Strasbourg University **TI Santé, IR, M1 ASI**
Télécom Physique Strasbourg **Year 2021-2022**

An Introduction to Optimization

Practical session

Part 1.

1.1 Write a Matlab function that solves a linear programming problem by finding and comparing all the BFS. Consider that the problem is given in standard form. The function takes as input A, b and c, and returns the optimal cost f, the optimal BFS x and the corresponding basis B (which contains the indices of the basic columns) :

```
[f,x,B] = LP_bourrin(A,b,c)
```

Test your solution on the LP problems in $Q4$ and $Q5$ of the exercises.
Hint : it would be useful to use the `nchoosek` Matlab function.

1.2 Write a Matlab function that solves a linear programming problem using the (one phase) simplex Method. Consider that the problem is given in standard form. The function takes as input A, b, c and v which contains the indices of the columns of the basis corresponding to an initial BFS. It returns the optimal cost f, the optimal BFS x and the corresponding basis B :

```
[f,x,B] = LP_Simplex(A,b,c,v)
```

Test your solution on the LP problem in $Q4$ of the exercises.

1.3 Write a Matlab function that solves a linear programming problem using the two-phase simplex Method. Consider that the problem is given in standard form. The function takes as input A, b and c. It returns the optimal cost f, the optimal BFS x and the corresponding basis B :

```
[f,x,B] = LP_Two_Phase_Simplex(A,b,c)
```

Test your solution on the LP problem of the example in slide 45 of the Linear Programming lecture and on the LP problem in $Q4$ of the exercises
Hint : it would be useful to use the `LP_Simplex` function.

Part 2.

2.1 Write a Matlab routine to solve the problem :

$$\begin{aligned} \min_{x} \quad & f(x) = x^4 + 4x^3 + 9x^2 + 6x + 6 \\ \text{s.t.} \quad & x \in [-2, 2] \end{aligned}$$

using the Golden section with a tolerance $\epsilon = 10^{-2}$.

2.2 Write a Matlab routine to solve the problem :

$$\begin{aligned} \min_{x} \quad & f(x) = 2x^4 - 5x^3 + 100x^2 + 30x - 75 \\ \text{s.t.} \quad & x \in \mathbb{R} \end{aligned}$$

with a stopping criterion $|\frac{df}{dx}(x_k)| \leq 10^{-4}$ using :

1. Newton's method, with an initial condition $x_0 = 2$
2. Secant method, with initial conditions $x_0 = 2.1$ and $x_1 = 2$

Part 3.

3.1 Consider the following problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = 1 + 2x_1 e^{-{x_1}^2 - {x_2}^2} \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{R}^2 \end{aligned}$$

Write a Matlab routine to solve the problem using the gradient methd with a fixed step size $\alpha = 0.1$. Consider the stopping criterion $\|\nabla f(\mathbf{x}_k)\| < \epsilon = 10^{-3}$. For the initial condition $\mathbf{x}_0$ consider the cases $[-0.5 \;\; 0.5]^T$, $[0.5 \;\; -0.5]^T$ and $[1 \;\; 1]^T$.

3.2 Consider the problem of finding the minimizer of Rosenbrock's function :

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = 100(x_2 - {x_1}^2)^2 + (1 - x_1)^2 \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{R}^2 \end{aligned}$$

1. Show analytically that $[1 \;\; 1]^T$ is the unique global minimizer.
2. Implement the steepest descent method with a stopping criterion $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| / \|\mathbf{x}_k\| < \epsilon_1 = 10^{-3}$, starting from $\mathbf{x}_0 = [-1 \;\; 2]^T$.
   At each iteration, use Newton's method for the line search, with initial value $\alpha_0 = 0.1$, and a stopping criterion $\frac{|\alpha_{k+1} - \alpha_k|}{\alpha_k} < \epsilon_2 = 10^{-4}$.

3.3 Consider the problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{q}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{R}^2 \end{aligned}$$

for the cases :

$$Q = \lambda I,\ \lambda > 0, \qquad \forall \mathbf{q} \in \mathbb{R}^2 \qquad \forall \mathbf{x}_0 \in \mathbb{R}^2$$

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{q} = \begin{bmatrix} -3 \\ -3 \end{bmatrix}, \qquad \mathbf{x}_0 = \begin{bmatrix} -2 \\ -7 \end{bmatrix}$$

$$Q = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{q} = \begin{bmatrix} -3 \\ -3 \end{bmatrix}, \qquad \mathbf{x}_0 = \begin{bmatrix} -2 \\ -7 \end{bmatrix}$$

1. Rank the three cases in term of speed of convergence when applying the steepest descend method.
2. Write a Matlab routine to implement the steepest descend method and the conjugate gradient method, and compare their performance.

3.4 Consider the function

$$f(\mathbf{x}) = \frac{5}{2}{x_1}^2 + \frac{1}{2}{x_2}^2 + 2x_1x_2 - 3x_1 - x_2$$

1. Express the function in a standard quadratic form.
2. Write in details all the steps of the conjugate gradient algorithm to find the minimizer of $f(\cdot)$ starting from $\mathbf{x}_0 = [0\ \ 0]^T$.

3.5 Using the DFP algorithm, we want to find the solution to the following problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \underbrace{\begin{bmatrix} 2 & 2 \\ 2 & 10 \end{bmatrix}}_{Q} \mathbf{x} + \underbrace{\begin{bmatrix} 2 & 0 \end{bmatrix}}_{\mathbf{q}^T} \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{R}^2 \end{aligned}$$

1. Find the formula for α_k in terms of Q, $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$ and $\mathbf{d}_k$.
2. Implement the algorithm using Matlab starting from $\mathbf{x}_0 = [0\ \ 0]^T$.

Part 4.

4.1 Let $A \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$ with $m \leq n$, rank $A = m$ and $\mathbf{x}_0 \in \mathbb{R}^n$. Consider the problem

$$\begin{array}{ll} \arg\min_{\mathbf{x}} & \|\mathbf{x} - \mathbf{x}_0\|^2 \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \end{array}$$

Show that the problem has the following unique solution

$$\mathbf{x}^\star = A^T(AA^T)^{-1}\mathbf{b} + (\mathbf{I} - A^T(AA^T)^{-1}A)\mathbf{x}_0$$

4.2 Plot the curves corresponding to the following equations

$$\begin{aligned} 4{x_2}^2 &= 20 - {x_1}^2 \\ x_2 &= {x_1}^4 - 10 \end{aligned}$$

1. Find a least square optimization problem whose solution is a solution of the previous equations (an intersection of the two curves).
2. Implement a Gauss-Newton method to solve the problem and run it for different initial conditions.

4.3 We want to find the parameters of a process that has an output which is linear in time. We conduct $m \geq 2$ measurements $\{y_1, \ldots, y_m\}$ at different instants $\{t_1, \ldots, t_m\}$, and we want to find the line $y = a^\star t + b^\star$ which has the least squared error with respect to the measurements. That is, we want to find $a^\star$ and $b^\star$ such that

$$\begin{array}{ll} [a^\star \ \ b^\star]^T & = \arg\min_{\mathbf{x}} \quad F(\mathbf{x}) = \sum_{i=1}^{m} (y_i - at_i - b)^2 \\ \text{s.t.} & \mathbf{x} \in \mathbb{R}^2 \end{array}$$

Find analytically the solution $[a^\star \ \ b^\star]^T$. Write a Matlab routine to implement the solution for the data :

t	y
1.0779	6.9959
1.4268	9.4782
2.1801	12.0585
2.3138	14.5837
2.9755	17.1019
3.4526	19.4104
4.1062	22.0719
4.6181	24.5620
4.8747	26.8650

(1)

4.4 Consider the set of m perturbed measurements of a sinusoidal signal $\{y_1, \ldots, y_m\}$ at different instants $\{t_1, \ldots, t_m\}$ in (2). We want to fit a sinusoidal signal to the measured data :

$$y = a\sin(\omega t + \phi)$$

by solving a nonlinear least-squares problem

$$\begin{aligned} \arg\min_{\mathbf{x}} \quad & F(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x})^2 \\ \text{s.t.} \quad & \mathbf{x} = [a \;\; \omega \;\; \phi]^T \in \mathbb{R}^3 \end{aligned}$$

1. Find the expression of $f_i(\mathbf{x})$ in the objective function.
2. Using Matlab, find a solution to the fitting problem using Gauss-Newton method. Consider $\mathbf{x}_0 = [0.5\ 1.25\ 0.1]^T$.

t	y
0	0.1128
0.2300	0.0876
0.4800	0.3971
0.7300	0.5766
0.9800	0.8538
1.2300	0.7856
1.4800	0.9596
1.7300	0.8259
1.9800	0.7542
2.2300	0.8918
2.4800	0.5151
2.7300	0.4231
2.9800	0.3269
3.2300	-0.1352
3.4800	-0.2729
3.7300	-0.4683
3.9800	-0.7837
4.2300	-0.8328
4.4800	-1.1336
4.7300	-0.8258
4.9800	-1.0839
5.2300	-0.9525
5.4800	-0.5864
5.7300	-0.5132
5.9800	-0.1718
6.2300	-0.0748

(2)

Solution to problem 2.1

First, note that here the solution can be found analytically :

$$\frac{df}{dx}(x) = 4x^3 + 12x^2 + 18x + 6 = 0$$

we find $x^\star = -0.4464$. The second derivative is strictly positive at this point, thus $x^\star$ is a local minimizer and the minimum is $f(x^\star) = 4.7989$.

After N steps, the search domain will be reduced to $(1-\rho)^N(b_0 - a_0)$. Thus, the number of steps necessary to obtain a precision ϵ is

$$N_{golden\,section} \geq \frac{\log(\frac{\epsilon}{b_0 - a_0})}{\log(1-\rho)} = 13$$

In the case of Fibonacci search, we need the generate the series for

$$F_{N_{Fibonnaci}} \geq \frac{b_0 - a_0}{\epsilon}$$

See the Matlab files for the implementation of the Golden section search (and the Fibonacci search) : after 13 iterations, we obtain the domain $[-0.4521, -0.4444]$.

Note that if we want to solve the problem by doing a simple griding, we need to calculate the function at approximately $\frac{b_0 - a_0}{\epsilon} = 400$ points.

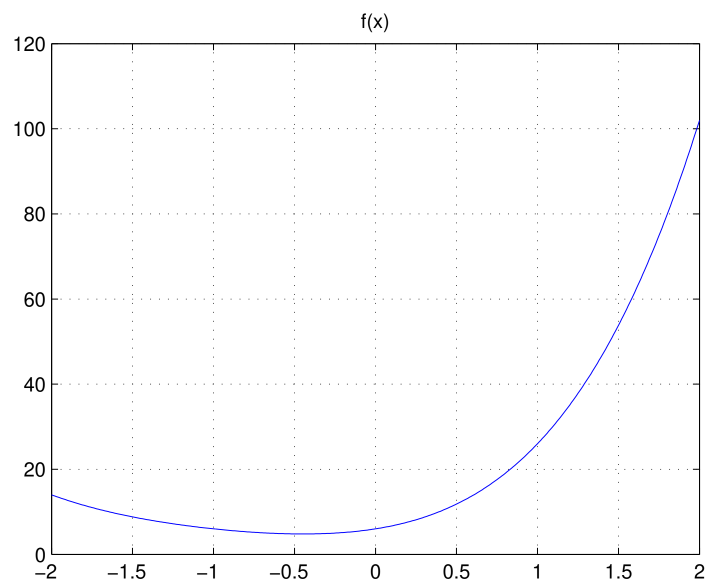


FIGURE 1 – $x^4 + 4x^3 + 9x^2 + 6x + 6$

Solution to problem 2.2

First, note that here the solution can be found analytically

$$\frac{df}{dx}(x) = 8x^3 - 15x^2 + 200x + 30 = 0$$

we find $x^\star = -0.1482$. The second derivative is strictly positive at this point, thus $x^\star$ is a local minimizer and the minimum is $f(x^\star) = -77.2324$.

Algorithm 1 Newton's method

procedure NEWTON(x_0, ϵ, N_{max})
  $x_k \leftarrow x_0$
  $\mathbf{g}_k \leftarrow 8{x_0}^3 - 15{x_0}^2 + 200x_0 + 30$
  $\mathbf{h}_k \leftarrow 24{x_0}^2 - 30x_0 + 200$
  $n \leftarrow 0$
  while $|\mathbf{g}_k| \geq \epsilon$, $n \leq N_{max}$ and $\mathbf{h}_k > 0$ **do**
    $x_k \leftarrow x_k - \frac{\mathbf{g}_k}{\mathbf{h}_k}$
    $\mathbf{g}_k \leftarrow 8{x_k}^3 - 15{x_k}^2 + 200x_k + 30$
    $\mathbf{h}_k \leftarrow 24{x_k}^2 - 30x_k + 200$
    $n \leftarrow n + 1$
  end while
  return x_k
end procedure

Algorithm 2 Secant method

procedure SECANT(x_0, x_1, ϵ, N_{max})
  $x_{k-1} \leftarrow x_0$
  $x_k \leftarrow x_1$
  $\mathbf{g}_{k-1} \leftarrow 8{x_0}^3 - 15{x_0}^2 + 200x_0 + 30$
  $\mathbf{g}_k \leftarrow 8{x_1}^3 - 15{x_1}^2 + 200x_1 + 30$
  $n \leftarrow 0$
  while $|\mathbf{g}_k| \geq \epsilon$, $n \leq N_{max}$ and $\frac{\mathbf{g}_k - \mathbf{g}_{k-1}}{x_k - x_{k-1}} > 0$ **do**
    $x_{k+1} \leftarrow x_k - \frac{x_k - x_{k-1}}{\mathbf{g}_k - \mathbf{g}_{k-1}}\mathbf{g}_k$
    $x_{k-1} \leftarrow x_k$
    $x_k \leftarrow x_{k+1}$
    $\mathbf{g}_{k-1} \leftarrow \mathbf{g}_k$
    $\mathbf{g}_k \leftarrow 8{x_k}^3 - 15{x_k}^2 + 200x_k + 30$
    $n \leftarrow n + 1$
  end while
  return x_k
end procedure

Using Newton's method we find $x^\star$ with an error of $-1.4619\ 10^{-12}$ in 4 iterations. Using the secant method we find $x^\star$ with an error of $-2.7173\ 10^{-12}$ in 5 iterations.

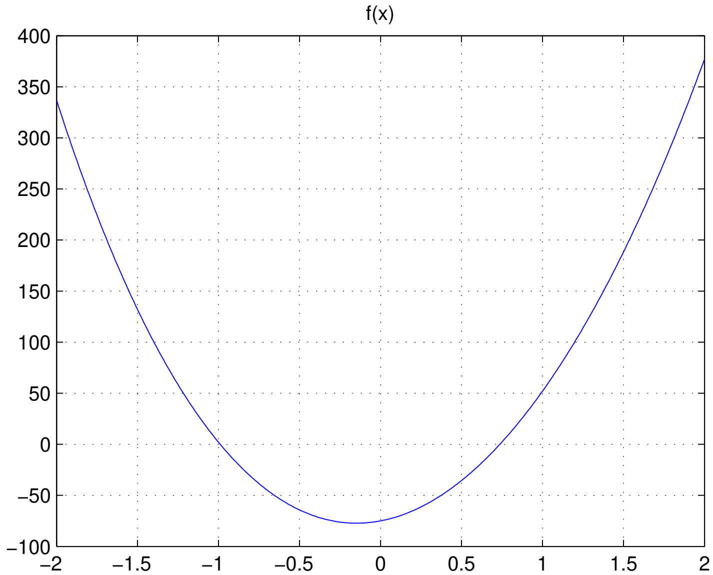


FIGURE 2 – $f(x) = 2x^4 - 5x^3 + 100x^2 + 30x - 75$

Part 3.

Solution to problem 3.1

The following solution is for the steepest-descent method. This year we make them do the fixed step gradient method ($\alpha_k = alpha > 0$) which is much more easier.

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 2 - 4{x_1}^2 \\ -4x_1x_2 \end{bmatrix} e^{-{x_1}^2 - {x_2}^2}$$

$$D^2 f(\mathbf{x}) = \begin{bmatrix} -12x_1 + 8{x_1}^3 & 8{x_1}^2 x_2 - 4x_2 \\ 8{x_1}^2 x_2 - 4x_2 & -4x_1 + 8x_1{x_2}^2 \end{bmatrix} e^{-{x_1}^2 - {x_2}^2}$$

(We will not need it here).
Considering the search direction

We use the notation $\mathbf{x}_k = [x_{k1} \quad x_{k2}]^T$. The search directions are defined at each iteration by

$$\mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = -\nabla f(\mathbf{x}_k) = \begin{bmatrix} (-2 + 4x_{k1}^2) e^{-x_{k1}^2 - x_{k2}^2} \\ (4x_{k1}x_{k2}) e^{-x_{k1}^2 - x_{k2}^2} \end{bmatrix}$$

thus

$$\mathbf{x}_k + \alpha \mathbf{d} = \begin{bmatrix} x_{k1} + \alpha d_1 \\ x_{k2} + \alpha d_2 \end{bmatrix}$$

At each iteration we conduct a line search :

$$\alpha_k = \arg\min_{\alpha \geq 0} h_k(\alpha) = \arg\min_{\alpha \geq 0} f\big(\mathbf{x}_k + \alpha \mathbf{d})\big)$$

$$\begin{aligned}
\dot{h}_k(\alpha) &= \nabla f(\mathbf{x}_k + \alpha \mathbf{d})^T (\mathbf{d}) \\
&= \left(\begin{bmatrix} 2 - 4(x_{k1} + \alpha d_1)^2 \\ -4(x_{k1} + \alpha d_1)(x_{k2} + \alpha d_2) \end{bmatrix} e^{-(x_{k1}+\alpha d_1)^2 - (x_{k2}+\alpha d_2)^2} \right)^T \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \\
&= \Big(2d_1 - 4d_1(x_{k1} + \alpha d_1)^2 \; - 4d_2(x_{k1} + \alpha d_1)(x_{k2} + \alpha d_2) \Big) e^{-(x_{k1}+\alpha d_1)^2 - (x_{k2}+\alpha d_2)^2}
\end{aligned}$$

From FONC we have that $\dot{h}_k(\alpha_k) = 0$. To find α_k we solve

$$\begin{aligned}
0 &= 2d_1 - 4d_1(x_{k1} + \alpha d_1)^2 \; - 4d_2(x_{k1} + \alpha d_1)(x_{k2} + \alpha d_2) \\
0 &= \alpha^2(-4d_1 d_2^2 - 4d_1^3) + \alpha(-8d_1^2 x_{k1} - 4d_1 d_2 x_{k2} - 4d_2^2 x_{k1}) + (2d_1 - 4{x_{k}}_1^2 d_1 - 4x_{k1} x_{k2} d_2)
\end{aligned}$$

We solve the last equation, and then decide which of the roots is α_k : we compare the value of the $h_k(\alpha) = f\big(\mathbf{x}_k + \alpha \mathbf{d})\big)$ at the found roots, and choose the one that minimizes the function. We can otherwise check the sign of $\ddot{h}(\alpha)$ and choose the root that makes it positive.

Algorithm 3 Steepest descent

procedure STEEPEST-DESCENT($\mathbf{x}_0$, ϵ, n_{max})
    $\mathbf{x}_k \leftarrow \mathbf{x}_0$
    $\mathbf{g}_k \leftarrow \nabla f(\mathbf{x}_0)$
    $k \leftarrow 0$
    while $\|\mathbf{g}_k\| \geq \epsilon$ and $k \leq k_{max}$ **do**

$$\mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = -\nabla f(\mathbf{x}_k) = \begin{bmatrix} (-2 + 4{x_k}_1^2) e^{-{x_k}_1^2 - {x_k}_2^2} \\ (4x_{k1} x_{k2}) e^{-{x_k}_1^2 - {x_k}_2^2} \end{bmatrix}$$

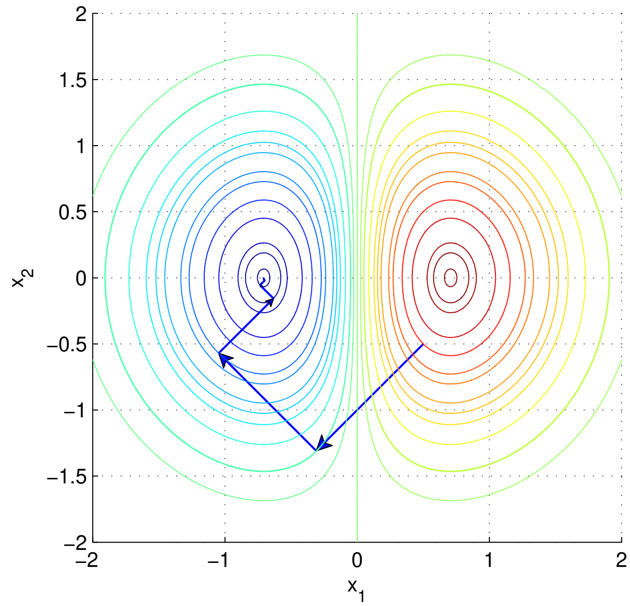
$$\alpha_k = \arg\min_{\alpha \geq 0} f\big(\mathbf{x}_k + \alpha \mathbf{d})\big)$$

        $\mathbf{x}_k \leftarrow \mathbf{x}_k + \alpha_k \mathbf{d}$
        $n \leftarrow n + 1$
    end while
    return x_k
end procedure

For $\epsilon = 0.001$ we find the minimizer

$$\mathbf{x}^\star = \begin{bmatrix} -0.7070 \\ -0.0002 \end{bmatrix}$$

and the minimum value is $f^\star = f(\mathbf{x}^\star) = 0.1422$.



Solution to problem 3.2

$$\nabla f(\mathbf{x}) = \begin{bmatrix} -400x_1(x_2 - x_1{}^2) - 2(1 - x_1) \\ 200(x_2 - x_1{}^2) \end{bmatrix}$$

$\nabla f(\mathbf{x}^\star) = \mathbf{0} \Rightarrow \mathbf{x}^\star = [1 \;\; 1]^T$. Moreover,

$$D^2 f(\mathbf{x}) = \begin{bmatrix} -400x_2 + 1200x_1{}^2 + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix}$$

thus

$$D^2 f(\mathbf{x}^\star) = \begin{bmatrix} 802 & -400 \\ -400 & 200 \end{bmatrix}$$

We can check that the matrix has two strictly positive eigenvalues, this shows that the $D^2 f(\mathbf{x}^\star) \succ 0$. From SOSC we see that $\mathbf{x}^\star$ a unique (local) minimizer.
In fact, it is easy to see that $f(\mathbf{x}) > f(\mathbf{x}^\star)\ \forall \mathbf{x} \in \mathbb{R}^2$ such that $\mathbf{x} \neq \mathbf{x}^\star$.

At each iteration a line search must be done :

$$\alpha_k = \arg\min_{\alpha \geq 0} h_k(\alpha) = \arg\min_{\alpha \geq 0} f\big(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)\big)$$

This is done using Newton's method starting from $\alpha_0 = 0.1$:

$$\alpha_{j+1} = \alpha_j + \frac{\dot{h}_k(\alpha_j)}{\ddot{h}_k(\alpha_j)}$$

with

$$\begin{aligned} \dot{h}_k(\alpha) &= Df(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)) Dg(\alpha) \\ &= \nabla f(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k))^T (-\nabla f(\mathbf{x}_k)) \end{aligned}$$

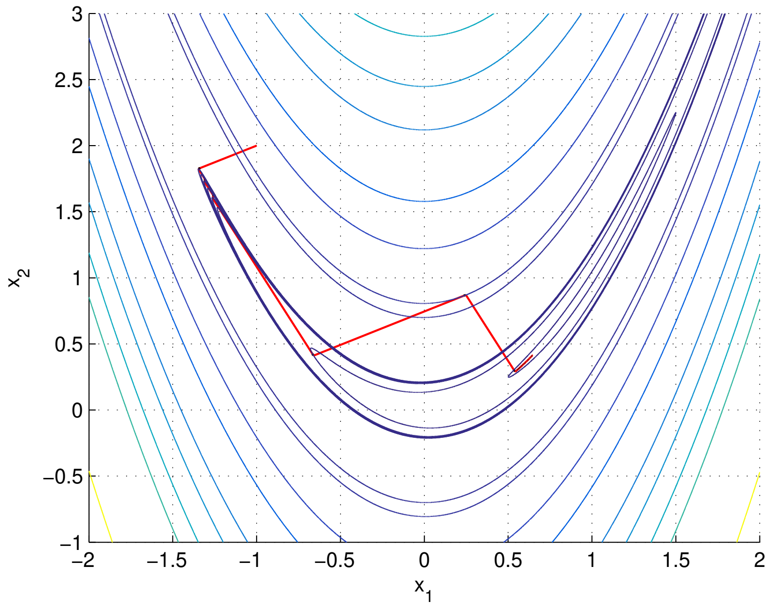


FIGURE 3 – Finding the minimizer of Rosenbrock's function using the steepest descent method

which can be calculated using the chain rule. The second derivative is

$$
\begin{aligned}
\ddot{h}_k(\alpha) &= -D\Big(\nabla f(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k))^T \nabla f(\mathbf{x}_k)\Big) \\
&= -\nabla f(\mathbf{x}_k)^T D\big(\nabla f(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k))\big) \\
&= -\nabla f(\mathbf{x}_k)^T \big(D^2 f(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k))\big)(-\nabla f(\mathbf{x}_k)) \\
&= \nabla f(\mathbf{x}_k)^T \big(D^2 f(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k))\big) \nabla f(\mathbf{x}_k))
\end{aligned}
$$

which can be calculated using the product and the chain rules

One can also calculate $\dot{h}_k(\alpha)$ and $\ddot{h}_k(\alpha)$ by simply developing $h_k(\alpha)$.

Note that the method 'gets stuck' with a very bad zigzag convergence near the optimum point

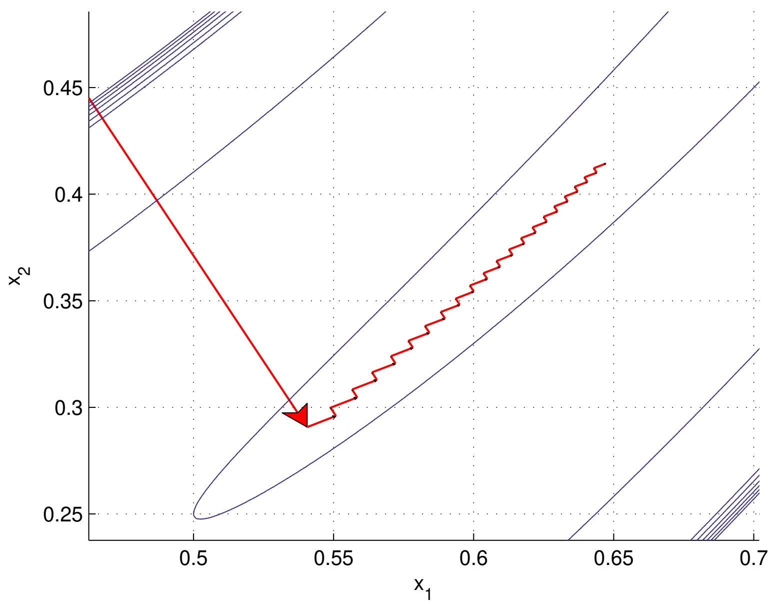
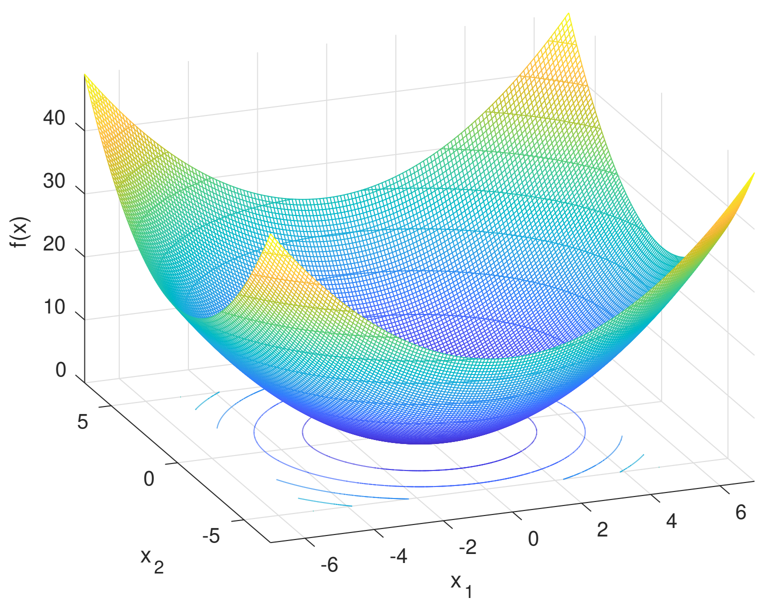
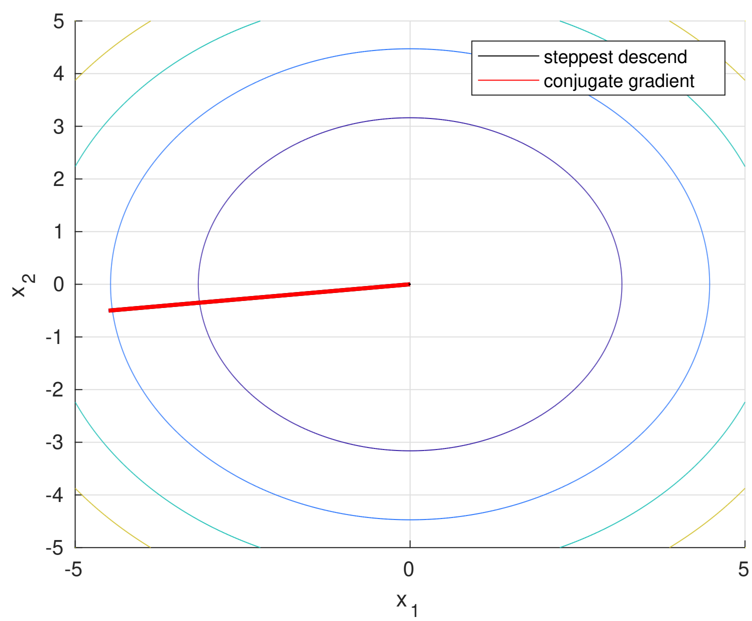


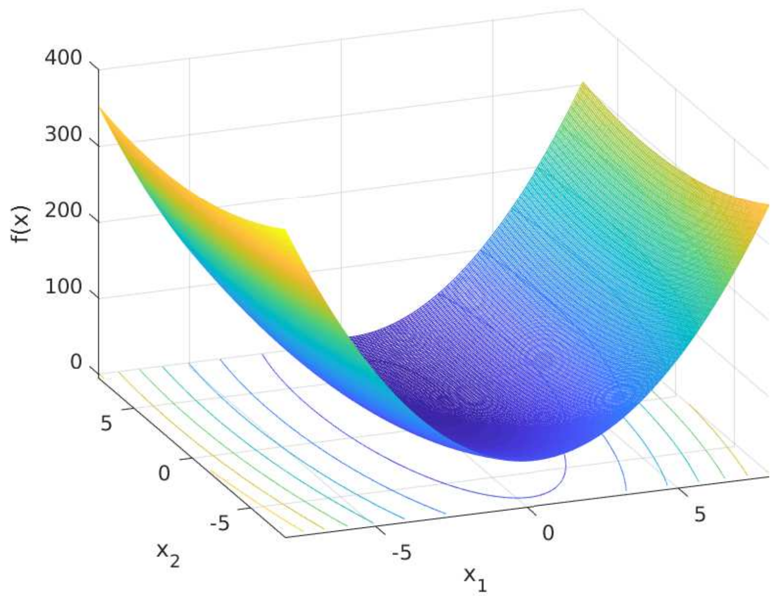
FIGURE 4 – A zoomed part of Figure 3

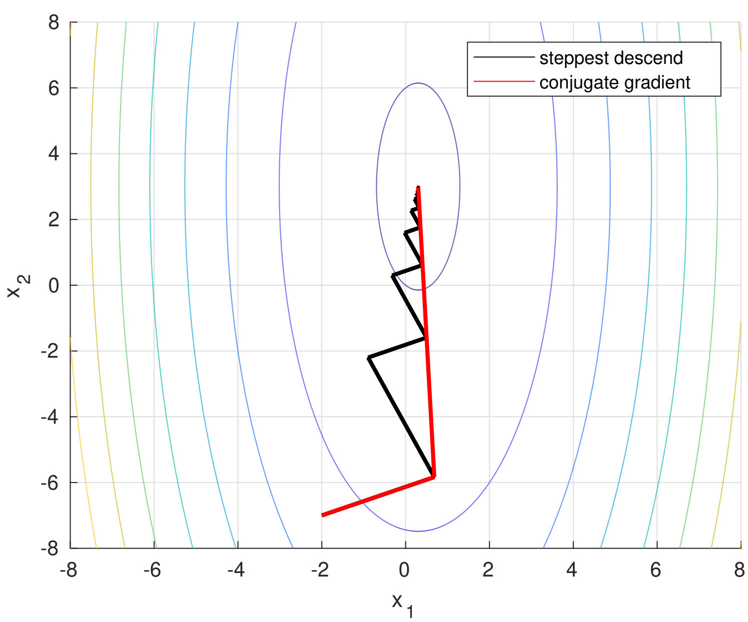
Solution to problem 3.3

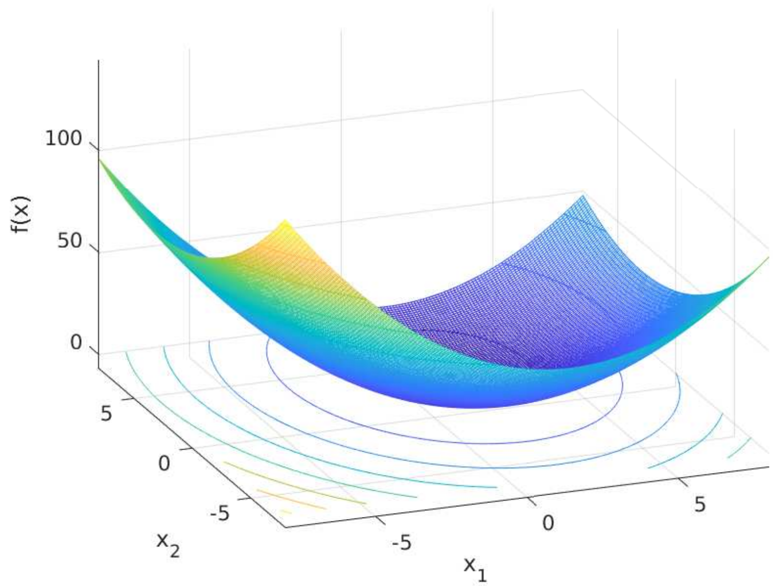
When applying the steepest descend method, it is easy to see from Theorem 4 in *Lecture 3 Part I* that the first case will converge in one step for any initial condition. The second case and the third case have a linear convergence, and since the matrix Q has a smaller condition number for the third case, then it will converges faster than the second case.
The conjugate gradient method converges in n steps at most.



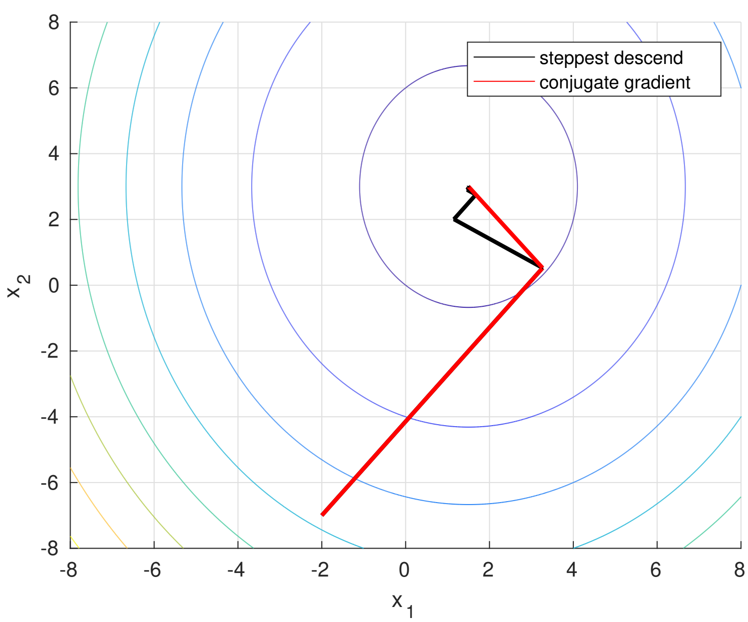



f(x)
x2
x1


steppest descend
conjugate gradient
x2
x1



100
50
0
f(x)
5
0
-5
x2
-5
0
5
x1



steppest descend
conjugate gradient
x2
x1

Solution to problem 3.4

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) = \frac{5}{2}{x_1}^2 + \frac{1}{2}{x_2}^2 + 2x_1x_2 - 3x_1 - x_2$$
$$\text{s.t.} \quad \mathbf{x} \in \mathbb{R}^2$$

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \underbrace{\begin{bmatrix} 5 & 2 \\ 2 & 1 \end{bmatrix}}_{Q} \mathbf{x} + \underbrace{\begin{bmatrix} -3 & -1 \end{bmatrix}}_{q^T} \mathbf{x}$$
$$\text{s.t.} \quad \mathbf{x} \in \mathbb{R}^2$$

Initialization :

$$\mathbf{x}_0 = \mathbf{0} \qquad \nabla \mathbf{g}_0 = f(\mathbf{x}_0) = Q\mathbf{x}_0 + q = \begin{bmatrix} -3 \\ -1 \end{bmatrix}$$

$$\mathbf{d}_0 = -\mathbf{g}_0 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

Iteration 1 :

$$\alpha_0 = -\frac{\mathbf{g}_0^T \mathbf{d}_0}{\mathbf{d}_0^T Q \mathbf{d}_0} = -\frac{\begin{bmatrix} -3 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix}}{\begin{bmatrix} 3 & 1 \end{bmatrix} \begin{bmatrix} 5 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix}} = \frac{10}{58} = \frac{5}{29}$$

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{d}_0 = \begin{bmatrix} \frac{15}{29} \\ \frac{5}{29} \end{bmatrix}$$

$$\mathbf{g}_1 = Q\mathbf{x}_1 + q = \begin{bmatrix} -\frac{2}{29} \\ \frac{6}{29} \end{bmatrix}$$

$$\beta_0 = \frac{\mathbf{g}_1^T Q \mathbf{d}_0}{\mathbf{d}_0^T Q \mathbf{d}_0} = \frac{\begin{bmatrix} -\frac{2}{29} & \frac{6}{29} \end{bmatrix} \begin{bmatrix} 5 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix}}{58} = \frac{\frac{8}{29}}{58} = \frac{4}{29^2} = \frac{4}{841}$$

$$\begin{aligned} \mathbf{d}_1 &= -\mathbf{g}_1 + \beta_0 \mathbf{d}_0 \\ &= -\begin{bmatrix} -\frac{2}{29} \\ \frac{6}{29} \end{bmatrix} + \frac{4}{841} \begin{bmatrix} 3 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{70}{841} \\ \frac{-170}{841} \end{bmatrix} \end{aligned}$$

Iteration 2 :

$$\alpha_1 = -\frac{\mathbf{g}_1^T \mathbf{d}_1}{\mathbf{d}_1^T Q \mathbf{d}_1} = -\frac{\begin{bmatrix} -\frac{2}{29} & \frac{6}{29} \end{bmatrix} \begin{bmatrix} \frac{70}{841} \\ -\frac{170}{841} \end{bmatrix}}{\begin{bmatrix} \frac{70}{841} & -\frac{170}{841} \end{bmatrix} \begin{bmatrix} 5 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} \frac{70}{841} \\ -\frac{170}{841} \end{bmatrix}} = \frac{58}{10}$$

$$\mathbf{x}_2 = \mathbf{x}_1 + \alpha_1 \mathbf{d}_1 = \begin{bmatrix} \frac{15}{29} \\ \frac{5}{29} \end{bmatrix} + \frac{58}{10} \begin{bmatrix} \frac{70}{841} \\ \frac{-170}{841} \end{bmatrix} = \begin{bmatrix} \frac{15\times 29+58\times 7}{841} \\ \frac{5\times 29-58\times 17}{841} \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

The minimizer is $\mathbf{x}^\star = [1 \quad -1]^T$ and $f^\star = -1$.

Solution to problem 3.5

$$\alpha_k = \arg\min_{\alpha \geq 0} h_k(\alpha) = \arg\min_{\alpha \geq 0} f\big(\mathbf{x}_k + \alpha \mathbf{d}\big)$$

$$\begin{aligned} 0 &= \dot{h}(\alpha_k) \\ &= \nabla f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)^T \mathbf{d}_k \\ &= (Q(\mathbf{x}_k + \alpha_k \mathbf{d}_k) + q)^T \mathbf{d}_k \\ &= (\mathbf{g}_k + \alpha_k Q \mathbf{d}_k)^T \mathbf{d}_k \end{aligned}$$

thus

$$\alpha_k = -\frac{\mathbf{g}_k^T \mathbf{d}_k}{\mathbf{d}_k^T Q \mathbf{d}_k}$$

See the Matlab code for the implementation $\mathbf{x}^\star = [-\frac{5}{4} \quad \frac{1}{4}]^T$. Note $\mathbf{H}_2 = Q^{-1}$ for any definite positive $\mathbf{H}_0$.

Solution to problem 4.1

By considering the change of variable $\mathbf{z} = \mathbf{x} - \mathbf{x}_0$, thus $A\mathbf{z} = A\mathbf{x} - A\mathbf{x}_0 = \mathbf{b} - A\mathbf{x}_0 = \tilde{\mathbf{b}}$. the problem is equivalent to

$$\begin{aligned} \min_{\mathbf{z}} \quad & \|\mathbf{z}\| \\ \text{s.t.} \quad & A\mathbf{z} = \tilde{\mathbf{b}} \end{aligned}$$

For this problem the solution is

$$\mathbf{z}^\star = A^T(AA^T)^{-1}\tilde{\mathbf{b}} = A^T(AA^T)^{-1}(\mathbf{b} - A\mathbf{x}_0) = A^T(AA^T)^{-1}\mathbf{b} - A^T(AA^T)^{-1}A\mathbf{x}_0$$

thus

$$\mathbf{x}^\star = \mathbf{z}^\star - \mathbf{x}_0 = A^T(AA^T)^{-1}\mathbf{b} + (I - A^T(AA^T)^{-1}A)\mathbf{x}_0$$

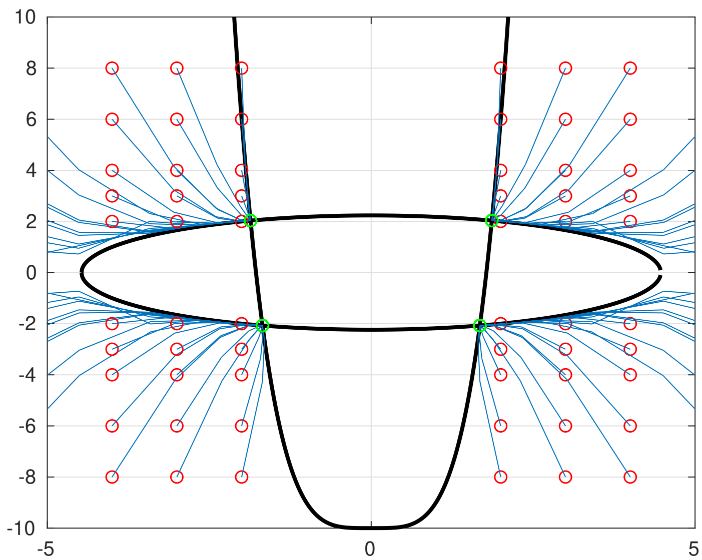
Solution to problem 4.2

$$\begin{aligned} \arg\min_{\mathbf{x}} \quad & F(\mathbf{x}) = (4{x_2}^2 - 20 + {x_1}^2)^2 + (2x_2 - 2{x_1}^4 + 20)^2 \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{R}^2 \end{aligned}$$

$$F(\mathbf{x}) = f(\mathbf{x})^T f(\mathbf{x}), \qquad f(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 4{x_2}^2 - 20 + {x_1}^2 \\ 2x_2 - 2{x_1}^4 + 20 \end{bmatrix}$$

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 2x_1 & 8x_2 \\ -8{x_1}^3 & 2 \end{bmatrix}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (J(\mathbf{x}_k)^T J(\mathbf{x}_k))^{-1} J(\mathbf{x}_k)^T f(\mathbf{x}_k)$$

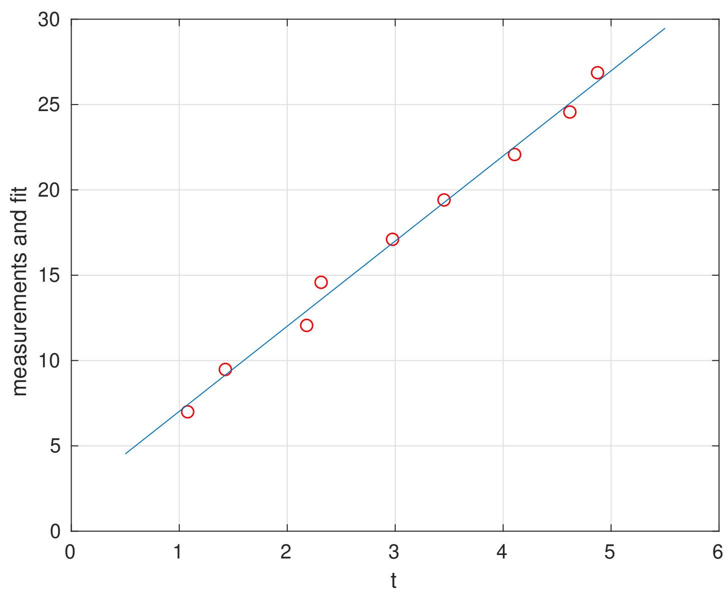


Solution to problem 4.3

$$A = \begin{bmatrix} t_1 & 1 \\ \vdots & \vdots \\ t_m & 1 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

$$\begin{aligned}
\mathbf{x}^\star &= \begin{bmatrix} a^* \\ b^* \end{bmatrix} = (A^T A)^{-1} A^T \mathbf{b} \\
&= \frac{1}{\frac{\sum_{i=1}^m t_i^2}{m} - (\frac{\sum_{i=1}^m t_i}{m})^2} \begin{bmatrix} \frac{\sum_{i=1}^m t_i y_i}{m} - \frac{\sum_{i=1}^m t_i}{m} \frac{\sum_{i=1}^m y_i}{m} \\ \frac{\sum_{i=1}^m t_i^2}{m} \frac{\sum_{i=1}^m y_i}{m} - \frac{\sum_{i=1}^m t_i}{m} \frac{\sum_{i=1}^m t_i y_i}{m} \end{bmatrix}
\end{aligned}$$

$$\mathbf{x}^\star = \begin{bmatrix} 4.9885 \\ 2.0345 \end{bmatrix}$$



Solution to problem 4.4

$$f_i(\mathbf{x}) = y_i - a\sin(\omega t_i + \phi)$$

$$F(\mathbf{x}) = f(\mathbf{x})^T f(\mathbf{x}), \qquad f(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}$$

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial a}(\mathbf{x}) & \frac{\partial f_1}{\partial \omega}(\mathbf{x}) & \frac{\partial f_1}{\partial \phi}(\mathbf{x}) \\ \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial a}(\mathbf{x}) & \frac{\partial f_m}{\partial \omega}(\mathbf{x}) & \frac{\partial f_m}{\partial \phi}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} -\sin(\omega t_1 + \phi) & -a t_1 \cos(\omega t_1 + \phi) & -a\cos(\omega t_1 + \phi) \\ \vdots & \vdots & \vdots \\ -\sin(\omega t_m + \phi) & -a t_m \cos(\omega t_m + \phi) & -a\cos(\omega t_m + \phi) \end{bmatrix}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (J(\mathbf{x}_k)^T J(\mathbf{x}_k))^{-1} J(\mathbf{x}_k)^T f(\mathbf{x}_k)$$

$$\mathbf{x}^\star = \begin{bmatrix} a^\star \\ \omega^\star \\ \phi^\star \end{bmatrix} = \begin{bmatrix} 0.9573 \\ 1.0086 \\ -0.0409 \end{bmatrix}$$