1 Introduction

This assigment was done using C++ as the programming language, in the CLion IDE. Furthermore, the code from Numerical Recipes as well as some of the code provided by Ole Wennerberg Nielsen is used in the assigment - This will be marked using comments in the source code.

2 Exercise 1 (20 points)

Consider the linear equation $\mathbf{Ax} = \mathbf{b}$ where the 10×6 dimensional coeffcient matrix A and the right hand side b are given in *Ex1A.dat* and *Ex1b.dat*.

i) (10 points) Find the Singular Value Decomposition (SVD) $\mathbf{A} = \mathbf{UWV^T}$. State the diagonal elements in $\mathbf{W}$. Submit the used code.

In figure 2, matrix A and its SVD elements are seen.

```
A = U*W*T(V):
A       Matrix 10x6:
    -0.2244865    -0.8671804     0.5538675     0.4667952    -0.5046398     0.2918244
     0.4702761     0.5446024      0.783741    -0.8283025     0.8424086      1.886011
     -0.695119   -0.05810601     0.8960382    -0.6301382     0.1990554     0.8703464
    -0.6719778    -0.3098617    -0.9502969    -0.9089616     0.5931869    -0.1991461
    0.03471944    -0.2753386    0.07279159    0.07263496     -0.740794     -0.350045
    -0.4081582     0.5189241     0.6058397     0.7638458     0.1215657    -0.4633721
     0.9743217     0.8220987     0.5921483      0.279157     0.6694407      1.120376
    -0.1197952     0.6961101   -0.09070482    -0.5296146    -0.4478174    -0.7390419
   0.005971777     -0.140408     -0.620653   -0.04100433     0.9712523     0.3308694
    -0.8650693     0.3065554     0.8863607     0.7120464     0.5430889   -0.03167097

U       Matrix 10x6:
   -0.03025452     0.3031402     0.2133251    -0.5367182     0.3083527    0.06697301
     0.7328128    -0.1567385    0.02954747    -0.1373583    -0.2073977     0.2305694
     0.3080013    0.05974315      0.567434    -0.2440015    -0.3136331    0.07401154
    -0.1035073     -0.582208     0.4320083     0.1798271   0.005015352    -0.1660911
    -0.1864684     0.1230006   -0.07733242    -0.2979609     -0.123999    0.09466832
   -0.04215443       0.44018    0.09944317     0.4239303   -0.05502176     0.6547462
     0.5161958     0.1373687    -0.4449636     0.1905457    0.09044809     -0.303095
    -0.1702634   -0.05338541   -0.09762262     0.2012482    -0.7107251    0.02982495
     0.1194595    -0.3253896    0.09108462     0.2868987     0.4737023     0.3970189
      0.101122     0.4522362     0.4618641      0.416623    0.08605733     -0.470376

W       Matrix 6x6:
      3.287839             0             0             0             0             0
             0      2.332613             0             0             0             0
             0             0      1.819115             0             0             0
             0             0             0      1.673514             0             0
             0             0             0             0      1.448961             0
             0             0             0             0             0  9.776714e-16

T(V)    Matrix 6x6:
      0.198969     0.2399872     0.3739251    -0.1415754      0.413119     0.7564962
   -0.09447506     0.1214687     0.6929888     0.6473554    -0.2617886   -0.09210855
     -0.870118    -0.3121299      0.215518    -0.1935954     0.2424076    0.05273592
    -0.1648424     0.6186285    -0.1634359     0.2313751     0.6141883    -0.3642152
      0.115731    -0.5649585    -0.2895974     0.6220158     0.4010038     0.1893513
    -0.3921912     0.3582556    -0.4721446     0.2880453    -0.4049773     0.4979375
```

Figure 1: A and its' SVD elements

ii) (5 points) Use SVD to compute the solution **x** to **Ax** = **b**. State the solution **x**. Submit the used code.

Below, two solutions **x** to **Ax** = **b** are shown for the minimization of **Ax - b**. For the SVD solution to be remotely usable, the residual error $\epsilon_{\text{residual}}$ should be **significantly** smaller than the random fitting $\sqrt{\frac{m-n}{m}}$. The aforementioned is not the case in either solutions, in fact $\epsilon_{\text{residual}}$ is larger than $\sqrt{\frac{m-n}{m}}$ in both solutions.

It is noticed that the results of **A**·**x** differ between the two solutions due to roundoff errors. Thus the comparision between $\epsilon_{\text{residual}}$ and $\sqrt{\frac{m-n}{m}}$ differs by a magnitude of 10^{-3}, which is insignificant given that both solutions are still unviable for the task of minimizing **Ax - b**.

```
x       Vector 6D:
   9.814596e+13  -8.965358e+13   1.181543e+14  -7.208342e+13   1.013457e+14   -1.24609e+14


A*x     Vector 10D:
      -0.109375       -0.65625       -0.09375     -0.1328125         0.1875        0.40625      -0.328125       0.609375
     -0.5859375      0.4604492


b       Vector 10D:
     -0.5415778     -0.6439477      0.1423662     -0.8561312      0.5782175     -0.3400578      -0.766929      0.6734834
     0.02603491      0.6539704

Random fitting 0.6324555
x Residual 0.7850958
x Compared to random fitting 0.1526403


x2      Vector 6D:
   9.814596e+13  -8.965358e+13   1.181543e+14  -7.208342e+13   1.013457e+14   -1.24609e+14


A*x2    Vector 10D:
      -0.140625       -0.65625         -0.125     -0.1210938       0.203125      0.3671875       -0.34375          0.625
       -0.59375      0.4130859


b       Vector 10D:
     -0.5415778     -0.6439477      0.1423662     -0.8561312      0.5782175     -0.3400578      -0.766929      0.6734834
     0.02603491      0.6539704

Random fitting 0.6324555
x2 Residual 0.7761954
x2 Compared to random fitting 0.1437398

Ax - Ax2        Vector 10D:
        0.03125              0        0.03125    -0.01171875      -0.015625      0.0390625       0.015625      -0.015625
      0.0078125     0.04736328

Difference in comparison: 0.008900452
```

Figure 2: A and its' elements

3 Exercise 2 (20 points)

In this exercise, the task is to derive the time of collision between two objects. Object A is as sphere with radius $r = 1$ residing at the origin. Object B is a shape changing ellipsoidal object, which at time t is centered at $(x_e(t); y_e(t); z_e(t))$. We can describe the surface of object B (hereafter called S_B) as the set of points $(x; y; z)$ satisfying

$$a(t)\left[x - x_e(t)\right]^2 + b(t)\left[y - y_e(t)\right]^2 + c(t)\left[z - z_e(t)\right]^2 = 1$$

where $a(t), b(t), c(t)$ are all positive. The point on object B that is closest to the center of object A (the origin) is then clearly given as

$$\min_{x,y,z\ \epsilon\ S_B}\ \left\{x^2 + y^2 + x^2\right\}$$

which is a constrained optimization problem where the solution must satisfy the first order conditions

$$\begin{aligned}
x - \lambda a(t)\left[x - x_e(t)\right] &= 0 \\
y - \lambda b(t)\left[y - y_e(t)\right] &= 0 \\
z - \lambda c(t)\left[z - z_e(t)\right] &= 0 \\
a(t)\left[x - x_e(t)\right]^2 + b(t)\left[y - y_e(t)\right]^2 + c(t)\left[z - z_e(t)\right]^2 - 1 &= 0
\end{aligned} \tag{1}$$

where λ is a socalled Lagrange Multiplier. We have now given

$$\begin{aligned}
a(t) &= 1.1(1 + t^2) \\
b(t) &= 2.1(1 + 2t^2) \\
c(t) &= 0.8(1 + 3t^2) \\
x_e(t) &= 2.3\exp(-t) \\
y_e(t) &= 4.8\exp(-t) \\
z_e(t) &= 0.7\exp(-t)
\end{aligned}$$

i) (6 points) At time $t = 1.2$, find a solution $(x; y; z; \lambda)$ satisfying the first order conditions in Eq.(1). Use Newtons method starting with $(x; y; z; \lambda) = (0; 0; 0; 0)$ and with a required tolerance of $10^{-}8$. State your result for $x; y; z$ and compute and state the distance to the origin $d = \sqrt{x^2 + y^2 + z^2}$. Submit the used code. As a check that you programmed everything correct, d should be approximately $d \approx 1.22$. As $d > 1$, the two objects are not penetrating each other.

```
2.i
Did newt find usable roots? true
Roots for t = 1.2       Vector 4D:
      0.4041804       1.170462      0.1453793     -0.5218518
d = 1.246787
```

Figure 3: Roots for $t = 1.2$

ii) (6 points) At time $t = 1.4$, find a solution $(x; y; z; \lambda)$ satisfying the first order conditions in Eq.(1). Use Newtons method starting with $(x; y; z; \lambda) = (0; 0; 0; 0)$ and with a required tolerance of $10^{-}8$. State your result for $x; y; z$ and compute and state the distance to the origin $d = \sqrt{x^2 + y^2 + z^2}$. Submit the used code. As we now have $d < 1$, the two objects penetrate.

```
2.i
Did newt find usable roots? true
Roots for t = 1.2       Vector 4D:
      0.4041804       1.170462      0.1453793     -0.5218518
d = 1.246787
```

Figure 4: Roots for $t = 1.4$

4 Exercise 5a (15 points) (Only Diploma Engineering Students)

Consider the integral

$$\int_a^b \cos(x^3)\exp(-x)dx$$

i) (5 points) For $N = 2$, state an analytical expression for the approximation of the integral as obtained by the Trapezoidal method.

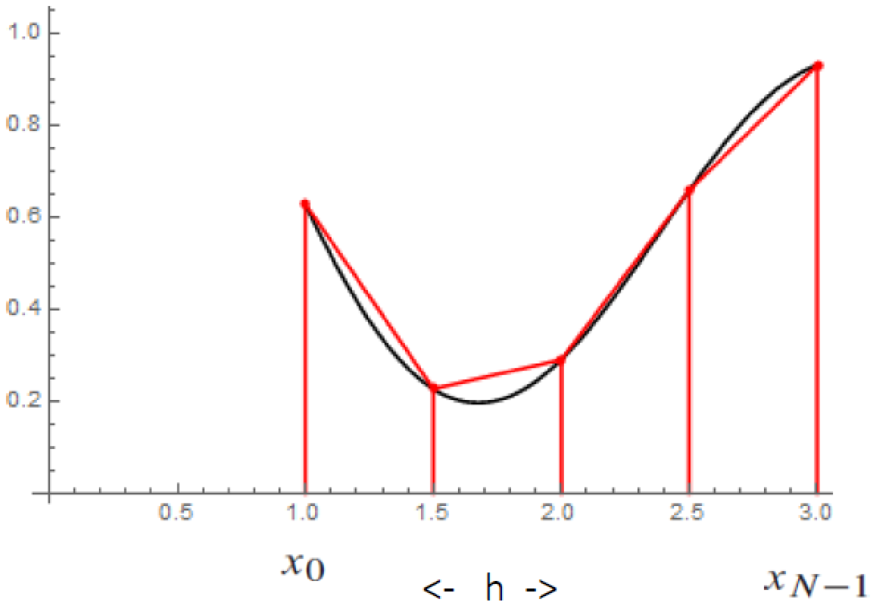

Figure 5: Visualisation of the trapezoidal method from the slides of lecture 8 in Numerical Methods.

The trapezoidal method is defined as being

$$\int_a^b f(x)dx = h \cdot \left[\frac{1}{2}f_0 + f_1 + f_2 + \cdots + f_{N-2} + \frac{1}{2}f_{N-1}\right]$$

where

$$h = \frac{b-a}{N}$$

Given $N = 2$, the trapezoidal method will look at follows

$$\int_a^b f(x)dx = h \cdot \left[\frac{1}{2}f_0 + f_1 + \frac{1}{2}f_{N-1}\right]$$

where

$$h = \frac{b-a}{2}$$

We now consider the case $a = 1, b = 3$. We wish to approximate the integral using the Trapezoidal method. For this, we as usual split the integration interval into N equidistant subintervals. You may without proof assume that the order is 2 as expected.

ii) (10 points) With $N = 2, 4, 8, \dots$ use the Trapezoidal method to approximate the integral. Terminate the subdivisions when you reach a proven accuracy of better than 10^{-5}. State your results for each N, preferably in a table similar to those used during the course. Submit the used code.

Below is a table showing the results of using the trapezoidal method with $N = 2, 4, 8, \dots$ to the point of the accuracy reaching more than 10^{-5}. It is seen that the goal is reached after six iterations.

i	N - 1	A(hi)	A(hi - 1) - A(hi)	Rich - alpha^k	Rich-error	F-Calculations
1	1	0.1842214	*	4	*	2
2	3	0.1150375	0.06918388	4	0.02306129	4
3	7	0.04245041	0.07258708	4	0.02419569	8
4	15	0.03846676	0.003983652	4	0.001327884	16
5	31	0.03772984	0.0007369203	4	0.0002456401	32
6	63	0.03766943	6.040678e-05	4	2.013559e-05	64
7	127	0.0376591	1.033829e-05	4	3.446095e-06	128

Figure 6: Approximated areas using trapezoidal method with up until an accuracy of better than 10^{-5}.