implicitly restarted Arnoldi method ([7]) and it was specifically designed for continuation algorithms. The output of BLIRAM is a 1-block Arnoldi factorization closely related to the basis of wanted eigenvectors for the next point in the continuation. Thus, we can expect this basis be a good starting matrix for the next BLIRAM iteration. This is the main reason preference is given to a block eigensolver instead of to a single vector one.

By embedding BLIRAM in a continuation method we are able not only to study the stability of the current solution and to detect special points, but also to propagate a basis for the desired invariant subspace. Moreover, in a predictor-corrector continuation procedure, linear systems have to be solved at the predictor and at the corrector step. Thus, the decomposition in (3) is useful for both steps.

Next, we discuss how to embed a block eigensolver in a predictor-corrector pseudoarclength continuation method, and how to do bifurcation and stability analysis during the continuation of solutions of (1).

The first equation of the system in step 4 of algorithm (1) is equivalent to

$$J\frac{t_{1x}}{t_{1\alpha}} = -L \quad \equiv \quad J\chi = -L, \tag{4}$$

while the second equation can be written as

$$t_{0x}\chi + t_{0\alpha} = \frac{1}{t_{1\alpha}}. \qquad (5)$$

If x_1 is a regular point, then $t_{1\alpha} \neq 0$; otherwise J would be singular and x_0 a critical point. Treatment of critical points is done in a different way, by solving an extended system, see [6] for details. Thus, (4) is well defined and after computing the solution χ, we can substitute it in (5) in order to compute $t_{1\alpha}$ as

$$t_{1\alpha} = \frac{1}{t_{0x}^T\chi + t_{0\alpha}}.$$

After that, we get t_{1x} by a direct computation:

$$\chi = \frac{t_{1x}}{t_{1\alpha}} \quad \Rightarrow \quad t_{1x} = t_{1\alpha}\chi.$$

The linear equation (4) is the base for introducing the block Arnoldi eigensolver in the continuation method. Given a 1-block Arnoldi factorization of J, we can consider $JV_{[1]} \approx V_{[1]}H_{[1]}$ and substitute this approximation into (4) in order to solve $JV_{[1]}y = -L$, where $V_{[1]}y = \chi$. This is equivalent to solve the smaller system:

$$H_{[1]}y = -V_{[1]}^T L. \qquad (6)$$

Hence, after solving the system (6), approximate solutions to (4) and (5) can be found, an consequently a new predictor (x_1^0, α_1^0). Previous ideas can be condensed in new statements in the computation of the predictor, These statements are outlined in algorithm (2).

Algorithm 2 Predictor with BLIRAM

```
Compute $k$ specified eigenvalues of $J$,   $JV_{[1]} \approx V_{[1]}H_{[1]}$
Study stability at the current point by checking the rigthmost eigenvalues
Check for special points: Bifurcation Analysis
Solve $H_{[1]}y = -V_{[1]}^T L$
$\chi = V_{[1]}y$
Compute a new tangent: $t_{1\alpha} = \frac{1}{\chi^T t_{0x} + t_{0\alpha}}$ ;   $t_{1x} = t_{1\alpha}\chi$
Compute a predictor: $x_1^0 = x_{i-1} + \frac{\Delta s}{\|t_1\|} t_{1x}$ ;   $\alpha_1^0 = \alpha_{i-1} + \frac{\Delta s}{\|t_1\|} t_{1\alpha}$
```

We propose a new arclength continuation method where the predictor step is given by this algorithm.

4. NUMERICAL RESULTS

In order to analyze the behavior of BLIRAM in a continuation framework we implemented (in MATLAB) an arclength continuation algorithm using the predictor outlined in algorithm (2). This predictor-corrector continuation code will be called PC-BLIRAM hereafter. Next, we present a comparison of PC-BLIRAM with CL-MATCONT, a MATLAB tool developed by Dhooge et. al [5]. Specifically we will use CL-MATCONT and PC-BLIRAM to find steady state solutions of a well known testing problem arising in chemical reactions: the Brusselator model.

Brusselator problem:

This is a model for a Belousov-Zhabotinsky reaction [2], which is described by the following system of PDEs:

$$\frac{\partial x}{\partial t} = \frac{D_1}{L^2}\frac{\partial^2 x}{\partial t^2} + \beta_1 - (\beta_2+1)x + x^2 y$$

$$\frac{\partial y}{\partial t} = \frac{D_2}{L^2}\frac{\partial^2 y}{\partial t^2} + \beta_2 x - x^2 y \tag{7}$$

with initial conditions: $x(0,r) = x_0(r); \quad y(0,r) = y_0(r)$. The variables r and t denote space and time respectively; x and y represent the concentration of the two reactants, while $\frac{D_1}{L^2}$ and $\frac{D_2}{L^2}$ are diffusion coefficients. The constants β_1 and β_2 come from the Dirichlet boundary conditions imposed at $r \in [0,1]$:

$$x(t,0) = x(t,1) = \beta_1; \qquad y(t,0) = y(t,1) = \beta_2/\beta_1.$$

A trivial solution of this system is $(x,y) = (\beta_1, \beta_2/\beta_1)$. The second order space derivative in (7) is approximated using a grid of N points in x and y, and the three-point difference formula

$$\frac{\partial^2 f}{\partial x^2} = \frac{1}{h^2}(f_{i-1} - 2f_i + f_{i+1}),$$

where $h = \frac{1}{N+1}$.

It is well known that near $D1 = 0.008$, $D2 = 0.004$, $\beta_1 = 2$, $\beta_2 = 5.45$, and $L = 0.51302$, the Jacobian matrix J has a pair of complex eigenvalues that cross the imaginary axis. Figure 1 depicts this situation. This figure shows the 30 rightmost eigenvalues of J for $L = 0.51302$. The red circles represent the six eigenvalues with the largest real part computed by BLIRAM. The interest in this problem is to determine values of L for which the rightmost eigenvalues of J are purely imaginary; these values represent Hopf bifurcations and signal the onset of periodic solutions. Therefore, chemists look for good continuation and bifurcation algorithms to detect these values of the parameter L. After discretization, equilibrium solutions are computed by solving a nonlinear system of the form

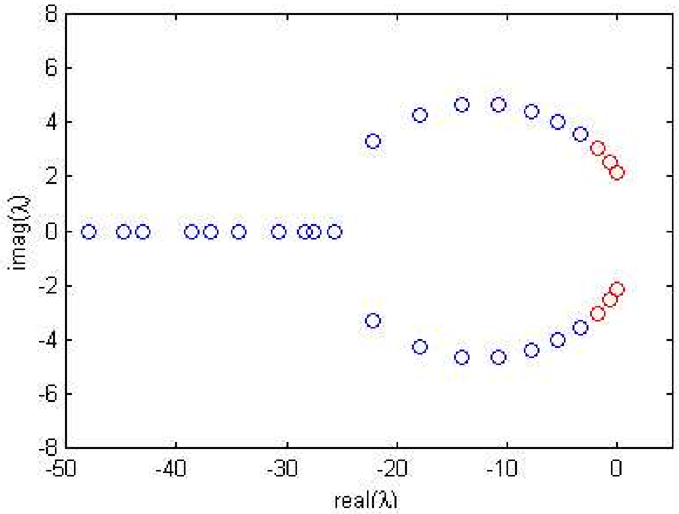

Figure 1. The 30 rightmost eigenvalues of the Brusselator matrix of order 2000.

$$G(X,L) = 0, \tag{8}$$

where X represents the concentration of the two reactants in the grid of points. The size of this problem is $2N+1$ and the Jacobian matrix is of order $n = 2N$. An approximate equilibrium for this system is:

$$\left\{ \begin{array}{l} x(r) = \beta_1 + 2\sin(\pi r) \\ y(r) = \frac{\beta_2}{\beta_1} - \frac{1}{2}\sin(\pi r). \end{array} \right. \tag{9}$$

Thus, a few Newton iterations starting from $(x,y) = (\beta_1, \beta_2/\beta_1)$ leads to an initial point x_0 that can be used to start a continuation procedure.

Test 1: In this test we compare CL-MATCONT and PC-BLIRAM in the continuation of equilibrium solutions of the discretized system that arises after the discretization of the Brusselator model. For this experiment we define $N = 40$, which means that for each continuation step a nonlinear system of the form given in (8) need to be solved. Here $G : \mathbb{R}^{81} \rightarrow \mathbb{R}$ and the Jacobian matrices are of order $n = 2N = 80$. The parameter L is varied in $[0.05, 0.4]$ and the continuation is stopped when a closed curve is found. Both, CL-MATCONT and PC-BLIRAM, start from the same point x_0. Since this problem is small, the block size and number of Arnoldi iteration in BLIRAM are chosen to be: $b = 4$ and $m = 6$ respectively. Thus, at each predictor step BLIRAM computes the 4 eigenvalues of the largest real part.

Results (Test 1): Both codes follow a path of solutions of the nonlinear system (8). For each equilibrium found, they plot the 40th component of the solution $X = [x; y]$ against the 81st component. The later represents the parameter value L. Figure 2 shows the bifurcation diagrams obtained for both codes. As we can see, both curves are very similar and contain the same information about bifurcation points (BP). However, the curve produced by PC-BLIRAM also contains information about the stability of each solution on the branch and position of turning points (T). Continuous line indicates stable steady states, while discontinuous line represents unstable equilibria. Thus, one advantage of using PC-BLIRAM is that we know whether or not the bifurcation involves a change in stability without any further calculation. The continuation of the first branch finishes when a closed curve is found or after the computation of a specified number of points. The output of both codes includes the set of computed points and the set of turning, bifurcation and Hopf points. In particular, when bifurcation points exist, the corresponding tangents of the emerging branches are also included in the output.

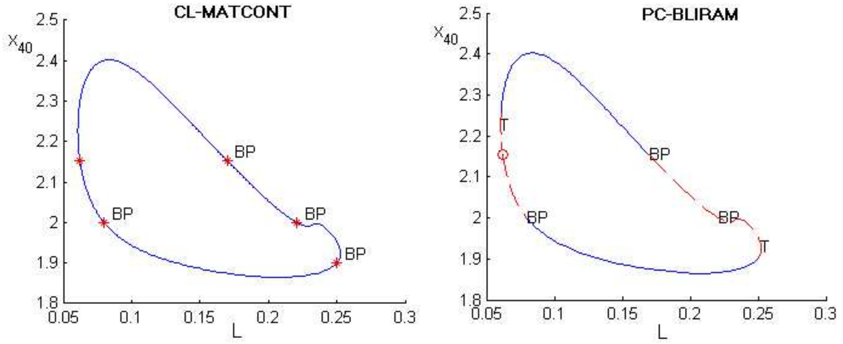

Figure 2. Comparison between CL-MATCONT and PC-BLIRAM. Both plots show position of Bifurcation points (BP). PC-BLIRAM also shows stability information.

Test 2: This test is designed to analize the ability of both codes to follow a new branch after computing a branching point. Thus, we use the bifurcation point represented by the point $(0.1697, 2.1532)$ in the figure 2 and the corresponding tangent. The parameters of BLIRAM remain the same.

Results (Test 2) : Figure 3 shows the bifurcation diagrams. Again both codes were able to follow the new branch and detect new bifurcation points. Additionally, the bifurcation diagram produced by PC-BLIRAM (rigth figure) reports changes in stability after each turning point; a situation that is very common in real applications.

Stability information given by PC-BLIRAM comes at a price. We need to solve an eigenvalue problem for each solution found. This makes a PC-BLIRAM continuation step more expensive than a CL-MATCONT continuation step. On the other hand, we notice that as the size of the problem increases, the difference in CPU time between the two codes decreases, while the difference in memory requirements increases. Table 1 supports this conclusion. This table shows CPU time and memory requirements for both codes. In all cases presented in this table, we keep $b = 4$ and $k = 6$. Because of the higher memory requirements of CL-MATCONT, after $n = 100$ the process run out of memory. This is mainly because CL-MATCONT checks the singularity of a bialternate matrix in order to predict Hopf points and the size of this bialternate matrix is $n(n-1)/2$. Hence, memory increase is of order n^2. Here is a second advantage of using PC-BLIRAM: since it is based on the Arnoldi method, BLIRAM needs additional memory mainly for the block Hessenberg matrix H, the block orthogonal Krylov basis V and the residual matrix F. Moreover, the number of columns of these matrices do not depend on the size of the problem, but the block-size b and the number of

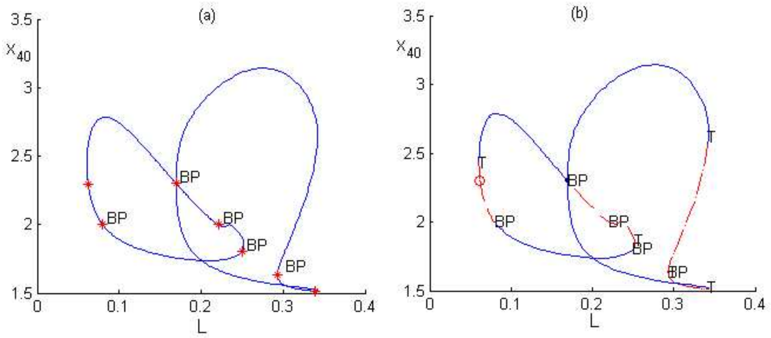

Figure 3. Tracing a new branch. (a) CL-MATCONT (b) PC-BLIRAM. Stability changes at each turning point are reported by PC-BLIRAM.

Table 1. CPU and memory requirements. CM=CL-MATCONT, PB=PC-BLIRAM

Jacobian-size	CPU(CM)	CPU(PB)	Memory(CM)	Memory(PB)
40	$9s$	45.3 s	72,696 KB	45,768 KB
50	$12.7s$	48.1 s	93,200 KB	46,854 KB
60	$13.8s$	53.8 s	143,122 KB	46,892 KB
70	$15.6s$	54.2 s	251,280 KB	47,076 KB
80	$20.1s$	56.85 s	367,106 KB	47,232 KB
90	24.9 s	60.79 s	562,174 KB	47,948 KB
100	N/A	$66.38s$	OUT OF MEMORY!!	48,015 KB

desired eigenvalues k, which are very small compared to n.

Now, not only is the computation of the predictor cheaper than in traditional approaches, but we can also take advantage of the continuation environment and use the current $V_{[1]}$ matrix as the initial block for the next application of the block Arnoldi eigensolver. In other words, we can carry the basis forward to the next continuation step.

5. FINAL REMARKS

A Matlab code was designed to illustrate the advantages of having an eigensolver embedded in a predictor-corrector method. This advantage has been previously presented by other authors, see for instance the works by Baglama et. al. [1] and by Calvetti and Reichel [3, 4] where large continuation problems are solved using an implicitly restarted block Lanczos method. Their results, like the results obtained with PC-BLIRAM, are encouraging. Moreover, results of the comparison between CL-MATCONT and PC-BLIRAM supports the idea that BLIRAM can be used to solve large continuation problems with a general Jacobian. Additional experiments on an industrial application will be presented in the near future.

Acknowledgements

This work was partially supported by CDCH-UCV project PI 03-00-5890-2005.

REFERENCES

[1]. J. Baglama, D. Calvetti, L. Reichel, and A. Ruttan. Computation of a few close eigenvalues of a large matrix with application to liquid crystal modeling. *J. Comput. Phys.*, 146:203–226, 1998.

[2]. I. Prigogine and R. Lefever. Stability and self-organization in open systems. In G. Nicolis and R. Lefever, editors, *Membranes, Dissipative Structures, and Evolution*, J. Wiley, New York, 1974.

[3]. D. Calvetti and L. Reichel. A block-Lanczos method for large continuation problems. *Numer. Algo.*, 21:109–118, 1999.

[4]. D. Calvetti and L. Reichel. Iterative methods for large continuation problems. *J. Comput. Appl. Math,*, 123:217–240, 2000.

[5]. A. Riet. A continuation toolboox in MATLAB. Master thesis, Mathematical Institute, Utrecht University, The Netherlands, 2000.
Available for download at http://www.math.uu.nl/people/kuznet/cm.
or at http://allserv.rug.ac.be/ajdhooge/research.html.

[6]. R. Seydel. *Practical Bifurcation and Stability Analysis.* Springer-Verlag, New York, second edition, 1994.

[7]. D.C. Sorensen. Implicit application of polynomial filters in a k-step arnoldi method. *SIAM J. Matrix. Anal. Appl.*, 13:357–385, 1992.