Assignment 3

October 3, 2022

1 Assignment 3

1.0.1 Filip Sjövall 19950308

1.1 Task 1

The task is to find a polynomial representation of some sample data using the least squares method. The residual $r = Ax - b$ will be minimized if it is orthogonal to the column space of A / range(A),m i.e. $A^T r = 0$. This gives us the normal form $A^T Ax = A^T b$ for which we will seek a solution in this task.

The input data is modelled by

$$y(x) = asin(x) + bcos(x) + csin(2x) + dcos(2x) \tag{1}$$

where a, b, c, d are the unknowns. Given the sample data (x_i, y_i) the following system of equations models the solution

$$Ax = y, \tag{2}$$

where $ x = (a,b,c,d)^T $, y the sampled data y_i and

$$A = \begin{bmatrix} sin(x_1) & cos(x_1) & sin(2x_1) & cos(2x_1) \\ sin(x_2) & cos(x_2) & sin(2x_2) & cos(2x_2) \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ sin(x_n) & cos(x_n) & sin(2x_n) & cos(2x_n) \end{bmatrix}. \tag{3}$$

The idea is to solve the system (**??**) not using A^{-1} which is expensive but instead using QR or SVD. To get square matrices we can multiply from the left to get

$$A^T Ax = A^T y = b, \tag{4}$$

so then $QR(A^T A)$ gives us

$$QRx = b, \tag{5}$$
$$Rx = Q^T b, \tag{6}$$

which can be solved using backwards substitution. Similarly for the SVD(A^T A) we get

$$U\Sigma V^T x = b, \tag{7}$$
$$x = V\Sigma^{-1}U^T b \tag{8}$$

which is also straight forward to solve (Σ is diagonal so the inverse is diag($1/\sigma\_i...$))$andtheremainingmatrixvectorproductsarealsostraightforward.Thecostofsolving$**??** using the above described methods are in both cases dominated by the factorization as the rest is matrix-vector multiplications or backwards substitution. Något om hur r vinkelrät mot range(A) och A^Tr=0

[214]:
```
from numpy import *
#from scipy import *
from numpy import linalg
#from scipy import linalg
from scipy import optimize
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter, MultipleLocator
import math
import sys, getopt, os, string
```

[215]:
```
# Open file and create vectors x and y
file = open('/home/filip/Documents/NumLinAlg/Assignment 3/signal.dat','r')
x0 = []
y = []
for row, line in enumerate(file, 1):
    fname = line.rstrip().split(',') #using rstrip to remove the \n
    x0.append(float64(fname[0]))
    y.append(float64(fname[1]))
```

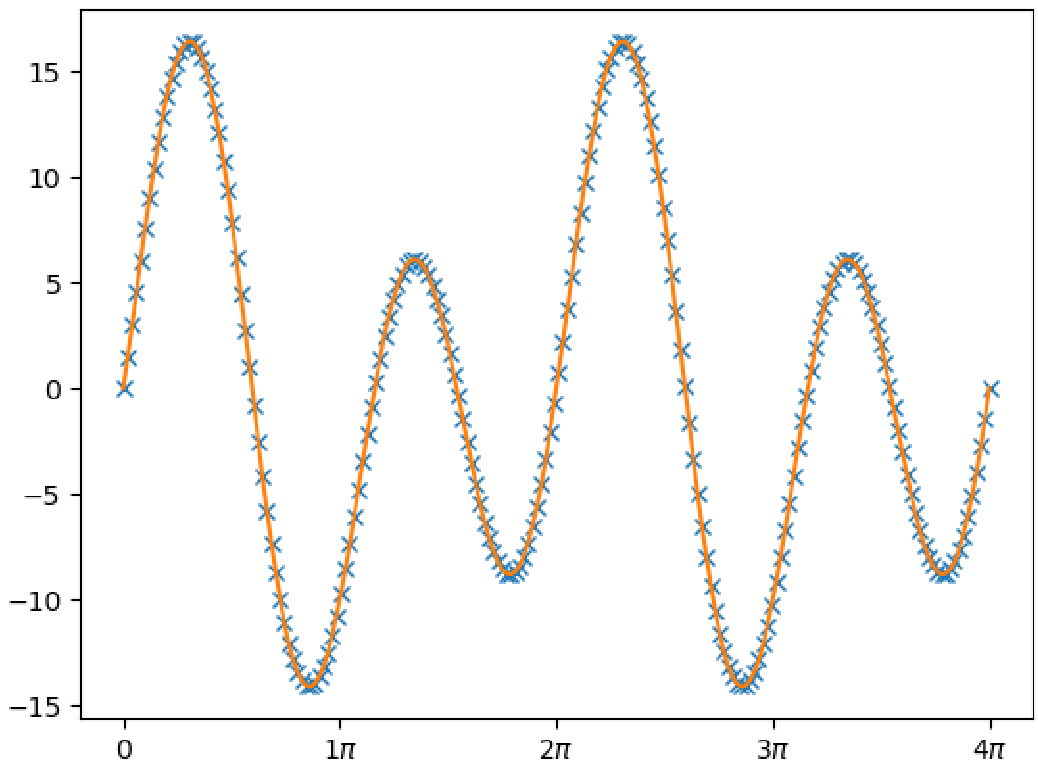
[244]:
```
# Create A for the linear system
A0 = zeros([len(x0),4])
for i, xi in enumerate(x0, 1 ):
    A0[int(i)-1,:] = [sin(xi),cos(xi),sin(2*xi),cos(2*xi)]
```

The system is solved both with QR factorization and with Singular value decomposition

[218]:
```
print(shape(A0))
A = A0.T@A0
b0 = A0.T@y
Q,R = linalg.qr(A)
```

```
b = Q@b0

#System solution, could probably find a closed form solution to loop over for␣
 ↪models with more parameters in x
x     = zeros([4,1])
x[3] =   b[3]/R[3,3]
x[2]  = (b[2] - R[2,3]*x[3] )/R[2,2]
x[1]  = (b[1] - R[1,2]*x[2] - R[1,3]*x[3] )/R[1,1]
x[0]  = (b[0] - R[0,1]*x[1] - R[0,2]*x[2] - R[0,3]*x[3] )/R[0,0]

# Visualize result
f,ax=plt.subplots(1)
ax.plot(array(x0),y,'x')
ax.plot(x0,A0@x)
ax.xaxis.set_major_formatter(FuncFormatter(
   lambda val,pos: '{:.0g}$\pi$'.format(val/np.pi) if val !=0 else '0'
))
ax.xaxis.set_major_locator(MultipleLocator(base=np.pi))

plt.show()

linalg.norm(A0@x-y,2)
```

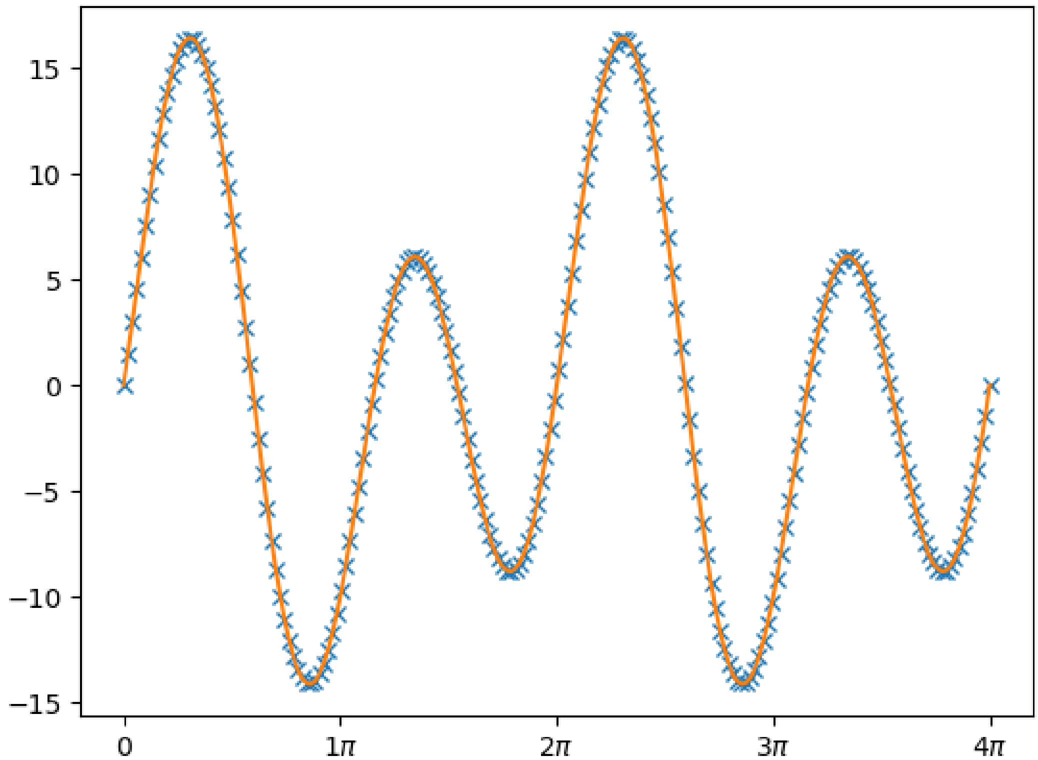
```
(200, 4)
```



[218]: 1778.7782869889143

[220]:
```
U,S,V = linalg.svd(A, full_matrices=True)


# Solve: S w = U b
b2 = U.T@b0
w  = zeros([4,1])
x = w
for i in range(0,4):
    w[i] = b2[i]/S[i]
x = V.T@w


# Visualize result
f,ax=plt.subplots(1)
ax.plot(array(x0),y,'x')
ax.plot(array(x0),A0@x)
ax.xaxis.set_major_formatter(FuncFormatter(
   lambda val,pos: '{:.0g}$\pi$'.format(val/np.pi) if val !=0 else '0'
))
ax.xaxis.set_major_locator(MultipleLocator(base=np.pi))
```

```
plt.show()

linalg.norm(A0@x-y,2)
```



[220]: 1778.7782869889138

2 Task 2

Condition number for solving the equation $Ax = b$. We consider perturbations in the solution δx caused by perturbations in the RHS δb i.e. $A(x + \delta x) = b + \delta$. Since $Ax = b$ we should have $A\delta x = \delta b$. Using SVD we get

$$U\Sigma V^T x = b \tag{9}$$
$$x = V\Sigma^{-1}U^T b. \tag{10}$$

if b is a left singular vector $b = u_j$ the RHS becomes $v_j \frac{1}{\sigma_j}$ so the norm becomes

$$||x|| = ||\frac{1}{\sigma_j} v_j|| = \frac{1}{\sigma_j}||v_j||. \tag{11}$$

Similarly if we use $\delta x = \delta b = \delta u_k = \varepsilon u_k$ we get

$$||\delta x|| = \frac{\varepsilon}{\sigma_k}||v_k||. \tag{12}$$

Now if we choose $b = u_1$ and $\delta b = \varepsilon u_n$ and we arrive at the qoutient

$$\frac{||\delta x||}{||x||} = \varepsilon\kappa(A), \tag{13}$$

so we have found a vector pair $(b, \delta b)$ for which the bound

$$\frac{||\delta x||}{||x||} \geq \kappa(A)\frac{||\delta b||}{||b||}, \tag{14}$$

is tight.

3 Task 3

A "worst case" vector pair is when the bound in (**??**) is tight. TThus we compare $x = A^{-1}b$ and $x + \delta x = A^{-1}(b + \delta b)$.

[240]:
```
from scipy import linalg as sl
H = sl.hilbert(50)
U,S,V = linalg.svd(copy(H))
b = U[:,0]
db= b + U[:,-1]
x_0 = sl.invhilbert(50)@b
x_1 = sl.invhilbert(50)@db
linalg.norm(x_1-x_0,2)
```

[240]: 1.3740438898204339e+73

4 Task 4

Sylvesters criterion. A symmetric positive definite matrix (SPD) A is defined

$$x^T A x > 0 \quad \forall x \neq \mathbf{0} \tag{15}$$

We consider the SPD matrix A of dimension nxn

$$A = \begin{bmatrix} a_1 1 & w^T \\ w & A_1 \end{bmatrix}, \tag{16}$$

where $a_1 1$ has dimension 1, w $n-1$ and A_1 is $(n-1)x(n-1)$. To show that a_{11} is positive if A is positive we consider the vector $x_1 = (\hat{x}, \mathbf{0})$, where $\mathbf{0}$ is the zero vector of dimension $n-1$ and $\hat{x} \neq 0$. Then, by the definition of an SPD matrix we have

$$x^T A x = \hat{x}^T A \hat{x} > 0. \tag{17}$$

In the same way we may consider $x_2 = (0, \hat{x}_2)$ where $\hat{x}_2 \neq \mathbf{0}$ and again use the definited of an SPD to show that A_1 is SPD.

5 Task 5

A strictly diagonally dominant matrix A has the property

$$a_{ii} = \sum_{j \neq i} a_{ij} \forall i. \tag{18}$$

To prove that A is nonsingular we consider

$$a_{ij} u_j = 0, \tag{19}$$

which should only have solutions for $u_j = 0$. (**??**) can be rewritten by subtracting the diagonal of A

$$a_{ii} u_i = -\sum_{j \neq i} a_{ij} u_j, \tag{20}$$

so

$$a_{ii} = -\sum_{j \neq i} a_{ij} \frac{u_j}{u_i}, \tag{21}$$

for which we may consider the norm

$$|a_{ii}| = |\sum_{j \neq i} a_{ij} \frac{u_j}{u_i}| \leq |\sum_{j \neq i} a_{ij}||\sum_{j \neq i} \frac{u_j}{u_i}|. \tag{22}$$

Now if there exist some $u_i \neq 0$ that has the largest absolute value in u then (**??**) would contradict **??**. On the other hand if all elements u_i in u_j have the same magnitude this would also be a contradiction as then $|\sum_{j \neq i} \frac{u_j}{u_i}| < 1$.