*American Journal of Applied Mathematics and Statistics, 2016, Vol. 4, No. 4, 113-117*
Available online at http://pubs.sciepub.com/ajams/4/4/3

DOI:10.12691/ajams-4-4-3

# Applying the Successive Over-relaxation Method to a Real World Problems

**T. Mayooran*, Elliott Light**

Department of Mathematics and Statistics, Minnesota state university, Mankato, USA
*Corresponding author: thevaraja.mayooran@mnsu.edu

**Abstract** Solving a system of equations by $Ax = b$, where A is a $n \times n$ matrix and b and $n \times 1$ vector, can sometime be a daunting task because solving for *x* can be difficult. If you were given an algorithm that was efficient, that's great! What if you could make it solve the problem even faster? That's even better. We will first take a look at establishing the basics of the successive over-relaxation method (SOR for short), then we'll look at a real-world problem we applied the SOR method to, solving the heat-equation when a constant boundary temperature is applied to a flat plate.

***Keywords:*** *Interactive Method, Successive Over-Relaxation Method (SOR)*

**Cite This Article:** T. Mayooran, and Elliott Light, "Applying the Successive Over-relaxation Method to a Real World Problems." *American Journal of Applied Mathematics and Statistics*, vol. 4, no. 4 (2016): 113-117. doi: 10.12691/ajams-4-4-3.

## 1. Introduction

Successive over-relaxation (SOR) is one of the most important method for solution of large linear system equations. It has applications in Fluid Dynamics, mathematical programming, linear elasticity and machine learning etc. The examples of applications of SOR in Dynamics include study of steady heat conduction, turbulent flows, boundary layer flows or chemically reacting flows. For this reason, SOR method is important for both researchers and business policymakers.

In the real world, time is always something valuable, something no one wants to waste; when it comes to solving systems of equations, it can sometimes be better to get a close approximation of the solution than to get the exact solution for this very reason, among others. This is where the successive over-relaxation method (SOR) can come into play. The industry standard for finding exact methods, Gaussian elimination, requires approximately $\frac{n^3}{3}$ operations to solve the system, which becomes time consuming when n gets big. SOR on the other hand, while only giving us an approximation, can give us these approximations much faster than Gaussian elimination can. SOR was developed in 1950 by David Young and H. Frankel in 1950 and was developed to be used on digital computers. It was developed by modifying the Gauss-Seidel iteration model. The Gauss-Seidel model is based on the following steps.

1. Given Ax = b. where A and b are known and an initial guess for x, $x_0$
2. $L_* x_{k+1} = b - U x_k$

Where $L_*$ is the lower triangular components of matrix A, U is the upper triangular components of A, b is our b vector and $x_k$ is the $k^{th}$ approximation of x and $x_{k+1}$ is the next iteration of x.For the numerical solution of the accelerated Overrelaxation method was introduced by Hadjidimos in [1] and is a two-parameter generalization of the successive Overrelaxation (SOR) method.The SOR method works this way.

1. Given Ax = b where A and b are known, x unknown, and an initial guess for x, $x_0$
2. Let $A = D + L + U$ where $D$ is the main diagonal of A, L the lower triangle components of A and U the upper triangle components of A.
3. $x_{k+1} = (1-\omega)x_i + \frac{\omega}{a_{i,i}}(b_i - \sum_{j<i} a_{i,j} x_{j(k+1)} - \sum_{j>i} a_{ij} x_{j\,k}$.

Where $x_k$ is the kth approximation of $x$, $x_{k+1}$is the next iteration of $x$, $a_{i,j}$ is the corresponding element of matrix A, b is our vector and $\omega$ is our relaxation factor. We'll talk more about selecting an appropriate relaxation factor when we get to the next section, but for now, note that if $\omega = 1$, we get the Gauss-Siedel method. The convergence is enhanced because the value at a particular iteration is made up of a combination of the old value and the newly calculated value, namely

$$x_i^{new} = \omega x_i^{new} + (1-\omega) x_i^{old}.$$

The SOR method is very similar to the Gauss-Seidel method except that it uses a scaling factor to reduce the approximation error. Consider the following set of equations

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, i = 1, 2, .....n.$$

For Gauss-Seidel method, the values at the *k* iteration are given by