Optimal Preconditioners for Interval Gauss–Seidel Methods

R. B. Kearfott and Xiaofa Shi

0 Introduction

Consider the following nonlinear system

$$F(X) = \begin{bmatrix} f_1(x_1, x_2, ..., x_n) \\ \vdots \\ f_n(x_1, x_2, ..., x_n) \end{bmatrix} = 0, \tag{1}$$

where bounds $\underline{x}_i$ and $\overline{x}_i$ are known such that

$$\underline{x}_i \le x_i \le \overline{x}_i, \quad \text{for } 1 \le i \le n.$$

We write $X = (x_1, x_2, ..., x_n)^T$ and denote by $\boldsymbol{B}$ the box given by the above inequalities on the variables x_i.

A general approach to such problems is to transform the nonlinear system $F(X) = 0$ to the interval linear system:

$$\boldsymbol{F}'(\boldsymbol{X}_k)(\tilde{\boldsymbol{X}}_k - \check{X}_k) \ni -F(\check{X}_k), \tag{2}$$

where $\boldsymbol{F}'(\boldsymbol{X}_k)$ is a suitable interval expansion of the Jacobi matrix over the box $\boldsymbol{X}_k$ ($\boldsymbol{X}_0 = \boldsymbol{B}$) and $\check{X}_k \in \boldsymbol{X}_k$ represents a predictor or initial guess point.

The general goal is to verify existence/uniqueness/non-existence of solutions in the box, and find all existing solutions.

1 What is a preconditioner? Why preconditioning?

Consider an interval linear system:

$$\boldsymbol{A}(\boldsymbol{X} - \check{X}) = \boldsymbol{b}.$$

We may multiply a matrix or a row vector Y to both sides of the above system. The matrix or the row vector Y is called a **preconditioner**. The main reason for preconditioning is to get improvement when solving the interval linear system.

Example 1 *Consider:*

$$\begin{bmatrix} 1 & [-1,0] \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

with initial box $[-1,1] \times [-1,1]$.

When we apply Gauss–Seidel steps to the above system, we obtain

$$\boldsymbol{x}_1 = -[-1,0] \cdot \boldsymbol{x}_2 = [-1,1],$$

$$\boldsymbol{x}_2 = -\boldsymbol{x}_1 = [-1,1].$$

Since neither coordinate interval has changed, the Gauss–Seidel method does not bring any improvement for this example.

On the other hand, let

$$Y = \begin{bmatrix} 2/3 & 1/3 \\ -2/3 & 2/3 \end{bmatrix}.$$

If we multiply both sides of the system by Y, we have

$$\begin{bmatrix} 1 & [-1/3,1/3] \\ 0 & [2/3,4/3] \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Now, if we apply Gauss–Seidel steps, we obtain

$$\boldsymbol{x}_1 = -[-1/3,1/3] \cdot \boldsymbol{x}_2 = [-1/3,1/3],$$

$$\boldsymbol{x}_2 = 0.$$

For x_1, we obtain a smaller interval. For x_2, we get an even better result, a point.

This example tells us that preconditioners are necessary for interval Gauss–Seidel steps.

In this paper, we will concentrate on optimal preconditioners, computed row-by-row, only. A preconditioned interval Gauss–Seidel method may be used to compute a new interval $\tilde{\boldsymbol{x}}_k$ for the k-th variable. Suppose $Y_k = (y_{k1}, y_{k2}, ..., y_{kn})$ is the preconditioner for x_k.

Algorithm 1 (Preconditioned Gauss–Seidel method)

1. Compute $Y_k \boldsymbol{F}' \cdot (\tilde{\boldsymbol{X}} - \check{X})$ *and* $-Y_k F$. *Then compute*

$$\tilde{\mathbf{x}}_k = \check{x}_k - \frac{\left[\sum_{i=1}^{n} y_{ki} f_i + \sum_{\substack{j=1 \\ j \neq k}}^{n} \left(\sum_{i=1}^{n} y_{ki} \boldsymbol{f}'_{ij} \right) (\boldsymbol{x}_j - \check{x}_j) \right]}{\sum_{i=1}^{n} y_{ki} \boldsymbol{f}'_{ik}}. \tag{3}$$

2. *If* $\tilde{\boldsymbol{x}}_k \cap \boldsymbol{x}_k = \emptyset$, *then return, indicating that there is no root of* F *in* $\boldsymbol{X}$.

3. *Replace* $\boldsymbol{x}_k$ *by* $\boldsymbol{x}_k \cap \tilde{\boldsymbol{x}}_k$.

2 Four different preconditioners

1. C^W preconditioner

A preconditioner Y_k^0 for x_k is called a C^W preconditioner, if

$$w(\tilde{\boldsymbol{x}}_k^0) = \min_{Y_k} w(\tilde{\boldsymbol{x}}_k),$$

where $w(\boldsymbol{x})$ is the width of interval $\boldsymbol{x}$, and

$$\tilde{\boldsymbol{x}}_k = \check{x}_k - \frac{\left[\sum_{i=1}^{n} y_{ki} f_i + \sum_{\substack{j=1\\ j\neq k}}^{n} \left(\sum_{i=1}^{n} y_{ki} \boldsymbol{f}'_{ij}\right)(\boldsymbol{x}_j - \check{x}_j)\right]}{\sum_{i=1}^{n} y_{ki} \boldsymbol{f}'_{ik}}.$$

Let $\tilde{\boldsymbol{x}}_i = \check{x}_k - \dfrac{\boldsymbol{n}_i(Y_i)}{\boldsymbol{d}_i(Y_i)}$. Then the C^W preconditioner Y_k^0 can be computed by solving the following optimization problem:

$$\min_{\underline{\boldsymbol{d}}_i(Y_i)=1} w(\boldsymbol{n}_i(Y_i)).$$

2. E^W preconditioner

Let $\tilde{\boldsymbol{x}}_i = \check{x}_k - \dfrac{\boldsymbol{n}_i(Y_i)}{\boldsymbol{d}_i(Y_i)}$. If $0 \in \boldsymbol{d}_i(Y_i)$ and $\boldsymbol{n}_i(Y_i) > 0$, then Kahan arithmetic gives

$$\frac{\boldsymbol{n}_i(Y_i)}{\boldsymbol{d}_i(Y_i)} = \left(-\infty, \frac{\underline{\boldsymbol{n}}_i(Y_i)}{\underline{\boldsymbol{d}}_i(Y_i)}\right] \bigcup \left[\frac{\underline{\boldsymbol{n}}_i(Y_i)}{\overline{\boldsymbol{d}}_i(Y_i)}, \infty\right).$$

A preconditioner Y_k^0 for x_k is called an E^W preconditioner if it is a solution of the following optimization problem:

$$\max_{\underline{\boldsymbol{n}}_i(Y_i)=1} w\left(\left[\frac{1}{\underline{\boldsymbol{d}}_i(Y_i)}, \frac{1}{\overline{\boldsymbol{d}}_i(Y_i)}\right]\right).$$

3. C^M preconditioner

A preconditioner Y_k^0 for x_k is called a C^M preconditioner, if

$$|\tilde{\boldsymbol{x}}_k^0 - \check{x}_k| = \min_{Y_k} |\tilde{\boldsymbol{x}}_k - \check{x}_k|.$$

Let $\tilde{\boldsymbol{x}}_i = \check{x}_k - \dfrac{\boldsymbol{n}_i(Y_i)}{\boldsymbol{d}_i(Y_i)}$. Then the C^M preconditioner Y_k^0 can be computed by solving the following optimization problem:

$$\min_{\underline{\boldsymbol{d}}_i(Y_i)=1} |\boldsymbol{n}_i(Y_i)|.$$

4. E^M preconditioner

Let $\tilde{\boldsymbol{x}}_i = \check{x}_k - \dfrac{\boldsymbol{n}_i(Y_i)}{\boldsymbol{d}_i(Y_i)}$. If $0 \in \boldsymbol{d}_i(Y_i)$ and $\boldsymbol{n}_i(Y_i) > 0$, then

$$\frac{\boldsymbol{n}_i(Y_i)}{\boldsymbol{d}_i(Y_i)} = \left(-\infty, \frac{\underline{\boldsymbol{n}}_i(Y_i)}{\underline{\boldsymbol{d}}_i(Y_i)}\right] \bigcup \left[\frac{\underline{\boldsymbol{n}}_i(Y_i)}{\overline{\boldsymbol{d}}_i(Y_i)}, \infty\right).$$

A preconditioner Y_k^0 for x_k is called an E^M preconditioner if it is a solution of the following optimization problem:

$$\max_{\underline{\boldsymbol{n}}_i(Y_i)=1} \min\left\{-\frac{1}{\underline{\boldsymbol{d}}_i(Y_i)}, \frac{1}{\overline{\boldsymbol{d}}_i(Y_i)}\right\} \tag{4}$$

However, $0 \in \boldsymbol{d}_i(Y_i)$ implies

$$\begin{aligned} \min\left\{-\frac{1}{\underline{\boldsymbol{d}}_i(Y_i)}, \frac{1}{\overline{\boldsymbol{d}}_i(Y_i)}\right\} &= \frac{1}{\max\{|\underline{\boldsymbol{d}}_i(Y_i)|, |\overline{\boldsymbol{d}}_i(Y_i)|\}} \\ &= \frac{1}{|\boldsymbol{d}_i(Y_i)|}, \end{aligned}$$

and $1/|\boldsymbol{d}_i(Y_i)|$ is maximum when $|\boldsymbol{d}_i(Y_i)|$ is minimum. Thus, Problem (4) can be replaced by

$$\min_{\underline{\boldsymbol{n}}_i(Y_i)=1} |\boldsymbol{d}_i(Y_i)|. \tag{5}$$

For C^W preconditioners, we have the following existence proposition. For the other preconditioners, similar results can be obtained.

Proposition 1 (Hu [1]) *There exists a C^W preconditioner Y_k if and only if at least one element of the k-th column of $\boldsymbol{A}$ does not contain 0.*

For each type of preconditioner discussed here, a nonlinear optimization problem is invoked in its definition. Fortunately, these nonlinear optimization problems can be simplified into linear programming problems. Thus, to obtain a preconditioner, we only need to solve a linear programming problem. Details can be found in [7].

3 Applications

C^W preconditioners minimize the width of $\tilde{\boldsymbol{x}}_k$, are appropriate for finding solutions of nonlinear systems. C^M preconditioners minimize the absolute value of $\tilde{\boldsymbol{x}}_k - \check{x}_k$, and are appropriate for verifying existence or uniqueness of solutions.

E^W and E^M preconditioners maximize the width and the absolute value of the gap that splits $\tilde{\boldsymbol{x}}_k - \check{x}_k$, respectively. They are appropriate for verifying that there is no solution in the initial box.

Example 2 *Consider:*

$$\begin{aligned} f_i(X) &= x_i + \sum_{1 \le j \le n} x_j - n - 1, \quad 1 \le i \le n-1, \\ f_n(X) &= \prod_{1 \le j \le n} x_j - 1, \end{aligned}$$

with $n = 5$ *and initial box* $\boldsymbol{X} = [0, 0.5] \times [0, 0.5] \times [0, 0.5] \times [0, 0.5] \times [0, 17]$.

The mean value extension, with a slope matrix, over the initial box is:

$$\boldsymbol{F}(\boldsymbol{X}) = \begin{bmatrix} [-6, 13.5] \\ [-6, 13.5] \\ [-6, 13.5] \\ [-6, 13.5] \\ [-1.996, 0.0625] \end{bmatrix}.$$

However, there is no root in the initial box. When the E^M preconditioner is used, this fact is discovered right after the first preconditioner is applied, but the C^W and C^M preconditioners will not show that there is no root.

In experiments to date, we have not found cases where existence could be verified with C^M preconditioners, but not C^W preconditioners. The reason is that the C^W preconditioners and the C^M preconditioners are the same if we choose the midpoint as the predictor point $\check{x}_k$. This is stated formally in the following theorem.

Theorem 1 *In (3), if we choose* $\check{X} = (\check{x}_1, \check{x}_2, \cdots, \check{x}_n)^T$ *to be the midpoint of* $\boldsymbol{X}$, *then the* C^W *preconditioners and the* C^M *preconditioners are the same.*

Proof: If $\check{X} = (\check{x}_1, \check{x}_2, \cdots, \check{x}_n)^T$ is chosen to be the midpoint of $\boldsymbol{X}$, then $\tilde{\boldsymbol{X}}$ will be symmetric about 0 (see [12]). Thus, $w(\tilde{\boldsymbol{x}}_k) = |\tilde{\boldsymbol{x}}_k - x_k|$. Therefore, the C^W and the C^M preconditioners are the same. □

References

[1] C. Hu. *Optimal Preconditioners for the Interval Newton Method.* PhD thesis, University of Southwestern Louisiana, 1990.

[2] R. B. Kearfott. Preconditioners for the interval Gauss–Seidel method. *SIAM J. Numer. Anal.*, 27(3):804–822, June 1990.

[3] R. B. Kearfott. Decomposition of arithmetic expressions to improve the behavior of interval iteration for nonlinear systems. *Computing*, 47(2):169–191, 1991.

[4] R. B. Kearfott. A Fortran 90 environment for research and prototyping of enclosure algorithms for nonlinear equations and global optimization. *ACM Trans. Math. Software*, 21(1):63–78, March 1995.

[5] R. B. Kearfott. Empirical evaluation of innovations in interval branch and bound algorithms for nonlinear algebraic systems, accepted for publication in the *SIAM J. Sci. Statist. Comput.*

[6] R. B. Kearfott. A review of techniques in the verified solution of constrained global optimization problems, to appear in *Applications of Interval Computations,* Kluwer Academic Publishers.

[7] R. B. Kearfott. *Verified Global Optimization – Interval Methods and Software.* Kluwer Academic Publishers, to appear.

[8] R. B. Kearfott, C. Hu, and M. Novoa. A review of preconditioners for the interval Gauss–Seidel method. *Interval Computations*, 1(1):59–85, 1991.

[9] R. B. Kearfott and M. Novoa. Algorithm 681: INTBIS, a portable interval Newton/bisection package. *ACM Trans. Math. Software*, 16(2):152–157, June 1990.

[10] R. B. Kearfott and X. Shi. A preconditioner selection heuristic for efficient iteration with decomposition of arithmetic expressions for nonlinear systems. *Interval Computations*, 1993(1):15–33, 1993.

[11] R. B. Kearfott and Z. Xing. An interval step control for continuation methods. *SIAM J. Numer. Anal.*, 31(3):892–914, June 1994.

[12] X. Shi. *Intermediate Expression Preconditioning and Verification for Rigorous Solution of Nonlinear Systems.* PhD thesis, University of Southwestern Louisiana, Department of Mathematics, August 1995.

Addresses:

R. B. Kearfott, The University of Southwestern Louisiana, Department of Mathematics, Lafayette, LA 70504, USA.

Xiaofa Shi, The University of Southwestern Louisiana, The Center for Advanced Computer Studies, Lafayette, LA 70504, USA.