Sudan University of Sciences and Technology
college of Graduate Studies

The Singular Value Decomposition (SVD) For Solving a System Of Algebraic Equations

A thesis Submitted in Partial Fullfillment for the degree of M.Sc in Mathematics

by:

Omnia Hussien AbdAlrhman Taha

Supervisor:

Dr. Mohamed Hassan Mohamed Khabir

May 2016

ABSTRACT

In thesis we present an iterative method to solve a system of equations approximately. Firstly we use two iterative methods for the solutions of a system of algebraic equations namely Gauss Jacobi iteration method and Gauss-Siedel iteration method. Most the iterative methods may converge or not. However certain class of systems of simultaneous equations which is diagonally dominant do always converge to a solution using Gauss-Siedel method. It is possible that a system of equation might be diagonally dominant if we exchanges the equations with each other.

Also, we presented the singular value decomposition(SVD). It has been used to determine the properties of matrix, matrix norm and rank. Since the inverse of a matrix is often difficult to compute accurately,the SVD is used to compute the matrix inverse and then solving a linear systems of equations.

Also, we use the SVD method to solve one of the least squares problems which is overdetermined problem. We use the MATLAB software for the solution.

ABSTRACT

In thesis we present an iterative method to solve a system of equations approximately. Firstly we use two iterative methods for the solutions of a system of algebraic equations namely Gauss Jacobi iteration method and Gauss-Siedel iteration method. Most the iterative methods may converge or not. However certain class of systems of simultaneous equations which is diagonally dominant do always converge to a solution using Gauss-Siedel method. It is possible that a system of equation might be diagonally dominant if we exchanges the equations with each other.

Also, we presented the singular value decomposition(SVD). It has been used to determine the properties of matrix, matrix norm and rank. Since the inverse of a matrix is often difficult to compute accurately,the SVD is used to compute the matrix inverse and then solving a linear systems of equations.

Also, we use the SVD method to solve one of the least squares problems which is overdetermined problem. We use the MATLAB software for the solution.

ACKNOWLEDGEMENT

I am truly indebted and thankful to my supervisor **Dr. Mohamed Hassan Mohamed Khabir** for the patient guidance, encouragement and advice he has provided throughout my time under his supervision. I have been extremely lucky to have a supervisor who responded to my questions and quires patiently and promptly.

Also I would like to express my gratitude to **Mr.Omran Salih**, for his kind assistant.

DEDICATION

This research is dedication:-

To my parents

Mrs. Maria Mohamed and Mr. Hussien Abdalrahman

To my sister

Mrs. Eglal Hussien

To all my family member

To my friends

To someone who has alot in my Deep down...

Contents

List of Figures

Chapter 1

Iterative Methods

1.1 Introduction

Iterative methods are based on the idea of successive approximations. We start with an initial approximation to the solution vector $x = x_0$, to solve the system of equations $Ax = b$, and obtain a sequence of approximate vectors $x_0, x_1, ..., x_k, ...$, which in the limit as k $\to \infty$, converges to the exact solution vector $x = A^{-1}b$. A general linear iterative method for the solution of the system of equations $Ax = b$, can be written in matrix form as

$$x^{(k+1)} = Hx^{(k)} + c, \quad k = 0, 1, 2, ..., \tag{1.1}$$

where $x^{(k+1)}$ and $x^{(k)}$ are the approximations for **x** at the $(k+1)th$ and kth iterations respectively. H is called the iteration matrix, which depends on A and **c** is a column vector, which depends on A and b. We stop the iteration procedure when the magnitudes of the differences between the two successive iterates of all the variables are smaller than a given accuracy or error tolerance or an error bound ϵ, that is,

$$|x_i^{(k+1)} - x_i^{(k)}| \leq \epsilon, \text{for all } i. \tag{1.2}$$

For example, if we require two decimal places of accuracy, then we iterate until $|x_i^{(k+1)} - x_i^{(k)}| \leq 0.005$, for all i. If we require three decimal places of accuracy, then we iterate until $|x_i^{(k+1)} - x_i^{(k)}| \leq 0.0005$, for all i. Convergence property of an iterative method depends on the iteration matrix H.

Now,we derive two iterative methods for the solution of the system of algebraic equations.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3. \end{aligned} \tag{1.3}$$

1.2 Gauss-Jacobi Iteration Method

We assume that the pivots $a_{ii} \neq 0$, for all i and we write the equations as

$$\begin{aligned} a_{11}x_1 &= b_1 - (a_{12}x_2 + a_{13}x_3), \\ a_{22}x_2 &= b_2 - (a_{21}x_1 + a_{23}x_3), \\ a_{33}x_3 &= b_3 - (a_{31}x_1 + a_{32}x_2). \end{aligned} \tag{1.4}$$

The Gauss Jacobi iteration method (also called Jacobi Method) is defined as

$$\begin{aligned} x_1^{(k+1)} &= \tfrac{1}{a_{11}}[b_1 - (a_{12}x_2^{(k)} + a_{13}x_3^{(k)})], \\ x_2^{(k+1)} &= \tfrac{1}{a_{22}}[b_2 - (a_{21}x_1^{(k)} + a_{23}x_3^{(k)})], \\ x_3^{(k+1)} &= \tfrac{1}{a_{33}}[b_3 - (a_{31}x_1^{(k)} + a_{32}x_2^{(k)})]. \end{aligned} \tag{1.5}$$

Since, we replace the complete vector $x^{(k)}$ in the right hand side of (1.2) at the end of each iteration, this method is also called the method of simultaneous displacement.

A sufficient condition for convergence of the Jacobi method is that the system of equations is diagonally dominant, that is, the coefficient matrix A is diagonally dominant.We can verify that $|a_{ii}| \geq \sum_{j=1,i\neq j}^{n} |a_{ij}|$. This implies that convergence may be obtained even if the system is not diagonally dominant. If the system is not diagonally dominant, we may exchange the equations, if possible, such that the new system is diagonally dominant and convergence is guaranteed [6].

Remark 1.2.1 *How do we find the initial approximations to start the iteration ? If the system is diagonally dominant, then the iteration converges for any initial solution vector. If no suitable approximation is available, we can choose $x = 0$, that is $x_i = 0$ for all i. Then, the initial approximation becomes $x_i = b_1/a_{ii}$, for all i. [6]*

Example 1.2.1 *let us consider the following system of equations*

$$\begin{aligned} 20x_1 + x_2 - 2x_3 &= 17, \\ 3x_1 + 20x_2 - x_3 &= -18, \\ 2x_1 - 3x_2 + 20x_3 &= 25. \end{aligned} \tag{1.6}$$

to apply the Jacobi iteration method,we start for the initial approximations as

$$x_i = 0, \quad i = 1, 2, 3.$$

Jacobi method gives the iterations as

$$x_1^{(k+1)} = 0.05[17 - (x_2^{(k)} - 2x_3^{(k)})],$$

$$x_2^{k+1} = 0.05[-18 - (3x_1^{(k)} - x_3^{(k)})],$$

$$x_3^{(k+1)} = 0.05[25 - (2x_1^{(k)} - 3x_2^{(k)})], k = 0, 1, ...$$

We have the following results(we perform five iteration in each case).
$x_1^{(0)} = 0, x_2^{(0)} = 0, x_3^{(0)} = 0.$
First iteration
$x_1^{(1)} = 0.5[17 - (x_2^{(0)} - 2x_3^{(0)})] = 8.5,$
$x_2^{(1)} = 0.5[-18 - (3x_1^{(0)} - x_3^{(0)})] = -0.9,$
$x_3^{(1)} = 0.5[25 - (2x_1^{(0)} - 3x_2^{(0)})] = 1.25.$
Second iteration
$x_1^{(2)} = 0.05[17 - (x_2^{(1)} - 2x_3^{(1)})] = 0.05[17 - (0.9 - 2(1.25))] = 1.02,$
$x^{(2)} = 0.05[-18 - (3x_1^{(1)} + x_3^{(1)})] = 0.05[-18 - (3(0.85) - 1.25)] = -0.965,$
$x_3^{(2)} = 0.05[25 - (2x_1^{(1)} + 3x_2^{(1)})] = 0.05[25 - (2(0.85) - 3(- 0.9))] = 1.03.$
Third iteration
$x_1^{(3)} = 0.05[17 - (x_2^{(2)} + 2x_3^{(2)})] = 0.05[17 - (-0.965 - (1.03))] = 1.00125,$
$x_2^{(3)} = 0.05[-18 - (3x_1^{(2)} + x_3^{(2)})] = 0.5[- 18 - (-3(1.02) - 1.03)] = -1.0015,$
$x_3^{(3)} = 0.05[25 - (2x_1^{(2)} + 3x_2^{(2)})] = 0.5[25 - (2(1.00125) - 3(- 1.0015))] = 1.00325.$
Fourth iteration
$x_1^{(4)} = 0.05[17 - (x_2^{(3)} + 2x_3^{(3)})] = 0.05[17 - (-1.0015 - 2(1.00325)) = 1.0004,$

$x_2^{(4)} = 0.05[-18-(3x_1^{(3)} + x_3^{(3)})] = 0.05[-18-(3(1.00125)-1.000325)] = -1.000025,$
$x_3^{(4)} = 0.05[25-(2x_1^{(3)} + 3x_2^{(3)})] = 0.05[25-(2(1.0004)-3(-1.000025))] = 0.99965.$

Fifth iteration

$x_1^{(5)} = 0.05[17-(x_2^{(4)} + 2x_3^{(4)})] = 0.05[17-(-1.000025-2(0.99965))] = 0.99996625,$
$x_2^{(5)} = 0.05[-18-(3x_1^{(4)} + x_3^{(4)})] = 0.05[-18-(3(1.0004)-0.99965)] = -1.0000775,$
$x_3^{(5)} = 0.05[25-(2x_1^{(4)} + 3x_2^{(4)})] = 0.05[25-(2(1.0004)-3(-1.000025))] = 0.99995625.$

Since, all the errors in magnitude are less than 0.0005, the required solution is $x_1 = 1, x_2 = -1, x_3 = 1$

Example 1.2.2 *let us consider the following system of equations*

$$\begin{aligned} 26x_1 + 2x_2 + 2x_3 &= 12.6, \\ 3x_1 + 27x_2 + x_3 &= -14.3, \\ 2x_1 + 3x_2 + 17x_3 &= 6.0. \end{aligned} \tag{1.7}$$

to apply the Jacobi iteration method and we obtain the result correct to three decimal places.

Solution

The given system of equations is strongly diagonally dominant. Hence, we can expect faster convergence. Jacobi method gives the iterations as

$$x_1^{(k+1)} = [12.6-(2x_2^{(k)} + 2x_3^{(k)})]/26,$$

$$x_2^{(k+1)} = [-14.3-(3x_1^{(k)} + x_3^{(k)})]/27,$$

$$x_3^{(k+1)} = [6.0-(2x_1^{(k)} + 3x_2^{(k)})]/17 \quad k = 0, 1, ...$$

We choose the initial approximation as $x_1^{(0)} = 0, x_2^{(0)} = 0, x_3^{(0)} = 0$. We obtain the following results.

First iteration

$x_1^{(1)} = 1/26[12.6-(2x_2^{(0)} + 2x_3^{(0)})] = 1/26[12.6] = 0.48462,$
$x_2^{(1)} = 1/27[-14.3-(3x_1^{(0)} + x_3^{(0)})] = 1/27[-14.3] = -0.52963,$
$x_3^{(1)} = 1/17[6.0-(2x_1^{(0)} + 3x_2^{(0)})] = 1/17[6.0] = 0.35294.$

Second iteration

$x_1^{(2)} = 1/26[12.6 - (2x_2^{(1)} + 2x_3^{(1)})] = 1/26[12.6 - 2(- 0.52963 + 0.35294)] = 0.49821,$

$x_2^{(2)} = 1/27[-14.3 - (3x_1^{(1)} + x_3^{(1)})] = 1/27[-14.3 - (3(0.48462) + 0.35294)] = -0.59655,$

$x_3^{(2)} = 1/17[6.0 - (2x_1^{(1)} + 3x_2^{(1)})] = 1/17[6.0 - (2(0.48462) + 3(-0.52963))] = 0.38939.$

Third iteration

$x_1^{(3)} = 1/26[12.6 - (2x_2^{(2)} + 2x_3^{(2)})] = 1/26[12.6 - 2(-0.59655 + 0.38939)] = 0.50006,$

$x_2^{(3)} = 1/27[-14.3 - (3x_1^{(2)} + x_3^{(2)})] = 1/27[-14.3 - (3(0.49821) + 0.38939)] = -0.59941,$

$x_3^{(3)} = 1/17[6.0 - (2x_1^{(2)} + 3x_2^{(2)})] = 1/17[6.0 - (2(0.49821) + 3(0.59655))] = 0.39960.$

Fourth iteration

$x_1^{(4)} = 1/26[12.6 - (2x_2^{(3)} + 2x_3^{(3)})] = 1/26[12.6 - 2(- 0.59941 + 0.39960)] = 0.50000,$

$x_2^{(4)} = 1/27[-14.3 - (3x_1^{(3)} + x_3^{(3)})] = 1/27[-14.3 - (3(0.50006) + 0.39960)] = -0.59999,$

$x_3^{(4)} = 1/17[6.0 - (2x_1^{(3)} + 3x_2^{(3)})] = 1/17[6.0 - (2(0.50006) + 3(- 0.59941))] = 0.39989.$

We find $|x_1^{(4)} - x_1^{(3)}| = |0.5 - 0.50006| = 0.00006,$

$|x_2^{(4)} - x_2^{(3)}| = | - 0.59999 + 0.59941| = 0.00058,$

$|x_3^{(4)} - x_3^{(3)}| = |0.39989 - 0.39960| = 0.00029.$

Three decimal places of accuracy have not been obtained at this iteration.

Fifth iteration

$x_1^{(5)} = 1/26[12.6 - (2x_2^{(4)} + 2x_3^{(4)})] = 1/26[12.6 - 2(- 0.59999 + 0.39989)] = 0.50001,$

$x_2^{(5)} = 1/27[-14.3 - (3x_1^{(4)} + x_3^{(4)})] = 1/27[-14.3 - (3(0.50000) + 0.39989)] = -0.60000,$

$x_3^{(5)} = 1/17[6.0 - (2x_1^{(4)} + 3x_2^{(4)})] = 1/17[6.0 - (2(0.50000) + 3(0.59999))] = 0.40000.$

We find $|x_1^{(4)} - x_1^{(3)}| = |0.50001 - 0.5| = 0.00001,$

$|x_2^{(4)} - x_2^{(3)}| = | - 0.6 + 0.59999| = 0.00001,$

$|x_3^{(4)} - x_3^{(3)}| = |0.4 - 0.39989| = 0.00011.$

Since, all the errors in magnitude are less than 0.0005, the required solution is $x_1 = 0.5, x_2 = 0.6, x_3 = 0.4.$

The disadvantage of the Gauss-Jacobi method is that at any iteration step, the value of the first variable x_1 is obtained using the values of the previous iteration. The value of the second variable x_2 is also obtained using the values of the previous iteration, even though the updated value of x_1 is available. In general, at every stage in the iteration, values of the previous iteration are used even though the updated values of the previous variables are available. If we use the updated values of $x_1, x_2, ..., x_{i-1}$ in computing the value of the variable x_i, then we obtain a new method called Gauss-Seidel iteration method.

1.3 Gauss-Siedel Iteration Method

In certain cases, such as when a system of equations is large, iterative methods of solving equations such as Gauss-Siedel method are more advantageous. Iterative methods, such as Gauss-Siedel method, allow the user the control of the roundoff error. Also if the physics of the problem are wellknown for faster convergence, initial guesses needed in iterative methods can be made more judiciously.More information on the use of iteration methods for solving linear systems can be found in[1,5,6]. Given a general set of n equations and n unknowns, we have

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= c_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= c_2, \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n &= c_n \end{aligned}$$

If the diagonal elements are non-zero, each equation is rewritten for the corresponding unknown.The first equation is rewritten with x_1 on the left hand side, second equation is rewritten with x_2 on the left hand side and so on as follows

$$\begin{aligned} x_1 &= \frac{c_1 - a_{12}x_2 - a_{13}x_3 \cdots - a_{1n}x_n}{a_{11}}, \\ x_2 &= \frac{c_2 - a_{21}x_1 - a_{23}x_3 \cdots - a_{2n}x_n}{a_{22}}, \\ \vdots &\quad \vdots \\ \vdots &\quad \vdots \\ x_{n-1} &= \frac{c_{n-1} - a_{n-1,1}x_1 - a_{n-1,2}x_2 \cdots - a_{n-1,n-2}x_{n-2} - a_{n-1,n}x_n}{a_{n-1,n-1}}, \\ x_n &= \frac{c_n - a_{n1}x_1 - a_{n2}x_2 \cdots - a_{n,n-1}x_{n-1}}{a_{nn}}. \end{aligned}$$

These equations can be rewritten in the summation form as

$$
\begin{aligned}
x_1 &= \frac{c_1 - \sum_{j=1, j\neq 1}^{n} a_{1j}x_j}{a_{11}},\\
x_2 &= \frac{c_2 - \sum_{j=1, j\neq 2}^{n} a_{2j}x_j}{a_{22}},\\
\vdots &\quad \vdots\\
\vdots &\quad \vdots\\
x_{n-1} &= \frac{c_{n-1} - \sum_{j=1, j\neq -1}^{n} a_{n-1,j}x_j}{a_{n-1,n-1}},\\
x_n &= \frac{c_n - \sum_{j=1, j\neq n}^{n} a_{n,j}x_j}{a_{nn}}.
\end{aligned}
$$

Hence for any row $'i'$,

$$x_i = \frac{c_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j}{a_{ii}}, \quad i = 1, 2, \ldots, n.$$

Now to find x_i's, we assumes an initial guess for the x_i's and then use the rewritten equations to calculate the new guesses. At the end of each iteration, we calculates the absolute relative approximate error for each x_i as

$$|\varepsilon_a|_i = \left| \frac{x_i^{new} - x_i^{old}}{x_i^{new}} \right| \times 100,$$

where x_i new is the recently obtained value of x_i, and x_i old is the previous value of x_i. When the absolute relative approximate error for each x_i is less than the prespecified tolerance, the iterations are stopped.

Example 1.3.1 *Find the solution of the system of equations*

$$
\begin{aligned}
45x_1 + 2x_2 + 3x_3 &= 58,\\
3x_1 + 22x_2 + 2x_3 &= 47, \qquad (1.8)\\
5x_1 + x_2 + 20x_3 &= 67.
\end{aligned}
$$

correct to three decimal places, using the Gauss-Seidel iteration method.

Solution

The given system of equations is strongly diagonally dominant. Hence, we can expect fast convergence. Gauss-Seidel method gives the iteration

$$x_1^{(k+1)} = 1/45(58 - 2x_2^{(k)} - 3x_3^{(k)}),$$

$$x_2^{(k+1)} = 1/22(47 + 3x_1^{(k+1)} - x_3^{(k)}),$$

$$x_3^{(k+1)} = 1/20(67 - 5x_1^{(k+1)} - x_2^{(k+1)}).$$

Starting with $x_1^{(0)} = 0, x_2^{(0)} = 0, x_3^{(0)} = 0$, we get the following results.

First iteration

$$x_1^{(1)} = 1/45(58 - 2x_2^{(0)} - 3x_3^{(0)}) = 1/45(58) = 1.28889,$$

$$x_2^{(1)} = 1/22(47 + 3x_1^{(1)} - x_3^{(0)}) = 1/22(47 + 3(1.28889) - 2(0)) = 2.31212,$$

$$x_3^{(1)} = 1/20(67 - 5x_1^{(1)} - x_2^{(1)}) = 1/20(67 - 5(1.28889) - (2.31212)) = 2.91217.$$

Second iteration

$$x_1^{(2)} = 1/45(58 - 2x_2^{(1)} - 3x_3^{(1)}) = 1/45(58 - 2(2.31212) - 3(2.91217)) = 0.99198,$$

$$x_2^{(2)} = 1/22(47 + 3x_1^{(2)} - x_3^{(1)}) = 1/22(47 + 3(0.99198)2(2.91217)) = 2.00689,$$

$$x_3^{(2)} = 1/20(67 - 5x_1^{(2)} - x_2^{(2)}) = 1/20(67 - 5(0.99198) - (2.00689)) = 3.00166.$$

Third iteration

$$x_1^{(3)} = 1/45(58 - 2x_2^{(2)} - 3x_3^{(2)}) = 1/45(58 - 2(2.00689) - 3(3.00166) = 0.99958,$$

$$x_2^{(3)} = 1/22(47 + 3x_1^{(3)} - x_3^{(2)}) = 1/22(47 + 3(0.99958) - 2(3.00166)) = 1.99979,$$

$$x_3^{(3)} = 1/20(67 - 5x_1^{(3)} - x_2^{(3)}) = 1/20(67 - 5(0.99958) - (1.99979)) = 3.00012.$$

Fourth iteration

$$x_1^{(4)} = 1/45(58 - 2x_2^{(3)} - 3x_3^{(3)}) = 1/45(58 - 2(1.99979) - 3(3.0001)) = 1.00000,$$

$$x_2^{(4)} = 1/22(47 + 3x_1^{(4)} - x_3^{(3)}) = 1/22(47 + 3(1.00000) - (3.00012) = 1.99999,$$

$$x_3^{(4)} = 1/20(67 - 5x_1^{(4)} - x_2^{(4)}) = 1/20(67 - 5(1.00000) - (1.99999)) = 3.00000.$$

We find $|x_1^{(4)} + x_1^{(3)}| = |1.00000 - 0.99958| = 0.00042$,
$|x_2^{(4)} + x_2^{(3)}| = |1.99999 - 1.99979| = 0.00020$,
$|x_3^{(4)} + x_3^{(3)}| = |3.00000 - 3.00012| = 0.00012$.
Since, all the errors in magnitude are less than 0.0005, the required solution is $x_1 = 1.0, x_2 = 1.99999, x_3 = 3.0$. Rounding to three decimal places, we get $x_1 = 1.0, x_2 = 2.0, x_3 = 3.0$.

Example 1.3.2 *We computationally show that Gauss-Seidel method applied to the following system of equations*

$$\begin{aligned} 3x_1 - 6x_2 + 2x_3 &= 23, \\ -4x_1 + x_2 - x_3 &= -8, \\ x_1 - 3x_2 + 7x_3 &= 17, \end{aligned} \tag{1.9}$$

diverges. We take the initial approximations as $x_1 = 0.9, x_2 = 3.1, x_3 = 0.9$. Interchange the first and second equations and solve the resulting system by the Gauss-Seidel method. Again take the initial approximations as $x_1 = 0.9, x_2 = 3.1, x_3 = 0.9$, and obtain the result correct to two decimal places. The exact solution is $x_1 = 1.0, x_2 = 3.0$, $x_3 = 1.0$.

Solution

Note that the system of equations is not diagonally dominant. Gauss-Seidel method gives the iteration

$$\begin{aligned} x_1^{(k+1)} &= [23 + 6x_2^{(k)} - 2x_3^{(k)})]/3, \\ x_2^{(k+1)} &= [-8 + 4x_1^{(k+1)} + x_3^{(k)}], \\ x_3^{(k+1)} &= [17 - x_1^{(k+1)} + 3x_2^{(k+1)}]/7. \end{aligned}$$

Starting with the initial approximations $x_1 = 0.9, x_2 = 3.1, x_3 = 0.9$, we obtain the following results.

First iteration

$$x_1^{(1)} = \frac{1}{3}[23 + 6x_2^{(0)} - 2x_3^{(0)})] = \frac{1}{3}[23 + 6(-3.1) - 2(0.9)] = 0.8667,$$

$$x_2^{(1)} = [-8 + 4x_1^{(1)} + x_3^{(0)}] = [-8 + 4(0.8667) + 0.9] = -3.6332,$$

$$x_3^{(1)} = [17 - x_1^{(1)} + 3x_2^{(1)}]/7 = \frac{1}{7}[17 - (0.8667) + 3(-3.6332)] = 0.7477.$$

Second iteration

$$x_1^{(2)} = \frac{1}{3}[23 + 6x_2^{(1)} - 2x_3^{(1)})] = \frac{1}{3}[23 + 6(-3.6332) - 2(0.7477)] = -0.0982,$$

$$x_2^{(2)} = [-8 + 4x_1^{(2)} + x_3^{(1)}] = [-8 + 4(-0.0982) + 0.7477] = -7.6451,$$

$$x_3^{(2)} = \frac{1}{7}[17 - x_1^{(2)} + 3x_2^{(2)}] = \frac{1}{7}[17 + 0.0982 + 3(-7.6451)] = -0.8339.$$

Third iteration

$$x_1^{(3)} = \frac{1}{3}[23 + 6x_2^{(2)} - 2x_3^{(2)})] = \frac{1}{3}[23 + 6(7.6451) - 2(-0.8339)] = -7.0676,$$

$$x_2^{(3)} = [-8 + 4x_1^{(3)} + x_3^{(2)}] = [-8 + 4(-7.0676) - 0.8339] = -37.1043,$$

$$x_3^{(3)} = \frac{1}{7}[17 - x_1^{(3)} + 3x_2^{(3)}] = \frac{1}{7}[17 + 7.0676 + 3(-37.1043)] = -12.4636.$$

It can be observed that the iterations are diverging very fast. Now, we exchange the first and second equations to obtain the system

$$4x_1 + x_2 - x_3 = -8,$$

$$3x_1 - 6x_2 + 2x_3 = 23,$$

$$x_1 - 3x_2 + 7x_3 = 17.$$

The system of equations is now diagonally dominant. Gauss-Seidel method gives iteration

$$x_1^{(k+1)} = [8 + x_2^{(k)} - x_3^{(k)}]/4,$$

$$x_2^{(k+1)} = [23 - 3x_1^{(k+1)} - 2x_3^{(k)}]/6,$$

$$x_3^{(k+1)} = [17 - x_1^{(k+1)} + 3x_2^{(k+1)}]/7.$$

Starting with the initial approximations $x_1 = 0.9, x_2 = 3.1, x_3 = 0.9$, we obtain the following results.

First iteration

$$x_1^{(1)} = \frac{1}{4}[8 + x_2^{(0)} - x_3^{(0)}] = \frac{1}{4}[8 - 3.1 - 0.9] = 1.0,$$

$$x_2^{(1)} = \frac{-1}{6}[23 - 3x_1^{(1)} - 2x_3^{(0)}] = \frac{-1}{6}[23 - 3(1.0) - 2(0.9)] = -3.0333,$$
$$x_3^{(1)} = \frac{1}{7}[17 - x_1^{(1)} + 3x_2^{(1)}] = [17 - 1.0 + 3(-3.0333)] = 0.9857.$$

Second iteration

$$x_1^{(2)} = \frac{1}{4}[8 + x_2^{(1)} - x_3^{(1)}] = \frac{1}{4}[8 - 3.0333 - 0.9857] = -0.9953,$$
$$x_2^{(2)} = \frac{-1}{6}[23 - 3x_1^{(2)} - 2x_3^{(1)}] = \frac{-1}{6}[23 - 3(0.9953) - 2(0.9857)] = -3.0071,$$
$$x_3^{(2)} = \frac{1}{7}[17 - x_1^{(2)} + 3x_2^{(2)}] = \frac{1}{7}[17 - 0.9953 + 3(-3.0071)] = 0.9976.$$

Third iteration

$$x_1^{(3)} = \frac{1}{4}[8 + x_2^{(2)} - x_3^{(2)}] = \frac{1}{4}[8 - 3.00710.9976] = 0.9988,$$
$$x_2^{(3)} = \frac{-1}{6}[23 - 3x_1^{(3)} - 2x_3^{(2)}] = \frac{-1}{6}[23 - 3(0.9988) - 2(0.9976)] = 3.0014,$$
$$x_3^{(3)} = \frac{1}{7}[17 - x_1^{(3)} + 3x_2^{(3)}] = \frac{1}{7}[17 - 0.9988 + 3(-3.0014)] = 0.9996.$$

Fourth iteration

$$x_1^{(4)} = \frac{1}{4}[8 + x_2^{(3)} - x_3^{(3)}] = \frac{1}{4}[8 - 3.00140.9996] = 0.9998,$$
$$x_2^{(4)} = \frac{-1}{6}[23 - 3x_1^{(4)} - 2x_3^{(3)}] = \frac{-1}{6}[23 - 3(0.9998) - 2(0.9996)] = -3.0002,$$
$$x_3^{(4)} = \frac{1}{7}[17 - x_1^{(4)} + 3x_2^{(4)}] = \frac{1}{7}[17 - 0.9998 + 3(-3.0002)] = 0.9999.$$

We find $|x_1^{(4)} - x_1^{(3)}| = |0.9998 - 0.9988| = 0.0010,$
$|x_2^{(4)} - x_2^{(3)}| = |3.0002 + 3.0014 = 0.0012, /$
$|x_3^{(4)} - x_3^{(3)}| = |0.9999 - 0.9996| = 0.0003.$
Since, all the errors in magnitude are less than 0.005, the required solution is

$$x_1 = 0.9998, x_2 = 3.0002, x_3 = 0.9999.$$

Rounding to two decimal places, we get $x_1 = 1.0, x_2 = 3.0, x_3 = 1.0.$

Most iterative methods might converge or not. However, certain class of systems of simultaneous equations do always converges to a solution using Gauss-Seidal method.

This class of system of equations is where the coefficient matrix A in $AX = C$ is diagonally dominant, that is

$$|a_{ii}| \geq \sum_{i=1, j \neq i}^{n} |a_{ij}| \quad \text{for all '}i\text{'}$$

and $|a_{ii}| > \sum_{i=1, j \neq i}^{n} |a_{ij}|$ for all 'i'. If a system of equations has a coefficient matrix that is not diagonally dominant, it may or may not converge. Fortunately, many physical systems that result in simultaneous linear equations have diagonally dominant coefficient matrices, which then assures convergence for iterative methods such as Gauss-Seidal method of solving simultaneous linear equations.

Example 1.3.3 *Given the system of equations.*

$$\begin{aligned} 12x_1 + 3x_2 - 5x_3 &= 1, \\ x_1 + 5x_2 + 3x_3 &= 28, \\ 3x_1 + 7x_2 + 13x_3 &= 76. \end{aligned}$$

We find the solution by using the initial guess

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Solution

The coefficient matrix

$$A = \begin{bmatrix} 12 & 3 & -5 \\ 1 & 5 & 3 \\ 3 & 7 & 13 \end{bmatrix}$$

is diagonally dominant as

$$\begin{aligned}
|a_{11}| &= |12| = 12 \geq |a_{12}| + |a_{13}|| = |3| + |5| = 8 \\
|a_{22}| &= |5| = 5 \geq a_{21} + |a_{23}| = |1| + |3| = 4 \\
|a_{33}| &= |13| = 13 \geq |a_{31}| + a_{32} = |3| + |7| = 10
\end{aligned}$$

and the inequality is strictly greater than for at least one row. Hence the solution should converge using Gauss-Seidal method. Rewriting the equations, we get

$$\begin{aligned}
x_1 &= \frac{1 - 3x_2 + 5x_3}{12}, \\
x_2 &= \frac{28 - x_1 - 3x_3}{5}, \\
x_3 &= \frac{76 - 3x_1 - 7x_2}{13}.
\end{aligned}$$

Assuming an initial guess of

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Iteration 1:

$$\begin{aligned}
x_1 &= \frac{1 - 3(0) + 5(1)}{12}, \\
&= 0.50000 \\
x_2 &= \frac{28 - (0.5) - 3(1)}{1}, \\
&= 4.9000 \\
x_3 &= \frac{76 - 3(0.50000) - 7(4.9000)}{13}. \\
&= 3.0923
\end{aligned}$$

The absolute relative approximate error at the end of first iteration is

$$\begin{aligned}
|\epsilon_a|_1 &= \left|\frac{0.50000-1.0000}{0.50000}\right| \times 100 \\
&= 67.662\%, \\
|\epsilon_a|_2 &= \left|\frac{4.9000-0}{4.9000}\right| \times 100 \\
&= 100.000, \% \\
|\epsilon_a|_3 &= \left|\frac{3.0923-1.0000}{3.0923}\right| \times 100 \\
&= 67.662\%.
\end{aligned}$$

The maximum absolute relative approximate error is 100.000%

Iteration 2:

$$\begin{aligned}
x_1 &= \frac{1-3(4.9000)+5(3.0923)}{12} \\
&= 0.14679, \\
x_2 &= \frac{28-(0.14679)-3(3.0923)}{5} \\
&= 3.7153, \\
x_3 &= \frac{76-3(0.14679)-7(4.900)}{13} \\
&= 3.8118.
\end{aligned}$$

At the end of second iteration, the absolute relative approximate error is

$$\begin{aligned}
|\epsilon_a|_1 &= \left|\frac{0.14679-0.50000}{0.14679}\right| \times 100 \\
&= 240.6\%, \\
|\epsilon_a|_2 &= \left|\frac{3.7153-4.9000}{3.7153}\right| \times 100 \\
&= 31.887\%, \\
|\epsilon_a|_3 &= \left|\frac{3.8118-3.0923}{3.8118}\right| \times 100 \\
&= 18.876\%.
\end{aligned}$$

The maximum absolute relative approximate error is 240.62%. This is greater than the value of 67.612% we obtained in the first iteration. Is the solution diverging? No, as you conduct more iterations, the solution converges as follows.

Iteration	a_1	$\|\epsilon_a\|_1$	a_2	$\|\epsilon_a\|_2$	a_3	$\|\epsilon_a\|_3$
1	0.50000	67.662	4.900	100.00	3.0923	67.662
2	0.14679	240.62	3.7153	31.887	3.8118	18.876
3	0.74275	80.23	3.1644	17.409	3.9708	4.0042
4	0.94675	21.547	3.0281	4.5012	3.9971	0.65798
5	0.99177	4.5394	3.0034	0.82240	4.0001	0.07499
6	0.99919	0.74260	3.0001	0.11000	4.0001	0.00000

This is close to the exact solution vector of

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$$

Example 1.3.4 *Given the system of equation*

$$\begin{aligned} 3x_1 + 7x_2 + 13x_3 &= 76, \\ x_1 + 5x_2 + 3x_3 &= 28, \\ 12x_1 + 3x_2 - 5x_3 &= 1. \end{aligned}$$

find the solution using Gauss-Seidal method. Use $\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$ as the initial guess.

Solution

Rewriting the equations, we get

$$
\begin{aligned}
x_1 &= \frac{76 - 7x_2 - 13x_3}{3}, \\
x_2 &= \frac{28 - x_1 - 3x_3}{5}, \\
x_3 &= \frac{1 - 12x_1 - 3x_2}{-5}.
\end{aligned}
$$

Assuming an initial guess of

$$
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}
$$

the next six iterative values are given in the table below

Iteration	a_1	$\lvert\epsilon_a\rvert_1$	a_2	$\lvert\epsilon_a\rvert_2$	a_3	$\lvert\epsilon_a\rvert_3$
1	21.000	110.71	0.80000	100.00	5.0680	98.027
2	-196.15	109.83	14.421	94.453	-462.30	110.96
3	-1995.0	109.90	-116.02	112.43	47636	109.80
4	-20149	109.89	1204.6	109.63	-47636	109.90
5	2.0364×10^5	109.90	-12140	109.92	4.8144×10^5	109.89
6	-2.0579×10^5	1.0990	1.2272×10^5	109.89	-4.8653×10^6	109.89

We can see that this solution is not converging and the coefficient matrix is not diagonally dominant. The coefficient matrix

$$
A = \begin{bmatrix} 3 & 7 & 13 \\ 1 & 5 & 3 \\ 12 & 3 & -5 \end{bmatrix}
$$

is not diagonally dominant as

$$
|a_{11}| = |3| = 3 \leq \ |a_{12}| + |a_{13}| = |7| = |13| = 20
$$

Hence Gauss-Seidal method may or may not converge. However, it is the same set of equations as the previous example and that converged. The only difference is that we exchanged first and the third equation with each other and that made the coefficient matrix not diagonally dominant. So it is possible that a system of equations can be made diagonally dominant if one exchanges the equations with each other. But it is not possible for all cases. For example, the following set of equations.

$$\begin{aligned} x_1 + x_2 + x_3 &= 3, \\ 2x_1 + 3x_2 + 4x_3 &= 9, \\ x_1 + 7x_2 + x_3 &= 9. \end{aligned}$$

can not be rewritten to make the coefficient matrix diagonally dominant.

Chapter 2

The Singular value Decomposition

Matrix decompositions play a critical role in numerical linear algebra. The singular value decomposition (SVD) is a matrix decomposition that applies to any matrix, real, or complex. The SVD is a powerful tool for many matrix computations because it reveals a great deal about the structure of a matrix, we will use the built-in MATLAB command svd to compute it. More information can be found in[3,6]

2.1 The SVD theorem

If A is an $m \times n$ matrix, then $A^T A$ is an $n \times n$ symmetric matrix with nonnegative eigenvalues. The singular values of an $m \times n$ matrix are the square roots of the eigenvalues of $A^T A$, and the 2-norm of a matrix is the largest singular value. The SVD factors A into a product of two orthogonal matrices and a diagonal matrix of its singular values.

Theorem 2.1.1 *Let $A \in \mathbb{R}^{m \times n}$ be a matrix having r positive singular values, $m \geq n$. Then there exist orthogonal matrices $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$, and a diagonal matrix $\tilde{\Sigma} \in \mathbb{R}^{m \times n}$ such that*

$$A = U\Sigma V^T \tag{2.1}$$

$$\tilde{\Sigma} = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \tag{2.2}$$

where $\Sigma = diag(\sigma_1, \sigma, \ldots, \sigma_r)$, *and* $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_r > 0$ *are the positive singular values of* A *.*

The columns of U and V are called the *left* and *right singular vectors*, respectively. The largest singular values are denoted, respectively, as $\sigma_{\max}$ and $\sigma_{\min}$.

Example 2.1.1 *The matrix* $A = \begin{bmatrix} 2 & 2 \\ 3 & 3 \end{bmatrix}$ *has SVD*

$$\begin{bmatrix} 2 & 2 \\ 3 & 3 \end{bmatrix} = \begin{bmatrix} -0.5547 & -0.8321 \\ -0.8321 & 0.5547 \end{bmatrix} \begin{bmatrix} 5.0990 & 0 \\ 0 & 0.0000 \end{bmatrix} \begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

The rank of A *is* 2 *.*

Example 2.1.2 *Let* $A = \begin{bmatrix} 1 & -1 & 3 \\ 1 & 0 & 1 \\ 1 & 2 & 0 \end{bmatrix}$ *and* $B = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 2 \\ 2 & 1 & 1 \end{bmatrix}$. *Here are SVDs for each matrix:*

$$\begin{aligned} A &= \begin{bmatrix} -0.9348 & 0.0194 & 0.3546 \\ 0.3465 & -0.2684 & -0.8988 \\ 0.0778 & -0.9631 & 0.2577 \end{bmatrix} \begin{bmatrix} 3.5449 & 0 & 0 \\ 0 & 2.3019 & 0 \\ 0 & 0 & 0.3676 \end{bmatrix} \begin{bmatrix} -0.3395 & 0.3076 & -0.88 \\ -0.5266 & 2.3019 & -0.09 \\ -0.7794 & 0.4371 & 0.448 \end{bmatrix} \\ r &= 3, \sigma_1 = 3.5449, \sigma_2 = 2.3019, \sigma_3 = 0.36081 \\ B &= \begin{bmatrix} -0.1355 & 0.8052 & -0.5774 \\ -0.6295 & -0.5199 & -0.5774 \\ -0.07651 & 0.2852 & 0.5774 \end{bmatrix} \begin{bmatrix} 3.1058 & 0 & 0 \\ 0 & 2.0867 & 0 \\ 0 & 0 & 0.0000 \end{bmatrix} \begin{bmatrix} -0.7390 & -0.2900 & -0 \\ 0.4101 & 0.5226 & -0 \\ 0.5345 & -0.8018 & -0 \end{bmatrix} \\ r &= 3, \sigma_1 = 3.5449, \sigma_2 = 2.3019, \sigma_3 = 0.36081 \end{aligned}$$

The rank of A *is* 3, *and the rank of* B *is* 2.

Example 2.1.3 *Consider the matrix*

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.3}$$

An SVD is

$$
\begin{aligned}
U &= \begin{bmatrix} -0.5000 & -0.8660 & 0 & 0.0000 & 0 & 0.0000 & 0 \\ 0.0000 & 0.0000 & 1 & 0.0000 & 0 & 0.0000 & 0 \\ 0.5000 & -0.2887 & 0 & 0.8165 & 0 & 0.0000 & 0 \\ 0.0000 & 0.0000 & 0 & 0.0000 & -1 & 0.0000 & 0 \\ -0.5000 & 0.2887 & 0 & 0.4082 & 0 & 0.7071 & 0 \\ 0.5000 & -0.2887 & 0 & -0.4082 & 0 & 0.7071 & 1 \\ 0.0000 & 0.0000 & 0 & 0.0000 & 0 & 0.0000 & 1 \end{bmatrix}, \\
\tilde{\Sigma} &= \begin{bmatrix} 4.4721 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
V &= \begin{bmatrix} -0.4472 & -0.3651 & -0.6712 & 0.4614 & 0.0581 \\ -0.4472 & -0.3651 & -0.0671 & 0.8074 & 0.1016 \\ -0.4472 & -0.5477 & 0.0000 & 0.0883 & 0.7016 \\ -0.4472 & -0.3651 & -0.7383 & 0.3460 & 0.0435 \\ -0.4472 & -0.5477 & -0.0000 & 0.0883 & 0.7016 \end{bmatrix} \\
\Sigma &= [4.4721], \sigma_1 = 4.4721, r = 1 \qquad (2.4)
\end{aligned}
$$

The rank of A is 1.

2.2 Using the SVD to determine properties of a matrix

The rank of a matrix is the number of linearly independent columns or rows.We notice that in Example 2.1.2, the matrix A has three nonzero singular values, and the matrix

B has two. The matrix of Example 2.1.3 has only one nonzero singular value. The rank of the matrices is $3, 2$, and 1, respectively.The rank of a matrix is the number of nonzero singular values in $\tilde{\Sigma}$. The result will allow us to show the relationship between the rank of a matrix and its singular values.

Theorem 2.2.1 *If A is an $m \times n$ matrix, X is an invertible $m \times m$ matrix, and Y is an invertible $n \times n$ matrix, then rank (XAY) = rank (A)*

Proof. Since X is invertible, it can be written as a product of elementary row matrices, so $X = E_k^{(X)} E_{k-1}^{(X)} \dots E_2^{(X)} E_1^{(X)}$. Similarly, Y is a product of elementary row matrices, $Y = E_p^{(Y)} E_{p-1}^{(Y)} \dots E_2^{(Y)} E * (Y)_1$, and so

$$XAY = E_k^{(X)} E_{k-1}^{(X)} \dots E_2^{(X)} E_1^{(X)} A E_p^{(Y)} E_{p-1}^{(Y)} \dots E_2^{(Y)} E * (Y)_1.$$

The product of the elementary row matrices on the left performs elementary row operations on A, and this does not change the rank of A. The product of elementary row matrices on the left perform elementary column operations, which also do not alter rank. Thus, rank(XAY)=$rank(A)$.

Theorem 2.2.2 *The rank of a matrix A is the number of nonzero singular values.*

Proof. Let $A = U\tilde{\Sigma}V^T$ be the SVD of A. Orthogonal matrices are invertible, so by

$$\text{rank}(A) = \text{rank}(U\tilde{\Sigma}V^T) = \text{rank}(\tilde{\Sigma}).$$

The rank of $\tilde{\Sigma}$ is r, since

$$[\sigma_1\, 0\, 0\, \dots\, 0]^T, [0\, \sigma_2\, 0\, \dots\, 0]^T, [0\, 0\, \sigma_3]^T, [0\, 0\, \dots\, \sigma_r\, 0\, \dots\, 0]^T$$

is a basis for the column space of $\tilde{\Sigma}$.

From the components of the SVD, we can determine other properties of the original matrix. Recall that the null space of a matrix A, written null(A), is the set of *vectors* x for which $Ax = 0$, and the range of A is the set of all linear combinations of the columns of A (the column space of A). Let $u_i, 1 \le i \le m$ and $v_i, 1 \le i \le n$ be the column vectors of U and V, respectively. Then

$$Av_i = U\tilde{\Sigma}V^T v_i.$$

The matrix V^T can be written as $\begin{bmatrix} v_1^T \\ \vdots \\ v_i^T \\ \vdots \\ v_n^T \end{bmatrix}$, where the v_i are the orthogonal columns of V. The product $V^T v_i = \begin{bmatrix} v_1^T \\ \vdots \\ v_i^T \\ \vdots \\ v_n^T \end{bmatrix} v_i = e_i$, where $e_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$ is the ith standard basis vector n $\mathbb{R}^n$. Now,

$$\tilde{\Sigma}e_i = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & \dots & 0 & 0 & 0 \\ 0 & \sigma_2 & \dots & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & \sigma_i & \dots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 & \ddots & \vdots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \dots & \sigma_r & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \end{bmatrix} e_i = \begin{bmatrix} 0 \\ \vdots \\ \sigma_i \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}, \quad i \le r,$$

and

$$U \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \sigma_i \\ \vdots \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1r} & \dots & u_{1m} \\ u_{21} & u_{22} & \dots & u_{2r} & \dots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & u_{rr} & \dots & u_{nm} \\ \vdots & \vdots & \dots & \vdots & \ddots & \dots \\ u_{m1} & u_{m2} & \dots & u_{mr} & \dots & u_{mm} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sigma_i \\ \vdots \\ 0 \end{bmatrix} = \sigma_i u_i, \quad 1 \leq i \leq r.$$

For v_i, $r+1 \leq i \leq m$, we have

$$Av_i = U\tilde{\Sigma}e_i = \begin{bmatrix} u_{11} & \dots & u_{1i} & \dots & u_{1m} \\ u_{21} & \dots & u_{2i} & \dots & u_{2m} \\ \vdots & \ddots & \vdots & \dots & \vdots \\ \vdots & \dots & u_{ii} & \dots & u_{im} \\ \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{m1} & \dots & u_{mi} & \dots & u_{mm} \end{bmatrix} \begin{bmatrix} \sigma_1 & \dots & 0 & \dots & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \dots & \vdots & \dots & \dots & \vdots \\ 0 & \dots & \sigma_r & \ddots & \vdots & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & 0 & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & \vdots \\ \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = 0$$

and $Av_i = 0, r+1 \leq i \leq m$.

In summary, we have

$$Av_i = \sigma_i u_i, \quad \sigma_i \neq 0, 1 \leq i \leq r$$

$$Av_i = 0, \quad r+1 \leq i \leq n$$

Since U and V are orthogonal matrices, all u_i and v_i are linearly independent. For $1 \leq i \leq r$, $Av_i = \sigma_i u_i, \sigma_i = 0$, and $u_i, 1 \leq i \leq r$ is in the range of A. Since by Theorem 2.2.1 the rank of A is r, the u_i are a basis for the range of A. For $r+1 \leq i \leq n$, $Av_i = 0$, so v_i is in $\text{null}(A)$. Since $\text{rank}(A) + \text{nullity}(A) = n$, $\text{nullity}(A) = n - r$. There are

$n-(r+1)+1=n-r$ orthogonal vectors v_i, so the v_i, $r+1 \leq i \leq n$, are a basis for the null space of A.

Example 2.2.1 *Let* $B = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 2 \\ 2 & 1 & 1 \end{bmatrix}$ *be the matrix in Example 2.1.2. From the SVD, the vector* $\begin{bmatrix} -0.1355 \\ -0.6295 \\ -0.7651 \end{bmatrix}$ *and* $\begin{bmatrix} 0.8052 \\ -0.5199 \\ 0.2852 \end{bmatrix}$ *are a basis for the rang of* B*, and the vector* $\begin{bmatrix} 0.5345 \\ -0.8018 \\ -0.2673 \end{bmatrix}$ *is a basis for the null space of* B*. Remember when looking at the decomposition of* B, V^T *appears, not* V*.*

2.3 SVD and matrix norms

The SVD provides a means of computing the 2-norm of a matrix, since $\|A\|_2 = \sqrt{\sigma_1}$. If A is invertible, then $\|A^{-1}\|_2 = \sqrt{\frac{1}{\sigma_n}}$. The SVD can be computed accurately, so using it is an effective way to find the 2-norm. The SVD also provides a means of computing the Frobenius norm. There is means of computing the Frobenius norm using the singular values of matrix A. Before developing the formula, we need to prove the invariance of the Frobenius norm under multiplication by orthogonal matrices.

Lemma 2.3.1 *If* U *is an* $m \times m$ *orthogonal matrix, and* V *is an* $n \times n$ *orthogonal matrix, then* $\|UAV\|_F^2 = \|A\|_F^2$*.*

Proof.

$$\|UA\|_F^2 = \operatorname{trace}\left((UV)^T(UV)\right) = \operatorname{trace}\left(\left(A^TU^T\right)(UA)\right) = \operatorname{trace}\left(A^TIA\right) = \operatorname{trace}\left(A^TA\right) = \|A\|_F^2,$$

showing that the Frobenius norm is invariant under left multiplication by an orthogonal matrix. Now,

$$\|AV\|_F^2 = \text{trace}\left((AV)(AV)^T\right) = \text{trace}\left((AV)\left(V^T A^T\right)\right) = \text{trace}(AA^T) = \|A\|_F^2,$$

so the Frobenius norm is invariant under right multiplication by an orthogonal matrix. Now form the complete product.

$$\|UAV\|_F^2 = \|U(AV)\|_F^2 = \|AV\|_F^2 = \|A\|_F^2.$$

Theorem 2.3.1 $\|A\|_F = \left(\sum_{r}^{i=1} \sigma_i^2\right)^{\frac{1}{2}}$

Proof. By the SVD, there exist orthogonal matrices U and V such that $A = U\tilde{\Sigma}V^T$. Then, $\|A\|_F = \|U\tilde{\Sigma}V^T\|_F = \|\tilde{\Sigma}\|_F$ by Lemma 2.3.1. The only nonzero entries in $\tilde{\Sigma}$ are the singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$ so $\|A\|_F = (\sum_{i=1}^r \sigma_i^2)^{\frac{1}{2}}$.

2.4 Computing the SVD using MATLAB

In this section ,we use the following MATLAB function svd to compute the SVD . The MATLAB function svd computes the SVD:

1. `[U S V] = svd(A)`
2. `S = svd(A)`

Form 2 returns only the singular values in descending order in the vector S.

Example 2.4.1 *Find the SVD for the matrix. Notice that* $\sigma_3 = 4.8021 \times 10^{-16}$ *and yet the rank is* 2*. In this case, the true value is* 0*, but roundoff error caused svd to return a very small singular value. MATLAB computes the rank using the SVD,that*

σ_3 *is* 0.

$$A = \begin{bmatrix} 1 & 4 & 2 \\ -1 & 0 & 2 \\ 5 & -1 & -11 \\ 0 & 2 & 2 \\ 1 & 1 & -1 \end{bmatrix},$$

```
>> [U,S,V] = svd(A)
U =
    0.15897      0.8589       -0.2112     -0.3642     -0.24447
    0.17398     -0.073783     -0.9063      0.22223     0.30581
   -0.95316      0.17263      -0.18552     0.14793    -0.073387
    0.16648      0.39256       0.21339     0.87898    -0.0063535
   -0.090746     0.27006       0.23251    -0.15324     0.91722

S =
    12.691            0               0
         0       4.7905               0
         0            0      4.8021e-16
         0            0               0
         0            0               0

V =
    -0.38387      0.43125        0.8165
      0.1443      0.90139      -0.40825
     0.91204     0.038895       0.40825

>> rank(A)
ans =
```

```
     2
```

The function svd applies equally well to a matrix of dimension $m \times n, m < n$. Of course, in this case the rank does not exceed m.

Example 2.4.2 *Let* $A = \begin{bmatrix} 7 & 9 & -5 & 10 & 10 & -8 \\ 9 & 3 & 1 & -7 & 0 & -2 \\ -8 & 8 & 10 & 10 & 6 & 9 \end{bmatrix}$

```
>> [U S V] = svd(A)


U =

   -0.42586   -0.89303    -0.14539
    0.26225   -0.27562     0.9248
   -0.86595    0.35571     0.35157


S =

     22.577           0           0           0           0           0
          0      20.176           0           0           0           0
          0           0      9.5513           0           0           0


V =

    0.27935     -0.57383      0.4704      0.60332     0.008204     0.085659
   -0.44176      -0.2983     0.44795     -0.37549     -0.51939     -0.32318
   -0.27763    0.38396       0.54102      0.10972      0.60466     -0.32426
   -0.65349      -0.1707    -0.46191      0.53138    0.0076852     -0.21915
   -0.41876     -0.33684    0.068636     -0.28352      0.38297       0.6924
   -0.21753       0.5401      0.2594      0.34673     -0.46673       0.5056


>> rank(A)

ans =
```

```
3
```

Since $m = 3$ and the rank is 3, A has full rank.

2.5 Computing A^{-1}

We know that the inverse is often difficult to be computed accurately and that,to compute A^{-1}, the SVD can be used. Since A is invertible, the matrix $\tilde{\Sigma}$ cannot have a 0 on its diagonal (rank would be $< n$), so $\tilde{\Sigma} = \Sigma$. From $A = U\Sigma V^T$, $A^{-1} = (V^T)^{-1}\Sigma^{-1}U^{-1}$, where all the matrices have dimension $n \times n$. U and V are orthogonal, so

$$A^{-1} = V \begin{bmatrix} \frac{1}{\sigma_1} & & & & 0 \\ & \frac{1}{\sigma_2} & & & \\ & & \ddots & & \\ & & & \frac{1}{\sigma_{n-1}} & \\ 0 & & & & \frac{1}{\sigma_n} \end{bmatrix} U^T.$$

Example 2.5.1 *Let* $A = \begin{bmatrix} 1 & -1 & 3 \\ 4 & 2 & 3 \\ 5 & 1 & -1 \end{bmatrix}$.

```
>> A = [1 -1 3;4 2 3;5 1 -1];
>> [U S V] = svd(A);
>> Ainv = V*diag(1./diag(S))*U

Ainv =
    0.1190   -0.0476    0.2143
   -0.4524    0.3810   -0.2143
    0.1429    0.1429   -0.1429
```

```
>> inv(A)

ans =
    0.1190   -0.0476    0.2143
   -0.4524    0.3810   -0.2143
    0.1429    0.1429   -0.1429
```

Example 2.5.2 *We solved this system in chapter 1(Example 1.2.1) by Gauss-Jacobi and now, we solve it by SVD*

$$\begin{bmatrix} 20 & 1 & -2 \\ 3 & 20 & -1 \\ 2 & -3 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 17 \\ -18 \\ 25 \end{bmatrix} \tag{2.5}$$

$$\begin{bmatrix} -0.5091 & 0.6685 & -0.5421 \\ -0.7041 & 0.0388 & 0.7091 \\ 0.4951 & 0.7427 & 0.4509 \end{bmatrix} \begin{bmatrix} 22.8325 & 0 & 0 \\ 0 & 20.1601 & 0 \\ 0 & 0 & 17.3276 \end{bmatrix}$$

$$\begin{bmatrix} -0.4951 & 0.7427 & -0.4509 \\ -0.7041 & -0.0388 & 0.7091 \\ 0.5091 & 0.6685 & 0.5421 \end{bmatrix}$$

$x_1 = 1, x_2 = -1, x_3 = 1.$

Example 2.5.3 *We solved this system in chapter 1(Example 1.3.1) by Gauss-Siedel and now, we solve it by SVD*

$$\begin{bmatrix} 45 & 2 & 3 \\ 3 & 22 & 2 \\ 5 & 1 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 58 \\ 47 \\ 67 \end{bmatrix} \tag{2.4}$$

$$\begin{bmatrix} -0.9777 & 0.1730 & -0.1188 \\ -0.1203 & -0.9258 & -0.3584 \\ -0.1720 & -0.3361 & 0.9260 \end{bmatrix} \begin{bmatrix} 45.9483 & 0 & 0 \\ 0 & 22.1446 & 0 \\ 0 & 0 & 18.9571 \end{bmatrix}$$

$$\begin{bmatrix} -0.9841 & 0.1502 & -0.0945 \\ -0.1039 & -0.9193 & -0.3796 \\ -0.1439 & -0.3637 & 0.9203 \end{bmatrix}$$

$x_1 = 1.0115, x_2 = 1.7247, x_3 = 3.0109.$

Chapter 3

Using SVD to Solve Least-Squares Problems

3.1 Least-Squares Problems

In many areas such as curve fitting, statistics, and geodetic modeling, A is either singular or has dimension $m \times n, m = n$. If $m > n$, there are more equations than unknowns, and the system is said to be overdetermined. In most cases, overdetermined systems have no solution. In the case $m < n$, there are more unknowns than equations, and we say such systems are underdetermined. In this situation, there are usually an infinite number of solutions.

Since singular, over- and underdetermined systems do not give us a solution in the exact sense, the solution is to find a vector $\mathbf{x}$ such that $\mathbf{Ax}$ is as close as possible to $\mathbf{b}$. A way of doing this is to find a vector $\mathbf{x}$ such that the residual $r(x) = \|Ax - b\|_2$ is a minimum. Recall that the Euclidean norm, $\|.\|_2$, of a vector in R^n is $\sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$, so if we want to minimize $\|Ax - b\|_2$, we call $\mathbf{x}$ a least-squares solution. Finding a least-squares solution to $\mathbf{Ax} = \mathbf{b}$ is known as the linear least-squares problem.

Definition 3.1.1 *The least-squares problem*

Given a real $m \times n$ matrix A and a real vector b, find a real vector $\boldsymbol{x} \in R^n$ such that the function $r(x) = \|Ax - b\|_2$ is minimized. It is possible that the solution x will not be unique [1,3,6].

Assume that $m > n$. Since $x \in R^n$, and A is an $m \times n$ matrix, Ax is a linear transformation from R^n to R^m, and the range of the transformation,$R(A)$, is a subspace of R^m. Given any $y \in R(A)$, there is an $x \in R^n$ such that $Ax = y$. If $b \in R^m$ is in $R(A)$, we have a solution. If b is not in $R(A)$, consider the vector $Ax - b$ that joins the endpoints of the vectors Ax and b.

Since b is not in $R(A)$, project b onto the plane $R(A)$ to obtain a vector $u \in R(A)$. There must be a vector $x \in R^n$ such that $Ax = u$. The distance between the two points is $\|Ax - b\|_2$ is as small as possible, so x is the solution we want.

The vector $b - Ax$ is orthogonal to $R(A)$, and since every vector in $R(A)$ is a linear combination of the columns of A (vectors in R^m), it must be the case that $b - Ax$ is orthogonal to the every column of A. Mathematically this means that the inner product of $b - Ax$ with each column of A must be zero. If

$$a_i = \begin{pmatrix} a_{1,i} \\ a_{2,i} \\ \vdots \\ a_{m-1,i} \\ a_{m,i} \end{pmatrix}$$

is column i, then $\langle ai, b - Ax \rangle = a_i^T(b - Ax) = 0$, $1 \le i \le n$, and

$$A^T(b - Ax) = 0 \ ,$$

3.2 existence and uniqueness of least-squares solutions

In order to prove the existence and uniqueness to the solution of the least-squares problem, we will consider the case $m \geq n$. Let A be an $m \times n$ matrix. Then, each of the n columns has m components and is a member of R^m, and each of m rows has n components and is a member of R^n. The columns of A span a subspace of R^m, and the rows of A span a subspace of R^m. The column rank of A is the number of linearly independent columns of A, and the row rank of A is the number of linear independent rows of A. proves that the column rank and row rank of A are equal.

3.2.1 Definition

An $m \times n$ matrix A has full rank if $rank(A) = min(m, n)$. If $m \geq n$, and A has full rank, then $rank(A) = n$, and the columns of A are linearly independent.

3.2.2 lemma

Let A be an $m \times n$ matrix, $m \geq n$. A has full rank if and only if the $n \times n$ matrix $A^T A$ is nonsingular

Proof. We proof by contradiction. Assume A has full rank, but $A^T A$ is singular. In this case, the $n \times n$ homogeneous system $A^t A x = 0$ has a nonzero solution x. As a result, $x^t A^t A x = 0$, which says that Ax, $Ax = |Ax|_2^2 = 0$, so $Ax = 0$, and A cannot have full rank.

Again, use proof by contradiction. Assume $A^T A$ is nonsingular, but A does not have full rank. Since A is rank deficient, there is a nonzero vector x such that $Ax = 0$. As a result, $A^T A x = 0$, $x \neq 0$, so $A^T A$ is singular.

Theorem 3.2.1 *a. Given an $m \times n$ matrix A with $m \geq n$ and an $m \times 1$ column vector b, an $n \times 1$ column vector x exists such that x is a least-squares solution to $Ax = b$ if and only if x satisfies the normal equations*

$$A^T Ax = A^T b.$$

b. The least-squares solution x is unique if and only if A has full rank.

Proof. *To prove part (1), assume that $A^T Ax = A^T b$, so that x is a solution of the normal equations. Now, if x is any vector in R^n,*

$$\begin{aligned}
\|Ax - b\|_2^2 &= \|Ax - A\bar{x} + A\bar{x} - b\|_2^2 = \langle[(A\bar{x} - b) + A(x - \bar{x})], [(A\bar{x} - b) + A(x - \bar{x})]\rangle \\
&= \|A\bar{x} - b\|_2^2 + 2\langle A(x - \bar{x}), (A\bar{x} - b)\rangle + \|A(x - \bar{x})\|_2^2 \\
&= \|A\bar{x} - b\|_2^2 + 2(A(x - \bar{x}))^T(A\bar{x} - b) + \|A(x - \bar{x})\|_2^2 \\
&= \|Ax \bar{-} b\|_2^2 + 2(x - \bar{x})^T(A^T A\bar{x} - A^T b) + \|A(x - \bar{x})\|_2^2 \\
&= \|A\bar{x} - b\|_2^2 + \|A(x - \bar{x})\|_2^2 \\
&\geq \|A\bar{x} - b\|_2^2,
\end{aligned}$$

and $\bar{x}$ is a solution to the least-squares problem

Now assume that $\bar{x} \in R^n$ is a solution to the least-squares problem, so that $\|A\bar{x} - b\|_2$ is minimum. Thus,$\|b - A\bar{x}\|_2^2 \geq \|b - Ay\|_2^2$ for any $y \in R^n$. Given any vector $z \in R^n$, let $y = x + tz$, where t is a scalar. Then,

$$\begin{aligned}
\|b - A\bar{x}\|_2^2 \leq \|b - A(x + tz)\|_2^2 &= ([b - A\bar{x}] - tAz)^T([b - A\bar{x}]tAz) \\
&= \|b - A\bar{x}\|_2^2 - 2t(b - A\bar{x})^T Az + t^2\|Az\|_2^2 \quad .
\end{aligned}$$

Thus,

$$0 \leq -2t(b - Ax)^T Az + t^2\|Az\|.$$

If $t > 0$,

$$0 \leq -2(b - A\bar{x})^T Az + t\|Az\|_2^2,$$

and

$$2(b - A\bar{x})^T Az \leq t\|Az\|_2^2.$$

If $t < 0$,

$$0 \leq 2(b - A\bar{x})^T Az + |t| \|Az\|_2^2.$$

As $t \longrightarrow 0^+$ *or* $t \longrightarrow 0^-$, *we have* $2(b - A\bar{x})^T Az \leq 0$ *and* $0 \leq 2(b - A\bar{x})^T Az$, *so*

$$(b - A\bar{x})^T Az = 0$$

for all $z \in R^n$. *Thus,*

$$(b - A\bar{x})^T Az = (Az)^T (b - A\bar{x}) = z^T A^T (b - A\bar{x}) = z^T (A^T b - A^T A\bar{x}) = 0$$

for all $z \in R^n$. *Choose* $z = (A^T b - A^T A\bar{x})$, *so* $\|A^T b - A^T Ax\|_2^2 = 0$ *and* $A^T Ax = A^T b$. *For part (2), if* x *is the unique solution to* $A^T Ax = A^T b$, *then* $A^T A$ *is nonsingular so* A *must have full rank . If* A *has full rank,*$A^T A$ *is nonsingular , and* $A^T Ax = A^T b$ *has a unique solution.*

3.3 solving overdetermined least-squares problems

To solve overdetermined least squares problem we use the svd. There is a reduced SVD, and it has the form

$$A^{m\times n} = U^{m\times n} \Sigma^{n\times n} (V^{n\times n})^T.$$

In MATLAB,we use the command $[USV] = svd(A, 0)$.*The reduced SVD is also a powerful tool for computing full-rank least-squares solutions,* $m \geq n$. *The following manipulations show how to use it.*
Apply the reduced SVD to A *and obtain* $A = U\Sigma V^T$, *where* $U \in R^{m\times n}$ *has orthonormal columns, and* $V \in R^{n\times n}$ *is orthogonal, and* $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_n) \in R^{n\times n}$, $\sigma_i > 0$,$1 \leq i \leq n$. *Form the normal equations*

$$A^T A = (U\Sigma V^T)^T U\Sigma V^T$$

and so

$$A^T A = (V\Sigma)\Sigma V^T.$$

Now, $A^T b = (V\Sigma)U^T b$, *and so the normal equations become*

$$(V\Sigma)\Sigma V^T x = (V\Sigma)U^T b$$

Since V *is orthogonal and is a diagonal matrix with positive entries,*V *is invertible, and after multiplying the previous equation by* $(V\Sigma)^{-1}$ *we have*

$$\Sigma V^T x = U^T b.$$

First solve $y = U^T b$, *followed by* $V^T x = y$. *Since is a diagonal matrix, the solution to* $y = U^T b$ *is simple. Let*

$$U^T b = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix}.$$

Then,

$$\begin{bmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & 0 & \\ & 0 & \ddots & \\ 0 & & & \sigma_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix},$$

and $y_i = \frac{c_i}{\sigma_i}$,$1 \leq i \leq n$. *The solution to* $V^T x = y$ *is* $x = Vy$.

As an application of using SVD to solve least-square problems, we see the following example:

Example 3.3.1 *The velocity of an enzymatic reaction with Michaelis-Menton kinetics is given by*

$$v(s) = \frac{\alpha s}{1 + \beta s} \tag{3.1}$$

Find the Michaelis-Menton equation which best fits the data: Inverting Equation 3.1

s	1	4	6	16
v	4	10	12	16

x	1.0000	0.2500	0.1667	0.0625
y	0.2500	0.1000	0.0833	0.0625

gives the Lineweaver-Burke equation:

$$\frac{1}{v} = \frac{1}{\alpha}\frac{1}{s} + \frac{\beta}{\alpha} \tag{3.2}$$

Perform the following change of variable: $y = \frac{1}{v}$ *and* $x = \frac{1}{s}$. *Let* $m = \frac{1}{\alpha}$ *and* $b = \frac{\beta}{\alpha}$. *Equation 3.2 then becomes*

$$y = mx + b$$

Recompute the table to reflect the change of variables. Find the least-squares fit for $y = mx + b$ *by solving the following* 4×2 *set of equations*

$$\begin{aligned} 1.0000m + b &= 0.2500 \\ 0.2500m + b &= 0.1000 \\ 0.1667m + b &= 0.0833 \\ 0.0625m + b &= 0.0625 \end{aligned}$$

that correspond to the matrix equation

$$\begin{bmatrix} 1.0000 & 1 \\ 0.2500 & 1 \\ 0.1667 & 1 \\ 0.0625 & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} 0.2500 \\ 0.1000 \\ 0.0833 \\ 0.0625 \end{bmatrix}$$

The following MATLAB code solves for m *and* b *and then computes* α, β.

```
>> A = [1.0000 1.0000;0.2500 1.0000;0.1667 1.0000;0.0625 1.0000];
>> b = [0.2500 0.1000 0.0833 0.0625]?;
>> x = svdlstsq(A,b);
```

```
>> m = x(1);
>> b = x(2);
>> alpha = 1/m

alpha =

     4.9996

>> beta = alpha*b
beta =

    0.2499
```

The least-squares approximation is (Figure 3.1)

$$v(s) = \frac{4.9996s}{1 + 0.2499s}$$

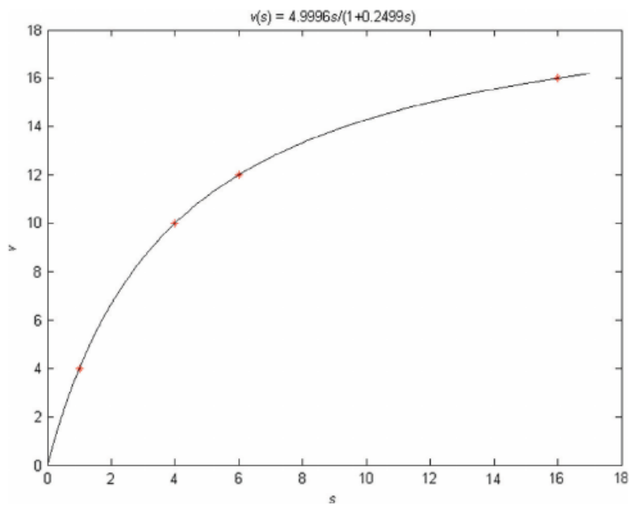

Figure 3.1: Velocity of an enzymatic reaction.

Example 3.3.2 *let*

$$A = \begin{bmatrix} 1 & -1 & 3 \\ 8 & 8 & 1 \\ 4 & 6 & -12 \\ 6 & -9 & 0 \\ 3 & 4 & 4 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 3 \\ -1 \\ 6 \\ 15 \end{bmatrix}$$

Find the solution using the SVD method

solution

$A = U\Sigma V^T$

$$A = \begin{bmatrix} 0.1268 & -0.1940 & -0.1338 & -0.9429 & -0.1983 \\ -0.5237 & -0.5756 & -0.3737 & 0.1987 & -0.4641 \\ -0.7830 & 0.5330 & -0.0888 & -0.2424 & 0.1902 \\ 0.2869 & 0.2723 & -0.9078 & 0.0887 & 0.1075 \\ -0.1195 & -0.5223 & -0.1024 & -0.0697 & 0.8352 \end{bmatrix} \begin{bmatrix} 16.0248 & 0 & 0 \\ 0 & 11.525 & 0 \\ 0 & 0 & 10.2160 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -0.3639 & -0.2255 & -0.9037 \\ -0.7535 & -0.4991 & 0.4280 \\ 0.5476 & -0.8367 & -0.0117 \end{bmatrix}$$

$v^T x = y, y_i = \frac{c_i}{\sigma_i}, c = U^T b.$

$$c = \begin{bmatrix} -0.7325 \\ -8.6542 \\ -8.1488 \\ -0.6179 \\ 11.3923 \end{bmatrix}, y = \begin{bmatrix} -0.0457 \\ -0.7509 \\ -0.7976 \end{bmatrix}$$

$$\begin{bmatrix} -0.3639 & -0.7535 & 0.5476 \\ -0.2255 & -0.4991 & -0.8367 \\ -0.9037 & 0.4280 & -0.0117 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -0.0457 \\ -0.7509 \\ -0.7976 \end{bmatrix}$$

$x_1 = 0.9068, x_2 = 0.0678, x_3 = 0.6125$

Example 3.3.3 *let*

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 4 \\ 3 & 8 \\ 2 & 9 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 3 \\ 5 \\ 8 \end{bmatrix}$$

Find the solution using the SVD method

solution

$A = U\Sigma V^T$

$$A = \begin{bmatrix} -0.2314 & -0.0471 & -0.6326 & -0.7376 \\ -0.3232 & -0.6113 & -0.4730 & 0.5461 \\ -0.6245 & -0.3998 & 0.6025 & -0.2953 \\ -0.6723 & 0.6814 & -0.1145 & 0.2656 \end{bmatrix} \begin{bmatrix} 13.6622 & 0 \\ 0 & 1.1596 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -0.2998 & -0.9540 \\ -0.9540 & 0.2998 \end{bmatrix}$$

$v^T x = y, y_i = \frac{c_i}{\sigma_i}, c = U^T b.$

$$c = \begin{bmatrix} -9.7020 \\ 1.5712 \\ 0.0451 \\ 1.5493 \end{bmatrix}, y = \begin{bmatrix} -0.7101 \\ 1.3549 \end{bmatrix}$$

$$\begin{bmatrix} -0.2998 & -0.9540 \\ -0.9540 & 0.2998 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -0.7101 \\ 1.3549 \end{bmatrix}$$

$x_1 = -1.0797, x_2 = 1.0837.$

Chapter 4

Conclusion

We have presented first two iterative methods The Gauss-Jacobi method and Gauss-Seidel method for solving a system of algebraic equations.

We apply it to solve systems of algebraic equations with square coefficient matrix which is diagonally dominant. The solution by these two iterative methods converges to the exact solution.

We find this two methods are simple and obtaining an approximate solution is converging to the exact solution for a system with a square coefficient matrix. If the system has not a square coefficient matrix (an overdetermined system), then these two methods will fail to solve this system.

The singular value decomposition (SVD)is used to solve an overdetermined system which is a power full tool for the solution. Also the SVD method is a power full tool to solve the least square problem.

Bibliography

[1] *B.N. Datta, Numerical Linear Algebra and Applications, second ed., SIAM, Philadelphia, 2010.*

[2] *.G.H. Golub, C.F.Van Loan, Matrix Computations, fourth ed.,The Johns Hopkins University Press, Baltimore, 2013.*

[3] *G.H. Golub, C. Reinsch, Singular value decomposition and least squares solutions, Numer. Math. 14 (1970) 403420.*

[4] *G. Strang, Introduction to Linear Algebra, fourth ed., Wellesley- Cambridge Press, Wellesley,MA, 2009*

[5] *J.W. Demmel, Applied Numerical Linear Algebra, SIAM, Philadelphia, 1997.*

[6] *W.ford Numerical Linear Algebra with Appalications,ELSEVIER,UK,2015.*

[7] *Y. Saad, Numerical Methods for Large Eigenvalue Problems, Revised ed., SIAM, Philadelphia, 2011.*