Systems of Neo Linear Equations

by
Namir Shammas

Introduction

Linear equations form one of the cornerstones of mathematics. In addition to representing simple equations, many problems in math, statistics, and science are solved as linear equations. This article looks at a new type of non-linear equations that closely resemble their linear cousins. The difference is that the variables in the diagonal element are raised to the same positive integer. I will call this type of non-linear equations the *System of Neo Linear Equations* (SNLE) and the exponents of the diagonal elements as the *order* of the SNLE. I have not been able to find real world applications for SNLE. So for now, the concept of solving SNLE remains a purely mathematical exercise.

Here is an example of a second order SNLE that has three sets of equations:

$$a_{11}\ x_1^2 + a_{12}\ x_2 + a_{13}\ x_3 = b_1$$
$$a_{21}\ x_1 + a_{22}\ x_2^2 + a_{23}\ x_3 = b_2$$
$$a_{31}\ x_1 + a_{32}\ x_2 + a_{33}\ x_3^2 = b_3$$

The above example is, in the mind of the math purists, just a set of nonlinear equations that can be solved using one of several iterative methods available for systems of non-linear equations. The above example also shows that the order of the SNLE is independent of the number of equations.

This article aims at using iterative methods for (large) systems of linear equations to solve the SNLE with real coefficients and real variables. In particular, I will use the Gauss-Seidel algorithm. This method is used typically with systems of linear equations **Ax**=**b.** The algorithm converges when **A** is either a diagonal dominant matrix or is symmetric positive-definite. The Gauss-Seidel algorithm has been used in solving non-liner equations in general, although Newton's method is preferred since it it more reliable. As you will see, applying the Gauss-Seidel algorithm to SNLE yields interesting and encouraging results.

The SNLE for order 2 in matrix form is:

$\mathbf{A}\,\mathbf{x} - \mathbf{I}\,\mathbf{x} + \mathbf{x}^T\,\mathrm{I}\,\mathbf{x} = \mathbf{b}$

Heads Up

Let's first look at the Gauss-Seidel algorithm as applied to linear equations. Here is the pseudo-code for the algorithm:

```
Give N linear equations Ax=b with an initial guess
vector X0, MaxIter, and MinNorm.

X=X0
Iter=0
Do
  For I = 1 to N
    sum = -b(I)
    x(I) = 0
    For J = 1 to N
      sum = sum + A(I,J) * x(J)
    Next J
    x(I) = sum / A(I,I)
  Next I
  NormVal = Norm(A*x-b)
  Iter = Iter + 1
Loop until NormVal <= MinNorm or Iter > MaxIter
```

Here is the version of the Gauss-Seidel algorithm that I will use with the SNLE:

```
Give N Neo linear equations of order P and with an
initial guess vector X0, MaxIter, and MinNorm.

X=X0
Iter=0
Do
  For I = 1 to N
    sum = -b(I)
    x(I) = 0
    For J = 1 to N
      sum = sum + A(I,J) * x(J)
    Next J
    x(I) = (sum / A(I,I))^(1/P)
  Next I
```

```
  NormVal = Norm(A*x-b)
  Iter = Iter + 1
Loop until NormVal <= MinNorm or Iter > MaxIter
```

Assigning exponents to the diagonal elements introduces new computational elements that we need to take in consideration. In the case of even exponents, the updated values for the guesses need to be calculated as the even roots (i.e. square root, fourth root, and so on) of positive values in order to avoid complex results. This condition dictates the types of values in matrix **A** that are needed to work with the Gauss-Seidel algorithm. By contrast, when the exponents are odd numbers, the updated values for the guesses can be calculated as odd roots (i.e. cube root, fifth root, and so on) of positive OR negative numbers. The lack of restriction should remove the limitations on the values of matrix **A**. Thus SNLE with odd orders should better lend themselves to work with the Gauss-Seidel algorithm.

Working with SNLE of Order 2

Table 1 shows the summary of the calculations done using Excel VBA functions and subroutines for SNLE order 2. The results show that the Gauss-Seidel algorithms works well with order 2 of SNLE when:

- The values of the solution vector **x** are positive.
- The values of the elements of coefficients matrix **A** are positive.
- The coefficients matrix **A** are diagonal dominant.

Table 1. The results of SNLE of order 2.

Number of Equations	*X(i)*	*Non-Diag Elements*	*Diag Elements*	*Results*
5	X(i)=i	Random and has mixed signs	Diagonal Dominant	Occasional error at first iteration.
5	X(i)=i	A(i,j) random and > 0	Diagonal Dominant	No errors
5	X(i)=i	Random and has mixed signs	A(i,i) = Max(|x(row i)|) + N	Persistent runtime errors

Number of Equations	*X(i)*	*Non-Diag Elements*	*Diag Elements*	*Results*
5	X(i) = +/- i with the alternating signs	A(i,j) random and > 0	Diagonal Dominant	No valid solutions: either runtime error or wrong answer
25	X(i)=i	Random and has mixed signs	Diagonal Dominant	Frequent error at first iteration.
25	X(i)=i	A(i,j) random and > 0	Diagonal Dominant	No errors
25	X(i)=i	Random and has mixed signs	A(i,i) = Max(|x(row i)|) + N	Persistent runtime errors
25	X(i) = +/- I with the alternating signs	A(i,j) random and > 0	Diagonal Dominant	Persistent runtime errors

Working with SNLE of Order 3

Table 2 shows the summary of the calculations done using Excel VBA functions and subroutines for SNLE order 3. The result for order 3 are far more encouraging than for order 2. The Gauss-Seidel algorithm is able to solve SNLE of order 3 without any reasonable restrictions on the values of the solution vector **x** and coefficients matrix **A**.

Table 2. The results of SNLE of order 3.

Number of Equations	*X(i)*	*Non-Diag Elements*	*Diag Elements*	*Results*
5	X(i)=i	Random and has mixed signs	Random and has mixed signs	No errors

Number of Equations	*X(i)*	*Non-Diag Elements*	*Diag Elements*	*Results*
5	X(i) is random and has mixed signs	Random and has mixed signs	Random and has mixed signs	No errors
25	X(i)=i	Random and has mixed signs	Random and has mixed signs	No errors
25	X(i) is random and has mixed signs	Random and has mixed signs	Random and has mixed signs	No errors

Working with SNLE of Order 4

Table 3 shows the summary of the calculations done using Excel VBA functions and subroutines for SNLE order 4. The results show that the Gauss-Seidel algorithms works well with order 4 of SNLE when:

- The values of the solution vector **x** are positive.
- The values of the elements of coefficients matrix **A** are positive.
- The coefficients matrix **A** are diagonal dominant.

Table 3. The results of SNLE of order 4.

Number of Equations	*X(i)*	*Non-Diag Elements*	*Diag Elements*	*Results*
5	X(i)=i	Random and has mixed signs	Diagonal Dominant	Runtime error appeared with low frequency.
5	X(i)=i	A(i,j) random and > 0	Diagonal Dominant	No errors
5	X(i)=i	Random and mixed signs	A(i,i) = Max(|x(row i)|) + N	Persistent runtime errors

Number of Equations	*X(i)*	*Non-Diag Elements*	*Diag Elements*	*Results*
5	X(i) = +/- i with the alternating signs	A(i,j) random and > 0	Diagonal Dominant	No valid solutions: either runtime error or wrong answer
25	X(i)=i	Random and has mixed signs	Diagonal Dominant	Frequent error at first iteration.
25	X(i)=i	A(i,j) random and > 0	Diagonal Dominant	No errors
25	X(i)=i	Random and has mixed signs	A(i,i) = Max(|x(row i)|) + N	Frequent error at first iteration.
25	X(i) = +/- i with the alternating sign	A(i,j) random and > 0	Diagonal Dominant	Persistent runtime errors

Working with SNLE of Order 5

Table 4 shows the summary of the calculations done using Excel VBA functions and subroutines for SNLE order 5. The result for order 5 are very similar to those for order 3. The Gauss-Seidel algorithm is able to solve SNLE of order 5 without any reasonable restrictions on the values of the solution vector **x** and coefficients matrix **A**.

Table 4. The results of SNLE of order 5.

Number of Equations	*X(i)*	*Non-Diag Elements*	*Diag Elements*	*Results*
5	X(i)=i	Random and has mixed signs	Random and has mixed signs	No errors

Number of Equations	***X(i)***	***Non-Diag Elements***	***Diag Elements***	***Results***
5	X(i) is random and has mixed signs	Random and has mixed signs	Random and has mixed signs	No errors
25	X(i)=i	Random and has mixed signs	Random and has mixed signs	No errors
25	X(i) is random and has mixed signs	Random and has mixed signs	Random and has mixed signs	No errors

Conclusions

The conclusion to draw from using the Gauss-Seidel algorithm with SNLE are:

1. The algorithm can solve odd orders of the SNLE without reasonable restrictions on the values of the solution vector **x** and coefficients matrix **A.** The requirement of diagonally-dominant matrix elements does not apply.
2. The algorithm can solve even orders of SNLE as long as the following conditions are observed:
    a. The values of the solution vector **x** are positive.
    b. The values of the elements of coefficients matrix **A** are positive.
    c. The coefficients matrix **A** are diagonal dominant

Epilogue: SNLE of the Second Kind

Further investigations have shown that the Gauss-Seidel algorithm works for SNLEs that have terms with power greater than 1 in non-diagonal terms. I will call this kind of equations as the *System of Neo-Linear Equations of the Second Kind*. They are defined with two order values, n and m—n is the order of the diagonal elements, and m is the order for non-diagonal elements.

Here is an example of a second order SNLE that has three sets of equations:

$$a_{11}\ x_1^5 + a_{12}\ x_2^2 + a_{13}\ x_3^2 = b_1$$
$$a_{21}\ x_1^2 + a_{22}\ x_2^5 + a_{23}\ x_3^2 = b_2$$

$$a_{31}\ x_1^2 + a_{32}\ x_2^2 + a_{33}\ x_3^5 = b_3$$

I will call the above system of equation as SNLE of the second kind with order (5, 2). The conclusions drawn earlier seem to apply for SNLE of the second kind with odd values of n (as long as they are greater than the values of m). The values for m can be odd or even. I have tested SNLEs of the second kind for 5 and 25 equations with different values for n and m. These tests showed for odd values or n and for n > m, the Gauss-Seidel method solved SNLE of the second kind in a few iterations.