Linearized Initialization of the Newton Krylov Algorithm for Nonlinear Elliptic Problems

Sanjay Kumar Khattri

Stord/Haugesund University College,
Bjørnsonsgt. 45 Haugesund 5528, Norway
sanjay.khattri@hsh.no

Abstract. It is known that the Newton Krylov algorithm may not always converge if the initial assumption or initialization is far from the exact solution. We present a technique for initializing Newton Krylov solver for nonlinear elliptic problems. In this technique, initial guess is generated by solving linearised equation corresponding to the nonlinear equation. Here, nonlinear part is replaced by the equivalent linear part. Effectiveness of the technique is presented through numerical examples.

1 Introduction

The past fifty to sixty years have seen generous improvement in solving linear systems. Krylov subspace methods are the result of the tremendous effort by the researchers during the last century. It is one among the ten best algorithms of the 20th century. There exists optimal linear solvers [16]. But, still there is no optimal nonlinear solver, or the one that we know of. Our research is in the field of optimal solution of nonlinear equations generated by the discretization of the nonlinear elliptic equations [15], [14], [13], [12]. Let us consider the following nonlinear elliptic partial differential equation [15]

$$\operatorname{div}(-\mathrm{K}\operatorname{grad} p) + f(p) = s(x,y) \qquad \textbf{in} \quad \Omega \tag{1}$$

$$p(x,y) = p^{\mathbf{D}} \qquad \textbf{on} \quad \partial\Omega_{\mathbf{D}} \tag{2}$$

$$g(x,y) = (-\mathrm{K}\,\nabla p)\cdot \hat{\mathbf{n}} \qquad \textbf{on} \quad \partial\Omega_{\mathbf{N}} \tag{3}$$

Here, Ω is a polyhedral domain in $\mathbb{R}^{\mathrm{d}}$, the source function $s(x,y)$ is assumed to be in $L^2(\Omega)$, and the medium property K is uniformly positive. In the equations (2) and (3), $\partial\Omega_{\mathbf{D}}$ and $\partial\Omega_{\mathbf{N}}$ represent Dirichlet and Neumann part of the boundary, respectively. $f(p)$ represents nonlinear part of the equation. p is the unknown function. The equations (1), (2) and (3) models a wide variety of processes with practical applications. For example, pattern formation in biology, viscous fluid flow phenomena, chemical reactions, biomolecule electrostatics and crystal growth [9], [5], [6], [7], [8], [10].

There are various methods for discretizing the equations (1), (2) and (3). To mention a few: Finite Volume, Finite Element and Finite Difference methods [12]. These methods convert nonlinear partial differential equations into a system

M. Bubak et al. (Eds.): ICCS 2008, Part I, LNCS 5101, pp. 975–982, 2008.

of algebraic equations. We are using the Newton Krylov algorithm for solving the discrete nonlinear system of equations formed by the Finite Volume method [15]. Since, initial guess or initialization is very important for the convergence of the Newton's algorithm. Thus, for starting the Newton Krylov algorithm, we are solving the corresponding linearised equation, and use this solution as the initial guess for the Newton Krylov algorithm. The corresponding linearized equations to the nonlinear equaion (1) is $\operatorname{div}(-\mathrm{K}\operatorname{grad} p)+\overline{f(p)} = s$. Here, $\overline{f(p)}$ is the linear representation of the nonlinear part $f(p)$.

2 Newton Krylov Algorithm

For formulating Newton algorithm, equation (1) is discretized in the residual form [15]

$$\operatorname{div}(-\mathrm{K}\operatorname{grad} p) + f(p) - s = 0.$$

Let the discretization of the nonlinear partial differential equations result in a system of nonlinear algebraic equations $\mathbf{A}(\mathbf{p}) = 0$. Each cell in the mesh produces a nonlinear algebraic equation [15], [12]. Thus, discretization of the equations (1), (2) and (3) on a mesh with n cells result in n nonlinear equations, and let these equations are given as

$$\mathbf{A}(\mathbf{p}) = \begin{pmatrix} A_1(\mathbf{p}) \\ A_2(\mathbf{p}) \\ \vdots \\ A_n(\mathbf{p}) \end{pmatrix}. \tag{4}$$

We are interested in finding the vector $\mathbf{p}$ which makes the operator $\mathbf{A}$ vanish. The Taylors expansion of nonlinear operator $\mathbf{A}(\mathbf{p})$ around some initial guess $\mathbf{p}_0$ is

$$\mathbf{A}(\mathbf{p}) = \mathbf{A}(\mathbf{p}_0) + \boldsymbol{J}(\mathbf{p}_0)\, \Delta\mathbf{p} + hot, \tag{5}$$

where *hot* stands for higher order terms. That is, terms involving higher than the first power of $\Delta\mathbf{p}$. Here, difference vector $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}_0$. The Jacobian $\boldsymbol{J}$ is a $n \times n$ linear system evaluated at the $\mathbf{p}_0$. The Jacobian $\boldsymbol{J}$ in the equation (5) is given as follows

$$\boldsymbol{J} = \left[\frac{\partial A_i}{\partial p_j}\right] = \begin{pmatrix} \dfrac{\partial A_1}{\partial p_1} & \dfrac{\partial A_1}{\partial p_2} & \cdots & \dfrac{\partial A_1}{\partial p_n} \\ \dfrac{\partial A_2}{\partial p_1} & \dfrac{\partial A_2}{\partial p_2} & \cdots & \dfrac{\partial A_2}{\partial p_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial A_n}{\partial p_1} & \dfrac{\partial A_n}{\partial p_2} & \cdots & \dfrac{\partial A_n}{\partial p_n} \end{pmatrix}$$

Since, we are interested in the zeroth of the non-linear vector function $\mathbf{A}(\mathbf{p})$. Thus, setting the equation (5) equals to zero and neglecting higher order terms will result in the following well known Newton Iteration Method

$$\begin{aligned} \boldsymbol{J}(\mathbf{p}_k)\,\Delta\mathbf{p}_k &= -\mathbf{A}(\mathbf{p}_k), \\ \mathbf{p}_{k+1} &= \mathbf{p}_k + \Delta\mathbf{p}_{k+1}, \quad k = 0, \ldots, n. \end{aligned} \tag{6}$$

The linear system (6) is solved by the Conjugate Gradient algorithm [16]. The pseudo code is presented in the Algorithm 1. The presented algorithm have been implemented in the C^{++} language. Three stopping criteria are used in the Algorithm 1. The first criterion is the number of iterations. Second and third criteria are based on the residual vector, $\mathbf{A}(\mathbf{p})$ and difference vector $\Delta\mathbf{p}_k$. If the method is convergent, L_2 norm of the difference vector, $\Delta\mathbf{p}$, and the residual vector, $\mathbf{A}(\mathbf{p})$, converge to zero [see 11]. We are reporting convergence of both of these vectors. For better understanding the error reducing property of the method, we report variation of $\|\mathbf{A}(\mathbf{p_k})\|_{L_2}/\|\mathbf{A}(\mathbf{p_0})\|_{L_2}$ and $\|\Delta(\mathbf{p_k})\|_{L_2}/\|\Delta(\mathbf{p_0})\|_{L_2}$ with iterations (k).

Algorithm 1. Newton Krylov algorithm.

```
1 Mesh the domain;
2 Form the non-linear system, A(p);
3 Find initial guess p_0;
4 Set the counter k = 0 ;
5 while k ≤ max_iter or ‖Δp_k‖_{L2} ≤ tol or ‖A(p_k)‖_{L2} ≤ tol do
6 |   Solve the discrete system J(p_k)Δp_k = −A(p_k);
7 |   p_{k+1} = p_k + Δp_k;
8 |   k^{++};
9 end
```

Our research work is focus on the initialization step of the above algorithm. Initialization (step three of the Algorithm 1) is a very important part of the Newton Krylov algorithm.

3 Numerical Work

3.1 Example 1

Without loss of generality let us assume that K is unity, and the boundary is of Dirichlet type. Let $f(p)$ be $\gamma\ \exp(p)$. Thus, the equations (1), (2) and (3) are written as

$$-\nabla^2 p + \gamma\ \exp(p) = f \qquad \textbf{in} \quad \Omega, \tag{7}$$

$$p(x, y) = p^{\mathrm{D}} \qquad \textbf{on} \quad \partial\Omega_{\mathrm{D}}. \tag{8}$$

Here, γ is a scalar. Let γ be 100. For computing the true error and convergence behavior of the methods, let us further assume that the exact solution of the equations (7) and (8) is the following bubble function

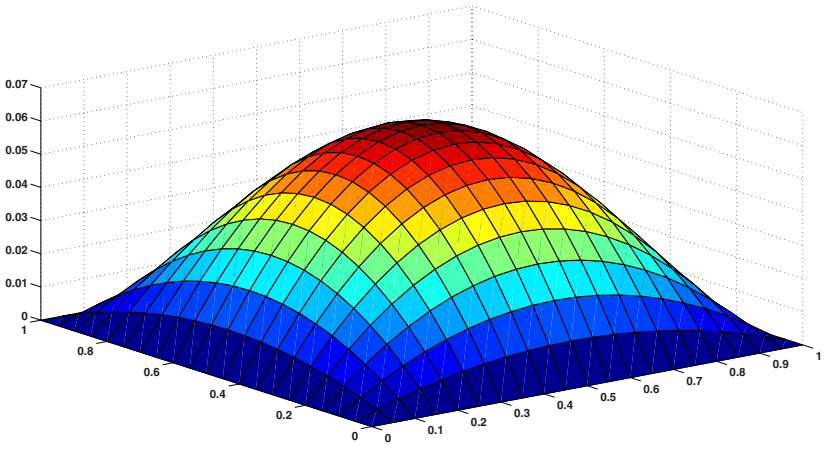


Fig. 1. Surface plot of the exact solution of example 3.1

$$p = x\,(x-1)\,y\,(y-1).$$

Let our domain be a unit square. Thus, $\Omega = [0,1] \times [0,1]$.

Figure 1 displays the surface plot of the exact solution. We are discretizing equations (7) and (8) on a 40×40 mesh by the method of Finite Volumes [11], [12], [13], [15]. Discretization results in a nonlinear algebraic vector (4) with 1600 nonlinear equations.

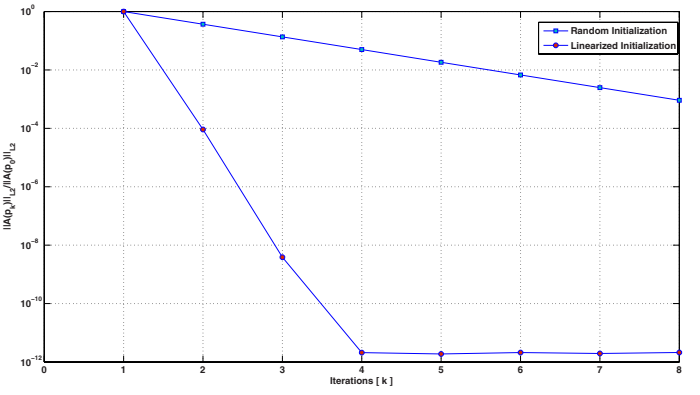
For making initial guess, we are using two approaches. In the first tradtional approach, we make a random initialization. The second approach is based on the linearization of the nonlinear part. Let us now form a linear approximation to the nonlinear part through Taylor series expansion. The Taylor series expansion of the nonlinear part (exponential funciton) is given as

$$\begin{aligned} e^p &= \sum_{i=0}^{\infty} \frac{p^i}{i}, \\ &= 1 + p + \frac{p^2}{2} + \frac{p^3}{3} + \cdots. \end{aligned}$$

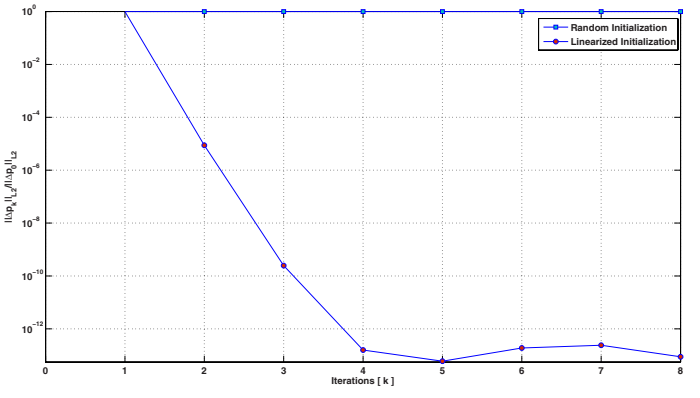
From the above expansion, the linear approximation of e^p is $(1+p)$. For forming a corresponding linearized equation to the nonlinear equation (7), we replace, e^p by $(1+p)$. Thus, for finding an initial guess for the Newton algorithm, we are solving the following corresponding linearised equation

$$-\nabla^2 p + \gamma\,(1+p) = f.$$

The Newton iteration for both of these initial guesses are reported in the Fig. 2(a). Figure 2(a) presents the convergence of the residual vector, while Fig. 2(b) presents the convergence of the difference vector for first eight



(a) Newton iteration vs $\|\mathbf{A}(\mathbf{p_k})\|_{L_2}$ for two different initialization.



(b) Newton Iteration vs $\|\Delta(\mathbf{p_k})\|_{L_2}$ for different initialization.

Fig. 2. Example 3.1

iterations. We are solving the Jacobian system by the ILU preconditioned Conjugate Gradient with a tolerance of 1×10^{-10}.

It is clear from the Figs. 2(a) and 2(b) that solving the corresponding linearized equation for the initial guess can make a big difference. With random initialization, the residual after five iterations is about $1/100$ of the initial residual. While with linearized initialization, the residual after five iteration is about $1/10^{12}$ of the initial residual. It is interesting to note in the Fig. 2(b), with random initialization the Newton Krylov algortithm is not converging in the L_2 norm of the difference vector. On the other hand, with a linearized initialization the Newton Krylov algorithm is still reducing the error in difference vector by $1/10^{12}$ of the initial error.

3.2 Example 2

Let us solve the following equations

$$-\nabla^2 p + \xi \sinh(\exp(\mathrm{p})) = f \qquad \textbf{in} \quad \Omega, \tag{9}$$

$$p(x,y) = p^{\mathrm{D}} \qquad \textbf{on} \quad \partial\Omega_{\mathrm{D}}. \tag{10}$$

Here, ξ is a scalar. We choose ξ to be 10. Let the exact solution be given as

$$p = \cos x + y\cos 3\,x - y + \cos x - y\sinh x + 3\,y + 5\,e^{-(x^2+y^2)/8}$$

Let our domain be a unit square. Thus, $\Omega = [0,1]\times[0,1]$.

Figure 3 portrays the surface plot of the exact solution. For forming a corresponding linearized equation. The Taylor series expansion of $\sinh(\exp(\mathrm{p}))$ around $p = 0$ is given as

$$\sinh e^p = \frac{1}{2}\,e - \frac{1}{2\,e} + \left(\frac{1}{2}\,e + \frac{1}{2\,e}\right) p + \frac{1}{2}\,e\,p^2 + \left(-\frac{1}{12e} + \frac{5\,e}{12}\right) p^3 + \ldots.$$

The above series expansion is found through the **Maple** by using the command "taylor(sinh(exp(p)), p = 0, 5)". From the above expansion, the linear approximation of $\sinh e^p$ is

$$\left(\frac{1}{2}\,e - \frac{1}{2\,e}\right) + \left(\frac{1}{2}\,e + \frac{1}{2\,e}\right) p.$$

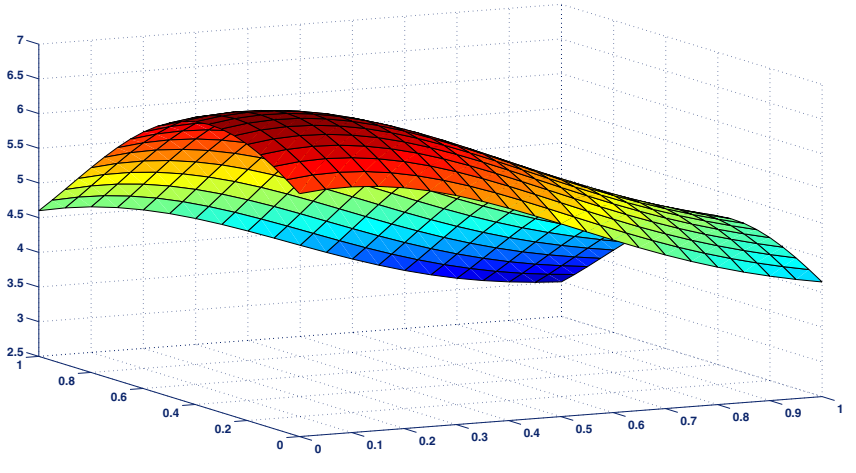


Fig. 3. Surface plot of the exact solution of example 3.2

For forming a corresponding linearized equation to the nonlinear equation (9), we replace, $\sinh e^p$ by $(1/2\, e - 1/2\, e) + (1/2\, e + 1/2\, e)\ p$. Thus, for finding an initial guess for the Newton algorithm, we are solving the following linearised equation

$$-\nabla^2 p + \xi \left[\frac{1}{2}\, e - \frac{1}{2\, e} + \left(\frac{1}{2}\, e + \frac{1}{2\, e} \right) p \right] = f.$$

4 Conclusions

Robust initialization of the Newton Krylov algorithm is very crucial for the convergence. Initialization plays very important role in the convergence of the Newton Krylov algorithm. We presented a technique for forming the initial guess. Numerical work shows that initializing the Newton Krylov algorithm through the solution of the corresponding linearized equation is computationally efficient.

Bibliography

[1] Khattri, S.K.: Newton-Krylov Algorithm with Adaptive Error Correction For the Poisson-Boltzmann Equation. MATCH Commun. Math. Comput. Chem. 1, 197–208 (2006)

[2] Khattri, S.K., Hellevang, H., Fladmark, G.E., Kvamme, B.: Simulation of long-term fate of CO_2 in the sand of Utsira. Journal of Porous Media (to be published)

[3] Khattri, S.K.: Grid generation and adaptation by functionals. Computational and Applied Mathematics 26, 1–15 (2007)

[4] Khattri, S.K.: Numerical Tools for Multicomponent, Multiphase, Reactive Processes: Flow of CO_2 in Porous Media. PhD Thesis, The University of Bergen (2006)

[5] Host, M., Kozack, R.E., Saied, F., Subramaniam, S.: Treatment of Electrostatic Effects in Proteins: Multigrid-based Newton Iterative Method for Solution of the Full Nonlinear Poisson-Boltzmann Equation. Proteins: Structure, Function, and Genetics 18, 231–245 (1994)

[6] Holst, M., Kozack, R., Saied, F., Subramaniam, S.: Protein electrostatics: Rapid multigrid-based Newton algorithm for solution of the full nonlinear Poisson-Boltzmann equation. J. of Bio. Struct. & Dyn. 11, 1437–1445 (1994)

[7] Holst, M., Kozack, R., Saied, F., Subramaniam, S.: Multigrid-based Newton iterative method for solving the full Nonlinear Poisson-Boltzmann equation. Biophys. J 66, A130–A130 (1994)

[8] Holst, M.: A robust and efficient numerical method for nonlinear protein modeling equations. Technical Report CRPC-94-9, Applied Mathematics and CRPC, California Institute of Technology (1994)

[9] Holst, M., Saied, F.: Multigrid solution of the Poisson-Boltzmann equation. J. Comput. Chem. 14, 105–113 (1993)

[10] M. Holst: MCLite: An Adaptive Multilevel Finite Element MATLAB Package for Scalar Nonlinear Elliptic Equations in the Plane. UCSD Technical report and guide to the MCLite software package. Available on line at, `http://scicomp.ucsd.edu/~mholst/pubs/publications.html`

[11] Khattri, S.: Convergence of an Adaptive Newton Algorithm. Int. Journal of Math. Analysis 1, 279–284 (2007)

[12] Khattri, S., Aavatsmark, I.: Numerical convergence on adaptive grids for control volume methods. The Journal of Numerical Methods for Partial Differential Equations 9999 (2007)
[13] Khattri, S.: Analyzing Finite Volume for Single Phase Flow in Porous Media. Journal of Porous Media 10, 109–123 (2007)
[14] Khattri, S., Fladmark, G.: Which Meshes Are Better Conditioned: Adaptive, Uniform, Locally Refined or Locally Adjusted? In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3992, pp. 102–105. Springer, Heidelberg (2006)
[15] S. Khattri, Nonlinear elliptic problems with the method of finite volumes. Differential Equations and Nonlinear Mechanics. Article ID 31797 (2006)
[16] van der Vorst, H.A.: Iterative Krylov Methods for Large Linear Systems. Cambridge monographs on applied and computational mathematics. Cambridge University Press, New York (2003)