Problem 1: Optimization Theory

a) Proof:

Let f(x) be a local minimum for f. By the definition of local minimum there is a d > 0 such that f(x) < f(x_0) for all x_0 in S with || x_0-x || < δ. Now, pick arbitrary y in S.

Since S is convex set, we have

$$||x - (\alpha y + (1 - \alpha)x)|| \leq |\alpha|\ ||x - y||$$

and thus that

$$||x - (ty + (1 - t)x)|| < d$$

whenever 0 < α < δ / (||x|| + ||y|| + 1).

Thus, we have

$$f(x) \leq f(\alpha y + (1 - \alpha)x) \leq \alpha f(y) + (1 - \alpha)f(x)$$

$$\Leftrightarrow tf(x) \leq tf(y)$$

whenever 0 < t < d(||x|| + ||y|| + 1).

Thus,

$$f(x) \leq f(y)$$

Thus, any local minimum x of a convex function f on a convex set S ⊆$\mathbb{R}^n$ is a global minimum of f on S.

b) For the upper left quadrant, differentiate $\mathscr{L}(\vec{x}, \vec{\lambda})$ with respect to $x_1, \ldots, x_n$

$$\begin{bmatrix} \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_1 \partial x_1} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_2 \partial x_1} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_n \partial x_1} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_n \partial x_n} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial f^2(\overrightarrow{x})}{\partial x_1 \partial x_1} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i^2(\overrightarrow{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial f^2(\overrightarrow{x})}{\partial x_1 \partial x_n} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i^2(\overrightarrow{x})}{\partial x_1 \partial x_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial f^2(\overrightarrow{x})}{\partial x_n \partial x_1} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i^2(\overrightarrow{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial f^2(\overrightarrow{x})}{\partial x_n \partial x_n} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i^2(\overrightarrow{x})}{\partial x_n \partial x_n} \end{bmatrix}$$

By definition of Hessian matrix, $H_f(\mathbf{x})_{ij} = \frac{\partial f(\overline{x})}{\partial x_i \partial x_j}$ and $H_g(\mathbf{x}) = \frac{\partial^2 g(\overline{x})}{\partial x_i \partial x_j}$. So in the upper left quadrant. The elements calculated:

$$H_f(\overline{x}) = \sum_{i=1}^{m} \lambda_i H_{g_i}(\overline{x})$$

For the upper right differentiate $\mathscr{L}(\vec{x}, \vec{\lambda})$ with respect to x_i , λ_j, $1 \le i \le n$, $1 \le j \le m$

$$\begin{bmatrix} \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_1 \partial x_1} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_1 \partial x_m} \\ \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_2 \partial x_1} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_2 \partial x_m} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_n \partial x_1} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_n \partial x_m} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial g_1(\overrightarrow{x})}{\partial x_1} & \frac{\partial g_2(\overrightarrow{x})}{\partial x_1} & \cdots & \frac{\partial g_m(\overrightarrow{x})}{\partial x_1} \\ \frac{\partial g_1(\overrightarrow{x})}{\partial x_2} & \frac{\partial g_2(\overrightarrow{x})}{\partial x_2} & \cdots & \frac{\partial g_m(\overrightarrow{x})}{\partial x_2} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial g_1(\overrightarrow{x})}{\partial x_n} & \frac{\partial g_2(\overrightarrow{x})}{\partial x_n} & \cdots & \frac{\partial g_m(\overrightarrow{x})}{\partial x_n} \end{bmatrix} = J_g^T(\overrightarrow{x})$$

For the lower left differential $\mathscr{L}(\vec{x}, \vec{\lambda})$ with respect to x_i , λ_j, $1 \le i \le n$, $1 \le j \le m$

$$\begin{bmatrix} \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_1 \partial \lambda_1} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_1 \partial \lambda_1} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_1 \partial \lambda_1} \\ \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_2 \partial \lambda_2} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_2 \partial \lambda_2} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_2 \partial \lambda_2} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_n \partial \lambda_m} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_n \partial \lambda_m} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_n \partial \lambda_m} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial g_1(\overrightarrow{x})}{\partial x_1} & \frac{\partial g_2(\overrightarrow{x})}{\partial x_2} & \cdots & \frac{\partial g_m(\overrightarrow{x})}{\partial x_n} \\ \frac{\partial g_1(\overrightarrow{x})}{\partial x_1} & \frac{\partial g_2(\overrightarrow{x})}{\partial x_2} & \cdots & \frac{\partial g_m(\overrightarrow{x})}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial g_1(\overrightarrow{x})}{\partial x_1} & \frac{\partial g_2(\overrightarrow{x})}{\partial x_2} & \cdots & \frac{\partial g_m(\overrightarrow{x})}{\partial x_n} \end{bmatrix} = J_g(\overrightarrow{x})$$

For the lower right, differentiate $\mathscr{L}(\vec{x}, \vec{\lambda})$ with respect to λ_1, $\lambda_{2, \ldots} \lambda_m$

$$\begin{bmatrix} \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial \lambda_1 \partial \lambda_1} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial \lambda_1 \partial \lambda_2} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial \lambda_1 \partial \lambda_m} \\ \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial \lambda_2 \partial \lambda_1} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial x_2 \partial \lambda_2} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial \lambda_2 \partial \lambda_m} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial \lambda_m \partial \lambda_1} & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial \lambda_m \partial \lambda_2} & \cdots & \frac{\partial^2 L(\overrightarrow{x},\overrightarrow{\lambda})}{\partial \lambda_m \partial \lambda_m} \end{bmatrix} = 0$$

Thus, the Hessian of the Lagrangian function $\mathscr{L}$: $R^{m+n} \to \mathbb{R}$ is given by the form in the question.

c) Let $x = [U \;\; P]^T$

where U and P are two parts of the vector **x**.

Then we have

$$x^T H_L(x,y)x = [U^T P^T] \begin{bmatrix} B(x,y) & J_g^T(x) \\ J_g(x) & 0 \end{bmatrix} \begin{bmatrix} U \\ P \end{bmatrix} = U^T B(x,y)U + 2P^T J_g(x)U$$

Where $B(x,y) \;=\; H_f(x) + \sum_{i=1}^{m} \lambda_i H_{gi}(x)$

Let U be zero vector and result in $x_0 = [\mathbf{0} \;\; \boldsymbol{P}]^T$. Then, we have

$x_0^T H_{\mathscr{L}}(x, \lambda)x_0 = 0$

Therefore, we could conclude that $H_{\mathscr{L}}(x, \lambda)$ cannot be positive definite.

d) From the result that $H_{\mathscr{L}}(x, \lambda)$ is non-positive-definiteness, we could conclude that the critical point of $\mathscr{L}$ is a saddle point (which is the solution of $\nabla\boldsymbol{\mathscr{L}}\mathbf{(x, \lambda) = 0}$) rather than a minimum or maximum.

Problem 2: Optimization on quadratic forms

a) The gradient $\nabla f(x) = \boldsymbol{A}x\text{-}b$

The Hessian $H_f(x) = \boldsymbol{A}$

b) f attains a minimum when $\nabla f = 0$ which is at x^* such that $Ax^* = b$

c) By Newton's method, the first step from the starting point x_0 is the solution of the linear system:

$$H_f(x_0)s_0 = -\nabla f(x_0)$$

By the result from a) and b), we have:

$$As_0 = b = Ax_0$$

Then we have $Ax_1 = A(x_0+s_0)=b$.

Thus, $x_1 = x^*$ and hence $e_1 = 0$

d) In the steepest descent method, in the first iteration we have to solve

$min_\alpha f(x_0 + \alpha s_0)$ where $s_0=b\text{-}\boldsymbol{A}x = r$.

Thus, we could write it as

$$min_\alpha f(x_0 + \alpha r)$$

The minimum can then be found when $f'(\alpha)=0$.

$$\alpha = \frac{r^T r}{x^T A r}$$

And thus

$$x = x_0 + \frac{r^T r}{r^T A r} r$$

If $e_0 := x_0 - x^*$ is an eigenvector of A, we have for the corresponding eigenvalue λ, $Ae_0 = \lambda e_0$. But $Ae_0 = Ax_0 - b$ since $Ax^* = b$ and $s_0 = -\nabla f(x_0) = b - Ax_0 = -\lambda e_0$.

Thus, the line search parameter is given by:

$$\alpha = \frac{(-\lambda e_0)^T(-\lambda e_0)}{(-\lambda e_0)^T A(-\lambda e_0)} = \frac{{e_0}^T e_0}{{e_0}^T \lambda e_0} = \frac{1}{\lambda}$$

Therefore, $x_1 = x_0 - \alpha \nabla f(x_0)$

$= x_0 + 1/\lambda(b - Ax_0)$

$= x_0 - \lambda e_0/\lambda$

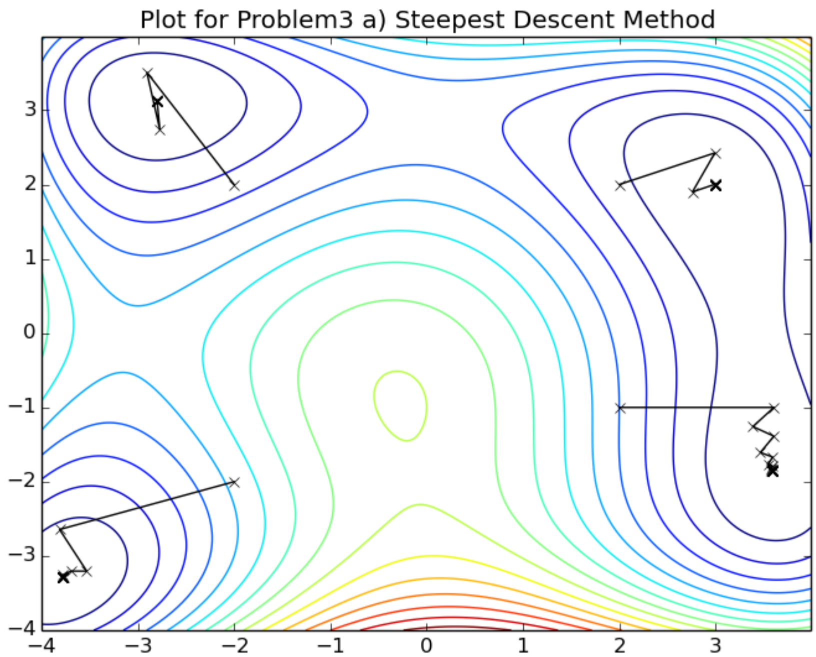
$= x_0 - e_0$

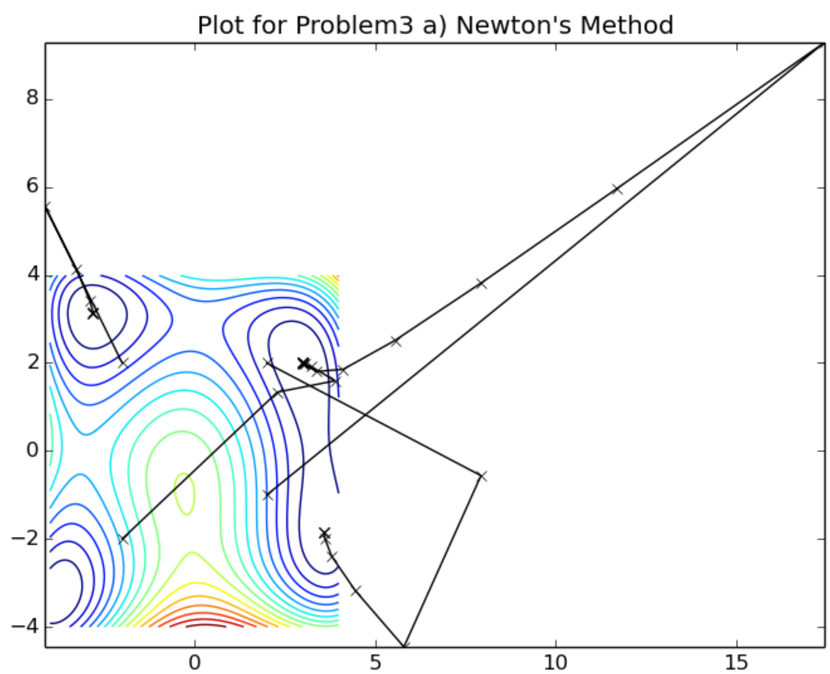
$= x^*$

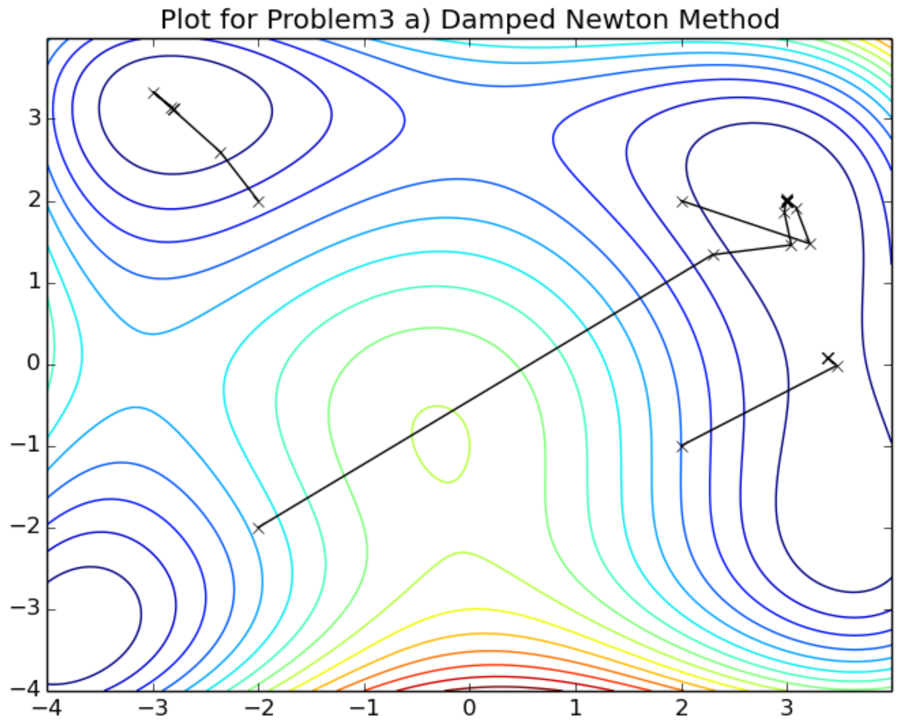
Thus, if the initial error, $e_0 := x_0 - x^*$, is an eigenvector of A, steepest descent method converges to the true solution in one step.

Problem 3: Optimization and Nonlinear Data Fitting

a) Check problem3_a.py







In this case, the method of Steepest descent works the best.Compare to Newton Method and Damped Newton Method, Steepest Descent Method converge to the local minimum very quickly while other method converge slower or even not converge to the local minimum.

b) Check problem3_b.py

i) For [0 0 0 0 1]^T

Terminate with tolerance 10^{-13} , but not with tolerance 10^{-14}

x_1 is

[-12.34845898 102.25484661 -105.62596847   33.6231109   1.   ]

x_2 is

[-398.85294624 -173.14732136 -239.75128678 419.05815536
-1.76747881]

x_3 is

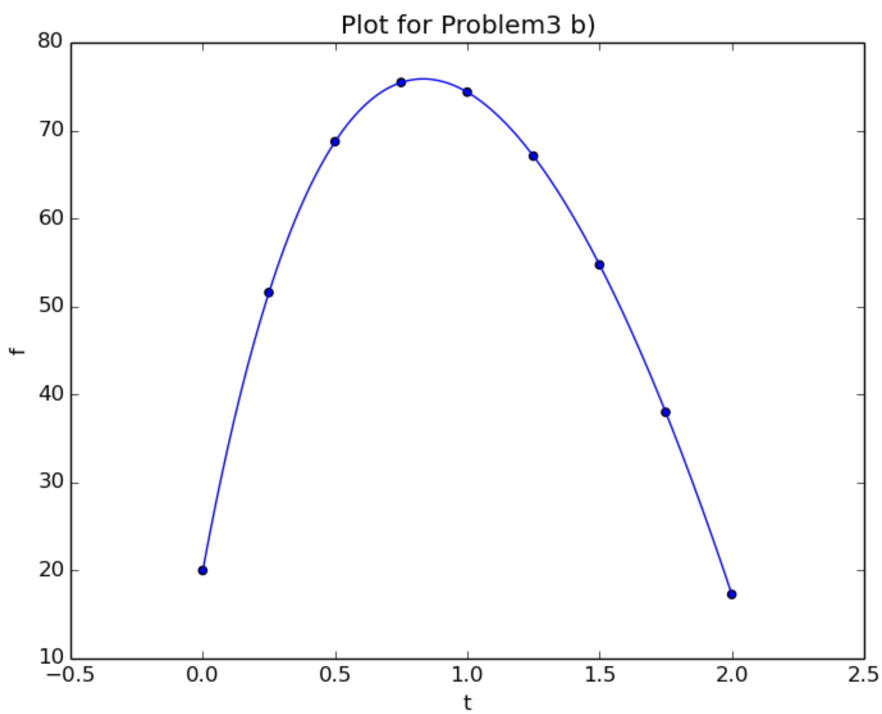
[109.71346884   5.24148438 -25.05623046 -89.7133018   -1.77056387]

x_4 is

[109.70634376   5.24617743 -25.05708885 -89.70617574
-1.75315475]

x_5 is

[109.72907178   5.23121378 -25.05435159 -89.72890644
-1.75288141]



ii) For $[1\ 0\ 0\ 0\ 0]^T$

Terminate with tolerance 10^{-14}

x_1 is

[13.09075758 105.90385281 -55.88692641 12.09075758 0.]

x_2 is

[1.30907576e+01 1.05903853e+02 -5.58869264e+01
1.20907576e+01 1.60497457e-15]

x_3 is

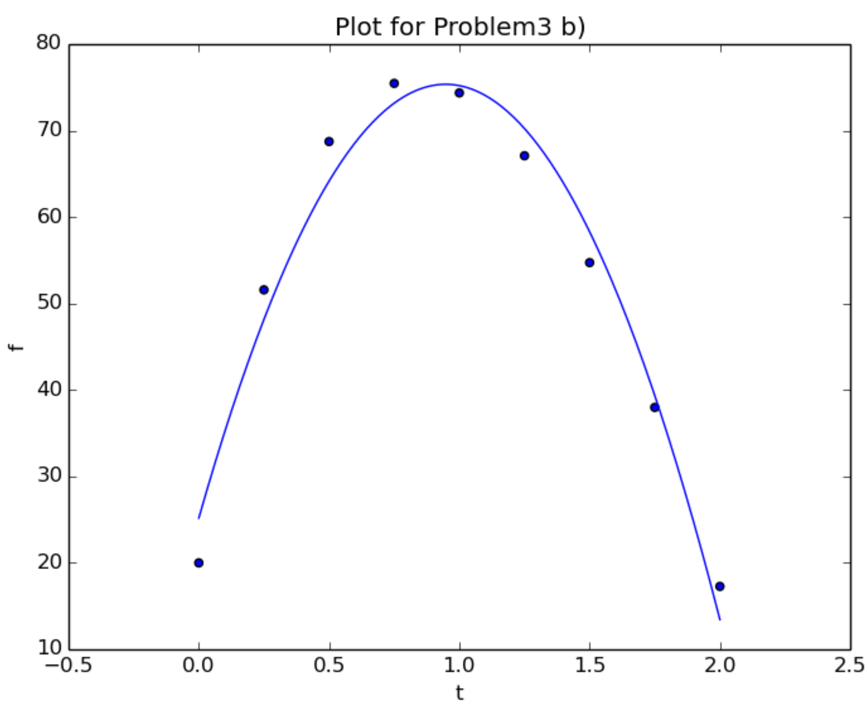
[1.30907576e+01 1.05903853e+02 -5.58869264e+01
1.20907576e+01 1.89982317e-15]

x_4 is

[1.30907576e+01 1.05903853e+02 -5.58869264e+01
1.20907576e+01 1.24825916e-15]

x_5 is

[1.30907576e+01 1.05903853e+02 -5.58869264e+01
1.20907576e+01 7.14818345e-16]



iii) For [1 0 0 1 0]^T

Terminate with tolerance 10^{-14}

x1 is

[12.59075758 52.95192641 -55.88692641 12.59075758 52.95192641]

x2 is

[1.25907576e+01   5.29519263e+01 -5.58869264e+01
5.97897021e-09   5.29519264e+01]

x3 is

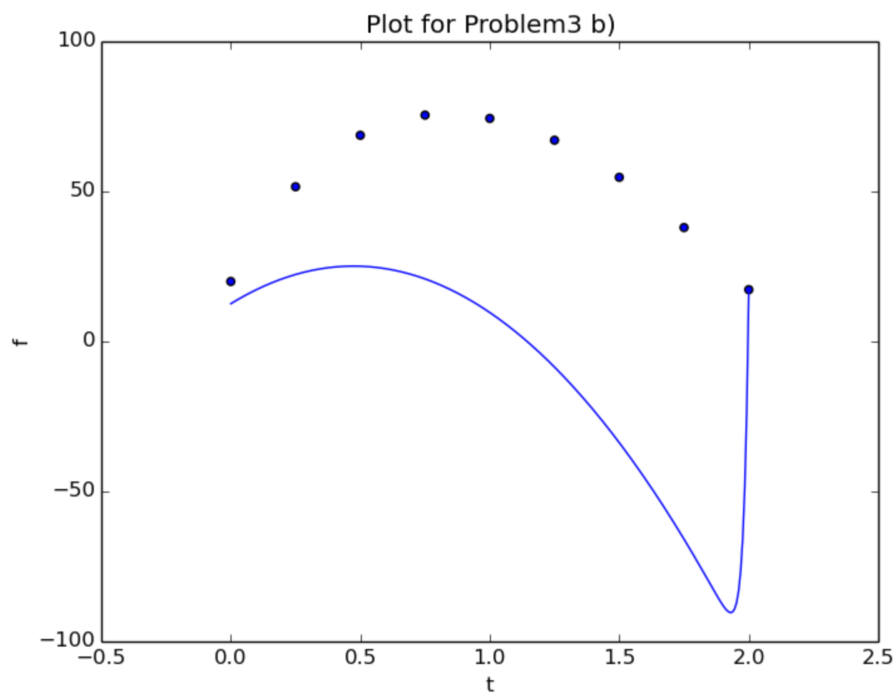
[1.25907576e+01   5.29519263e+01 -5.58869264e+01
2.62148012e-18   5.29519264e+01]

x4 is

[12.59075758 52.95192635 -55.88692641   0.   52.95192641]

x5 is

[1.25907576e+01   5.29519263e+01 -5.58869264e+01
1.24189744e-44   5.29519264e+01]



Compare the result from i), ii) and iii), we observed that the result would be different with different starting vector.

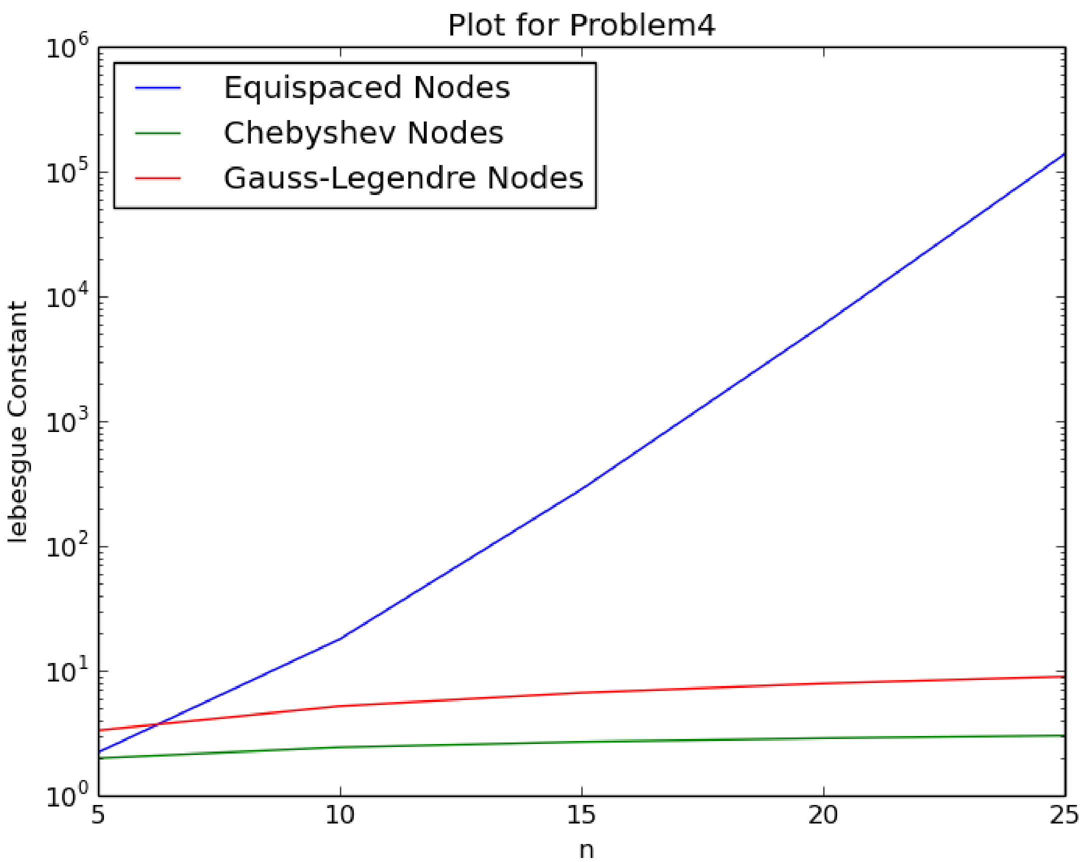
In i) , the starting vector result in a perfect result.

In ii) , the starting vector result in result that is still not too bad.

In iii) , the starting vector result in a bad result.

Problem 4: Stability of Interpolation

Check problem4.py



The result shows that Equispaced Nodes result in a large Lesbegue Const when n increases which matches the Runge phenomenon that the resulting interpolation oscillates toward the end of the interval.

Problem 5: Chebyshev polynomials, Vandermonde matrices

a) First we test it on the two base cases

We have

$F_0(t) = \cos(0 \times \arccos(t)) = \cos(0) = 1$

$F_1(t) = \cos(1 \times \arccos(t)) = \cos(\arccos(t)) = t$

Then we consider F_{n+1} which is

$$
\begin{aligned}
F_{n+1}(t) &= \cos((n+1)\arccos(t)) \\
&= \cos(n \times \arccos(t) + \arccos(t)) \\
&= \cos(n \times \arccos(t))\cos(\arccos(t)) - \sin(n \times \arccos(t))\sin(\arccos(t)) \\
&= 2t\cos(n \times \arccos(t)) - t\cos(n \times \arccos(t)) \\
&\quad - \sin(n \times \arccos(t))\sin(\arccos(t)) \\
&= 2t\cos(n \times \arccos(t)) - [\cos(n \times \arccos(t))\cos(\arccos(t)) \\
&\qquad + \sin(n \times \arccos(t))\sin(\arccos(t))] \\
&= 2t\cos(n \times \arccos(t)) - [\cos(n\times \arccos(t) - \arccos(t))] \\
&= 2t\cos(n \times \arccos(t)) - [\cos((n-1)\times \arccos(t))] \\
&= 2t\,F_n(t) - F_{n-1}(t)
\end{aligned}
$$

Therefore, we could conclude that the given function satisfies the Chebyshev three-term recurrence.

b) Find the roots of charismatic equation $F_n(t) - 2t\, F_{n-1}(t) + F_{n-2}(t) = 0$, we have

$$r_1,\ r_2 = \frac{2t \pm \sqrt{4t^2-1}}{2} = t \pm \sqrt{t^2 - 1}$$

Then we have

$$F_n(t) = \frac{F_1(t) - F_0(t) r_2}{r_1 - r_2}(r_1)^n + \frac{F_1(t) - F_0(t) r_1}{r_2 - r_1}(r_2)^n$$

Since $F_0=1$ and $F_1=t$

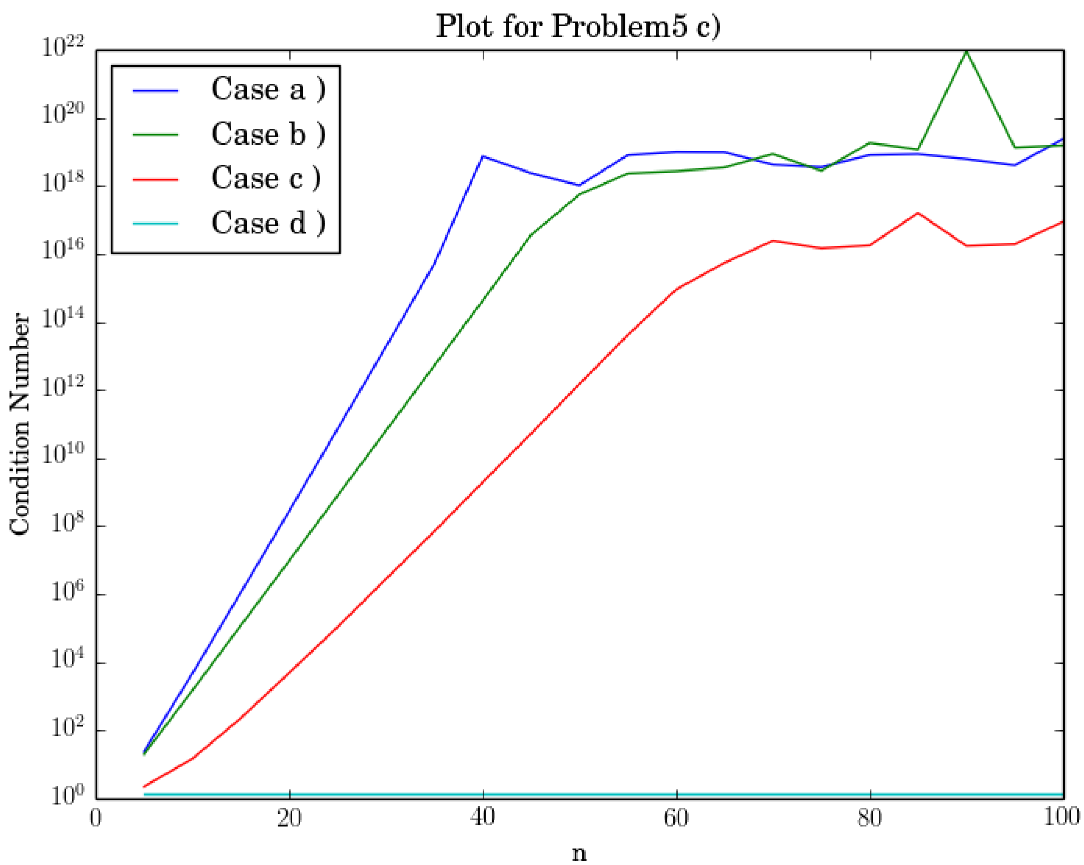
$$F_n(t) = \frac{t - r_2}{r_1 - r_2}(r_1)^n + \frac{t - r_1}{r_2 - r_1}(r_2)^n$$

where r_1 and r_2 are the roots we have above. Then

$$\begin{aligned}
F_n(t) &= \frac{t - t + \sqrt{t^2-1}}{2\sqrt{t^2-1}}(t + \sqrt{t^2 - 1})^n + \frac{t - t - \sqrt{t^2-1}}{-2\sqrt{t^2-1}}(t - \sqrt{t^2 - 1})^n \\
&= \tfrac{1}{2}(t + \sqrt{t^2 - 1})^n + \tfrac{1}{2}(t - \sqrt{t^2 - 1})^n \\
&= \tfrac{1}{2}[(t + \sqrt{t^2 - 1})^n + (t - \sqrt{t^2 - 1})^n] \\
&= \frac{1}{2}\left[\sum_{i=0}^{n}(C_a^b\, t^i(\sqrt{t^2 - 1})^{n-i}) + \sum_{i=0}^{n}(C_a^b\, t^i(-\sqrt{t^2 - 1})^{n-i})\right] \\
&= \frac{1}{2}\left[\sum_{i=0}^{k} C_{2k}^{2\,i}\, t^{2(k-i)}(t^2 - 1)^i\right] \text{ where k is n/2}
\end{aligned}$$

Hence $F_n(t)$ is a polynomial. Here, C_n^i means combination of i out of n.

c) Check problem5_c.py



d) The combination of Chebyshev nodes with the Chebyshev polynomials performs best