Patrick Ørsted Povey Andersen

Numeriske Metoder

Eksamen 2025

Table of Contents

Patrick Ørsted Povey Andersen

Exercise 1

i)

Find the Singular Value Decomposition and State the diagonal elements in W.

Solution

The code reads the data file NUM S25_Ex1A.dat for the matrix A and uses numerical recipes SVD method to print svd.w.

```
----------------- Problem i -----------------------

Diagonal elements of W:

11.951932409770585      5.397266515466192      4.719596639028399      3.9539065956033674      3.623090993000244      3.368300112475189      3.091666468243567      4.729403607652192e-15
```

ii)

There is a single element in W that is basically zero. Use the information from the SVD matrices to state a unit vector in the null space of A.

Solution

The code uses numerical recipes SVD method svd.nullspace to find the unit vector in the null space of A. The threshold is there to find the element near zero.

```
void problemII(MatDoub A) {
    SVD svd(a: A);

    util::print(mat: svd.nullspace(thresh: 1e-10));
}
```

```
----------------- Problem ii -----------------------

-0.3041495323362373
0.6995439243733442
4.862063999926544e-17
3.9381502970548994e-16
-0.42580934527073155
1.7340582620100244e-17
-9.82417271311071e-17
0.48663925173797906
```

iii)

Use the Singular Value Decomposition to compute the solution x to Ax = b. State the solution x.

Solution

Using SVD to find the best-fit least squares solution and returns the vector x

```
----------------- Problem iii -----------------------

Best fit solution parameters x:

-47.24836361643928      0.36884252263498496      79.65376424734222      -4.711970963467443      40.404271525143045      69.74266521165167      -42.73505350336738      5.293299197937798
```

Patrick Ørsted Povey Andersen

iv)

State an estimate of the accuracy on the solution x. State an explanation of how you computed the accuracy.

Solution

The relative residual error was computed in the residualError() function. This measures how closely the solution satisfies the original system.

The randomFittingError() gives an estimate of the expected residual error if the data were purely random.

The sigErrorEstimate() function estimates the standard deviation sigma on each component of the solution vector x, based on the values in W.

```
----------------- Problem iV -----------------------


Relative residual error epsilon_residual = 5.42797e-06

Random fit Error= 0.912871


sigma   Vector 8D:
0.0846856316939977      0.16099885651146364     0.2770351882338667      0.27503010240108106     0.2266191205749653      0.2597660651878675      0.25037451202513783     0.22830996340629964
```

Patrick Ørsted Povey Andersen

Exercise 2

i)

With $x_0 = -0.7,\ x_1 = 1.2,\ x_2 = 2.3,\ x_3 = -4.1$ state (with at least 7 digits) the values of the left-hand side of the four equations. (HINT: you should get something around (2.36, 6.03, 49.32, −14.12).

Solution

```
-------------------- Problem I ------------------------


2.360427776065722        6.033039085967225        49.32299999999999        -14.118275390737853
```

ii)

Search for a solution to the equations by performing 7 iterations with the globally convergent Newton method with initial guess (x0, x1, x2, x3) = (0,0,0,0).
State for each iteration the values of x0, x1, x2, x3 and for each iteration whether backtracking was applied, and if so, what the value of λ was. Submit the used code.

Solution

Looking at the table below you can see backtracking is happening at k = 3 because lambda value is lower than 1 and at k = 4 the convergence and error gets a spike as a result.

```
--------------------Problem II & III------------------------

k         x0          x1          x2          x3        dx_k      lambda  convergence      error
1  -1.400000    1.275000   -1.733333    0.900000    0.000000         nan          nan        nan
2  -2.529812    0.206707   -1.659476    0.892282   -1.129812         nan          nan        nan
3  -2.374943    0.382163   -1.699814    0.772997    0.154870    0.111687     0.109669   0.007745
4  -1.274613    1.222814   -1.368963    0.577609    1.100330         nan    20.347122  42.017974
5  -0.971073    1.118543   -1.118002    0.724769    0.303540         nan     0.209767   0.039362
6  -0.966679    1.134405   -1.102446    0.741389    0.004394         nan     0.149706   0.000118
7  -0.966346    1.134616   -1.102211    0.741495    0.000332         nan     0.596128   0.000000
```

iii)

State an estimate the accuracy after 7 iterations. The estimate must be based on the data obtained from the 7 iterations and must be stated with a clear argument of how you computed it. Without such an argument, there is no points for the answer.

Solution

The accuracy after 7 Newton iterations was estimated using $\in_k \approx C_k * \|dx_k\|^2$

- Where $dx_k = x_k - x_{k-1}$ is the difference
- Where C_k is the estimated convergence

These values are computed in printRoots()


```
--------------------Problem II & III------------------------

k         x0          x1          x2          x3        dx_k      lambda  convergence      error
1  -1.400000    1.275000   -1.733333    0.900000    0.000000         nan          nan        nan
2  -2.529812    0.206707   -1.659476    0.892282   -1.129812         nan          nan        nan
3  -2.374943    0.382163   -1.699814    0.772997    0.154870    0.111687     0.109669   0.007745
4  -1.274613    1.222814   -1.368963    0.577609    1.100330         nan    20.347122  42.017974
5  -0.971073    1.118543   -1.118002    0.724769    0.303540         nan     0.209767   0.039362
6  -0.966679    1.134405   -1.102446    0.741389    0.004394         nan     0.149706   0.000118
7  -0.966346    1.134616   -1.102211    0.741495    0.000332         nan     0.596128   0.000000
```

Exercise 3

i)

- Rewrite Eq. (1) into a set of two first order ODE's. State the two first order ODE's.

$$x''(t) = a_{max}\left[1 - \left[\frac{x'(t)}{v_{des}}\right]^4 - \left(\frac{D_0 + \text{Max}\left\{0, x'(t) * T_{react} + \frac{x'(t)\left(x'(t) - X'_F(t)\right)}{2 * a_{com}}\right\}}{X_F(t) - x(t)}\right)^2\right]$$

Solution

To convert the equation into a set of two first order ODE’s I introduce a new variable:

$$x_1(\mathrm{t}) = \mathrm{x_0}'(\mathrm{t})$$

This means that: $\mathrm{x_0}''(\mathrm{t}) = x_1'(\mathrm{t})$

Substituting this into the equation we get the two first order ODE’s:

1. $x_0'(t) = x_1(t)$
2. $x_1'(\mathrm{t}) = a_{max}\left[1 - \left[\frac{x'(t)}{v_{des}}\right]^4 - \left(\frac{D_0 + \text{Max}\left\{0, x'(t) * T_{react} + \frac{x'(t)\left(x'(t) - X'_F(t)\right)}{2 * a_{com}}\right\}}{X_F(t) - x(t)}\right)^2\right]$

ii)

State with at least 7 digits the value of x’’(t0). Submit the used code. HINT: You should get X´´(t0) ≃ −1.20.

Solution

```
---------------------Problem II-------------------------
      15          -1.2035371846808856
```

iii)

Use the Midpoint method with N = 80; h= 0.25 to generate a solution for x(t); −10 ≤t≤ 10. State the value of x(t) at t = 10. State plots of x(t), x’(t) and XF(t)−x(t). Submit the used code.

Solution

If I knew how to save the data in a file and then maybe use python or MATLAB to read and process it I would have continued with the problem. In the code I also print out the 80 steps.

```
--------------------Problem III-------------------------
284.3842345949725         10.092177946269342
```

Patrick Ørsted Povey Andersen

Exercise 4

i)

With N−1 = 2^k ; k= 1,...,20 use the Simpson Method method to approximate the integral. State the results in a table similar to those used during the course. Submit the used code.

Solution

```
--------------------- Problem I -------------------------
Using Simpson's rule

  i       A(h_i) A(h_{i-1}) - A(h_i)      alpha^k   Rich error   Order est.    f comps
  1     3.624376                 nan          nan          nan          nan          2
  2    18.432966          -14.808590          nan     4.936197          nan          4
  3    54.698586          -36.265621     0.408337    12.088540    -1.292168          8
  4    86.718877          -32.020290     1.132583    10.673430     0.179616         16
  5   100.111967          -13.393091     2.390807     4.464364     1.257497         32
  6   104.354005           -4.242037     3.157231     1.414012     1.658660         64
  7   105.706640           -1.352635     3.136128     0.450878     1.648984        128
  8   106.159909           -0.453269     2.984177     0.151090     1.577333        256
  9   106.316225           -0.156316     2.899703     0.052105     1.535905        512
 10   106.370852           -0.054628     2.861479     0.018209     1.516761       1024
 11   106.390059           -0.019207     2.844194     0.006402     1.508020       2048
 12   106.396831           -0.006772     2.836107     0.002257     1.503912       4096
 13   106.399222           -0.002391     2.832215     0.000797     1.501931       8192
 14   106.400067           -0.000845     2.830308     0.000282     1.500959      16384
 15   106.400366           -0.000299     2.829364     0.000100     1.500478      32768
 16   106.400471           -0.000106     2.828895     0.000035     1.500238      65536
 17   106.400509           -0.000037     2.828661     0.000012     1.500119     131072
 18   106.400522           -0.000013     2.828544     0.000004     1.500060     262144
 19   106.400526           -0.000005     2.828486     0.000002     1.500030     524288
 20   106.400528           -0.000002     2.828454     0.000001     1.500014    1048576
```

ii)

Use Richardson extrapolation to estimate the order at N−1 = 2^20. State the result. Submit the used code.

Solution

Can also be seen in the table above.

```
-------------------Problem II----------------------

Order estimates for Problem II:
 0: nan
 1: nan
 2: -1.292168
 3: 0.179616
 4: 1.257497
 5: 1.658660
 6: 1.648984
 7: 1.577333
 8: 1.535905
 9: 1.516761
 10: 1.508020
 11: 1.503912
 12: 1.501931
 13: 1.500959
 14: 1.500478
 15: 1.500238
 16: 1.500119
 17: 1.500060
 18: 1.500030
 19: 1.500014
```

iii)

The estimated order is different than the expected order. State an explanation for the difference and your guess on the exact order that you have estimated.

Solution

The function has square roots at the ends, so its slope changes too fast at x=0 and x=2. Simpsons method needs the function slopes to not change this way, so the error wont go down as fast. This would be why the order is around 1.5 instead of 4.

iv)

State the estimated accuracy on the result at N−1 = 2^20 using the estimated order. State clearly how you compute the accuracy estimate.

Solution

The estimated accuracy is computed using Richardson extrapolation, equation 7 in Richardson.pdf from numerical methods course in itslearning resources.

```
--------------------Problem IV------------------------

Estimated accuracy at N = 2^20: 9.018488e-07
```

v)

State what other method could have been used to achieve high accuracy with likely much fewer f(x) computations?

Solution

The midpoint method could be better because it doesn’t use the values at the edges like Simpson at x=0 and x=2. Midpoint evaluates the function inside the interval.

Patrick Ørsted Povey Andersen

Exercise 5b

i)

Consider N = 2. State an analytical expression of the semi discrete form for this problem. State also the value of $\frac{du_1(t)}{dt}$ for t = 0

Solution

I use the semidiscrete formula for $\frac{du_j(t)}{dt}$ explained in uge14pres.pdf slide 5.

$$\frac{du_j(t)}{dt} = \frac{4}{\frac{1}{4}}\left(u_{j-1}(t) - 2 * u_j(t) + u_{j+1}(t)\right) + f_j(t)$$

$$u_j(0) = g(x_j)$$

$$u_0(t) = a(t)$$

$$u_n(t) = b(t)$$

I use j = 1, which is the only inner point when N=2 and find $\frac{du_1(t)}{dt}$ at t = 0

$$\frac{du_1(t)}{dt} = \frac{4}{\frac{1}{4}}\left(u_{1-1}(t) - 2 * u_1(t) + u_{1+1}(t)\right) + f_j(t)$$

$$\frac{du_1(t)}{dt} = \frac{4}{\frac{1}{4}}\left(u_0(t) - 2 * u_1(t) + u_2(t)\right) + f_j(t)$$

$$f_j(t) = \sin(\pi * x) * \exp(-t)$$

$$\frac{du_1(t)}{dt} = \frac{4}{\frac{1}{4}}(0^2 - 2 * 0.5^2 + 1^2) + \sin(\pi * 0.5) * \exp(-0)$$

$$8 + \sin(pi * 0.5) * \exp(-0)$$

$$8 + 1 * 1 = 9$$