CX 4640 Homework 1

Wenqi He

September 7, 2017

1.5

Suppose the changes in input data are Δx and Δy.

$$\begin{aligned}
\text{relative change in input} &= \frac{||(x+\Delta x, y+\Delta y)-(x,y)||}{||(x,y)||} \\
&= \frac{||(\Delta x, \Delta y)||}{||(x,y)||} \\
&= \frac{|\Delta x|+|\Delta y|}{|x|+|y|} \\
&\approx |\Delta x|+|\Delta y|
\end{aligned}$$

$$\begin{aligned}
\text{relative change in output} &= \left|\frac{f(x+\Delta x, y+\Delta y)-f(x,y)}{f(x,y)}\right| \\
&= \left|\frac{[(x+\Delta x)-(y+\Delta y)]-(x-y)}{x-y}\right| \\
&= \frac{|\Delta x-\Delta y|}{|x-y|} \\
&\approx \frac{|\Delta x-\Delta y|}{\epsilon}
\end{aligned}$$

By definition,

$$\text{cond} = \frac{\text{relative change in output}}{\text{relative change in input}}$$

$$\approx \frac{|\Delta x-\Delta y|/\epsilon}{|\Delta x|+|\Delta y|} = \frac{|\Delta x-\Delta y|}{|\Delta x|+|\Delta y|}\cdot\frac{1}{\epsilon} \leq \frac{|\Delta x|+|-\Delta y|}{|\Delta x|+|\Delta y|}\cdot\frac{1}{\epsilon} = \frac{1}{\epsilon}$$

Thus, subtraction is extremely sensitive when ϵ is close to zero.

1.6

(a)

When $x = 0.1$,

$$\begin{aligned}\text{forward error} &= \hat{f}(0.1) - f(0.1) = 0.1 - \sin(0.1) \approx 1.67 \times 10^{-4} \\ \hat{x} &= \arcsin(\hat{f}(0.1)) = \arcsin(0.1) \\ \text{backward error} &= \hat{x} - x = \arcsin(0.1) - 0.1 \approx 1.67 \times 10^{-4}\end{aligned}$$

When $x = 0.5$,

$$\begin{aligned}\text{forward error} &= \hat{f}(0.5) - f(0.5) = 0.5 - \sin(0.5) \approx 2.06 \times 10^{-2} \\ \hat{x} &= \arcsin(\hat{f}(0.5)) = \arcsin(0.5) \\ \text{backward error} &= \hat{x} - x = \arcsin(0.5) - 0.5 \approx 2.36 \times 10^{-2}\end{aligned}$$

When $x = 1.0$,

$$\begin{aligned}\text{forward error} &= \hat{f}(1.0) - f(1.0) = 1.0 - \sin(1.0) \approx 1.59 \times 10^{-1} \\ \hat{x} &= \arcsin(\hat{f}(1.0)) = \arcsin(1.0) \\ \text{backward error} &= \hat{x} - x = \arcsin(1.0) - 1.0 \approx 5.71 \times 10^{-1}\end{aligned}$$

(b)

When $x = 0.1$,

$$\begin{aligned}\text{forward error} &= \hat{f}(0.1) - f(0.1) = (0.1 - 0.1^3/6) - \sin(0.1) \approx -8.33 \times 10^{-8} \\ \hat{x} &= \arcsin(\hat{f}(0.1)) = \arcsin(0.1 - 0.1^3/6) \\ \text{backward error} &= \hat{x} - x = \arcsin(0.1 - 0.1^3/6) - 0.1 \approx -8.37 \times 10^{-8}\end{aligned}$$

When $x = 0.5$,

$$\begin{aligned}\text{forward error} &= \hat{f}(0.5) - f(0.5) = (0.5 - 0.5^3/6) - \sin(0.5) \approx -2.59 \times 10^{-4} \\ \hat{x} &= \arcsin(\hat{f}(0.5)) = \arcsin(0.5 - 0.5^3/6) \\ \text{backward error} &= \hat{x} - x = \arcsin(0.5 - 0.5^3/6) - 0.5 \approx -2.95 \times 10^{-4}\end{aligned}$$

When $x = 1.0$,

$$\begin{aligned}\text{forward error} &= \hat{f}(1.0) - f(1.0) = (1.0 - 1.0^3/6) - \sin(1.0) \approx -8.14 \times 10^{-3} \\ \hat{x} &= \arcsin(\hat{f}(1.0)) = \arcsin(1.0 - 1.0^3/6) \\ \text{backward error} &= \hat{x} - x = \arcsin(1.0 - 1.0^3/6) - 1.0 \approx -1.49 \times 10^{-2}\end{aligned}$$

1.17

If we express x as

$$\pm \left(d_0 + \frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \cdots + \frac{d_{p-1}}{\beta^{p-1}} \right) \beta^E,$$

then since y is adjacent to x,

$$y = \pm \left(d_0 + \frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \cdots + \frac{d_{p-1} \pm 1}{\beta^{p-1}} \right) \beta^E.$$

The spacing between x and y is

$$\frac{1}{\beta^{p-1}} \cdot \beta^E = \beta^{E-p+1},$$

where E is bounded by $[L, U]$.

(a)

The minimum possible spacing is β^{L-p+1}. For single-precision, it's

$$2^{-126-24+1} \approx 1.40 \times 10^{-45}$$

For double-precision, it's

$$2^{-1022-53+1} \approx 4.94 \times 10^{-324}$$

(b)

The maximum possible spacing is β^{U-p+1}. For single-precision, it's

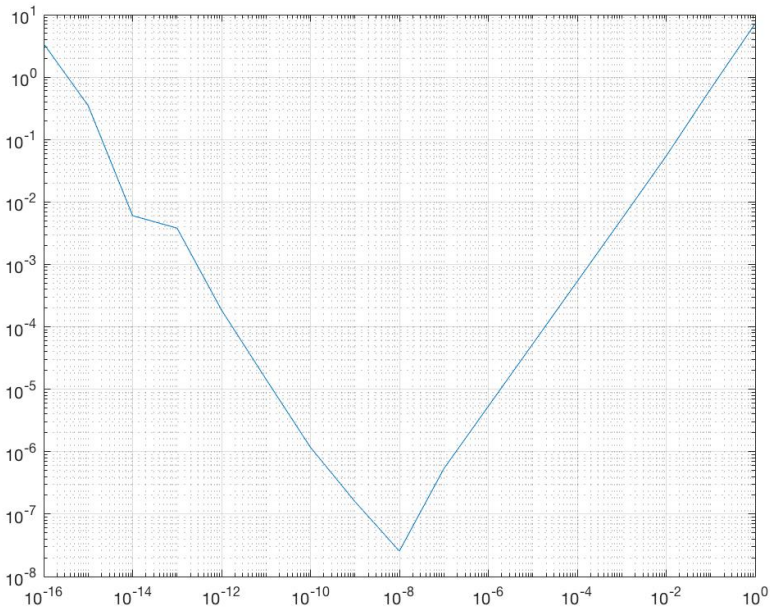
$$2^{127-24+1} \approx 2.03 \times 10^{31}$$

For double-precision, it's

$$2^{1023-53+1} \approx 2.00 \times 10^{292}$$

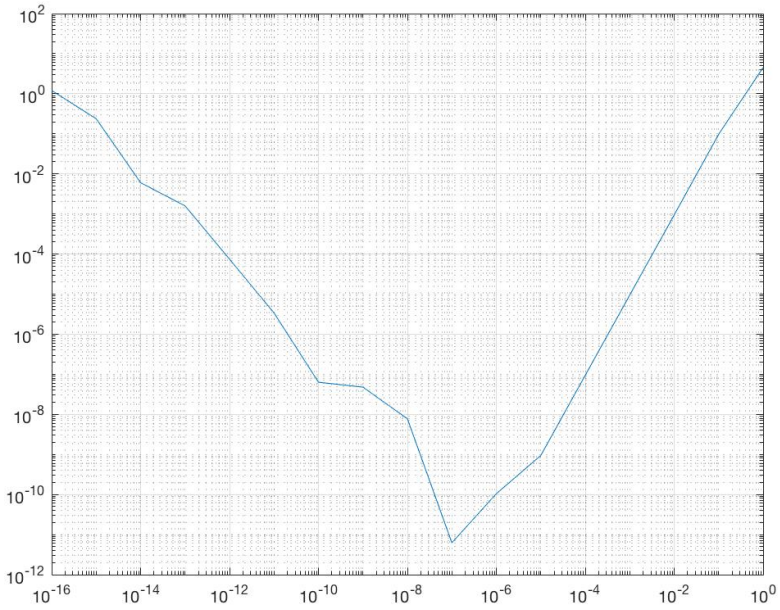
1.7

(a)



The minimum value of the magnitude of the error is approximately 2.5×10^{-8}, and the corresponding h is approximately $10^{-8} = \sqrt{10^{-16}} \approx \sqrt{\epsilon_{mach}}$

(b)



The minimum value of the magnitude of the error is approximately 6×10^{-12}, and the corresponding h is approximately 10^{-7}

Using Taylor expansion,

$$f(x+h) = f(x) + f'(x) \cdot h + \frac{f''(x)}{2} \cdot h^2 + \frac{f'''(c_1)}{3!} \cdot h^3 \quad \text{, for some } c_1 \in [x, x+h]$$

$$\begin{aligned} f(x-h) &= f(x) + f'(x) \cdot (-h) + \frac{f''(x)}{2} \cdot (-h)^2 + \frac{f'''(c_2)}{3!} \cdot (-h)^3 \\ &= f(x) - f'(x) \cdot h + \frac{f''(x)}{2} \cdot h^2 - \frac{f'''(c_2)}{3!} \cdot h^3 \quad \text{, for some } c_2 \in [x-h, x], \end{aligned}$$

$$\begin{aligned} f(x+h) - f(x-h) &= \left(f(x) + f'(x) \cdot h + \frac{f''(x)}{2} \cdot h^2 + \frac{f'''(c_1)}{3!} \cdot h^3 \right) \\ &\quad - \left(f(x) - f'(x) \cdot h + \frac{f''(x)}{2} \cdot h^2 - \frac{f'''(c_2)}{3!} \cdot h^3 \right) \\ &= 2f'(x) \cdot h + \frac{f'''(c_1) + f'''(c_2)}{6} \cdot h^3 \end{aligned}$$

$$\frac{f(x+h)-f(x-h)}{2h} = f'(x) + \frac{f'''(c_1)+f'''(c_2)}{12}\cdot h^2$$
$$\frac{f(x+h)-f(x-h)}{2h} - f'(x) = \frac{f'''(c_1)+f'''(c_2)}{12}\cdot h^2$$

Suppose $f'''(x) \le M$ for $x \in [x-h, x+h]$, then:

$$\left|\frac{f(x+h)-f(x-h)}{2h} - f'(x)\right| = \frac{|f'''(c_1)+f'''(c_2)|}{12}\cdot h^2 \le \frac{|f'''(c_1)|+|f'''(c_2)|}{12}\cdot h^2 \le \frac{2M}{12}\cdot h^2 = \frac{Mh^2}{6}$$

The upperbound for truncation error is $\frac{Mh^2}{6}$.
Suppose the errors in function values are bounded by ϵ, that is

$$\left|\hat{f}(x) - f(x)\right| = \delta \le \epsilon, \text{ for all } x$$

Then,

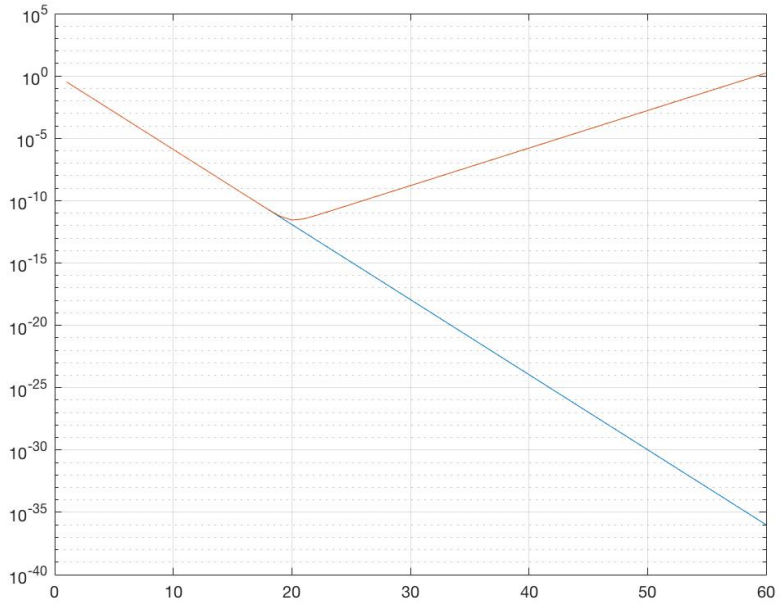
$$\begin{aligned}
&\left|\frac{\hat{f}(x+h)-\hat{f}(x-h)}{2h} - \frac{f(x+h)-f(x-h)}{2h}\right| \\
&= \frac{\left|\left(\hat{f}(x+h)-f(x+h)\right) - \left(\hat{f}(x-h)-f(x-h)\right)\right|}{2h} \\
&\le \frac{\left|\hat{f}(x+h)-f(x+h)\right| + \left|-\left(\hat{f}(x-h)-f(x-h)\right)\right|}{2h} \\
&= \frac{\left|\hat{f}(x+h)-f(x+h)\right| + \left|\hat{f}(x-h)-f(x-h)\right|}{2h} \\
&= \frac{\delta_1+\delta_2}{2h} \le \frac{2\epsilon}{2h} = \frac{\epsilon}{h}
\end{aligned}$$

The upperbound for rounding error is $\frac{\epsilon}{h}$.
The total computational error bound is therefore $\frac{Mh^2}{6} + \frac{\epsilon}{h}$
The minimum magnitude of error occurs when

$$\left(\frac{Mh^2}{6} + \frac{\epsilon}{h}\right)' = \frac{Mh}{3} - \frac{\epsilon}{h^2} = 0$$

$$h = \sqrt[3]{\frac{3\epsilon}{M}}$$

1.17



The graph exhibits expected behavior for small k's, however, after k reaches 20, the sequence suddenly starts to increase.

Let A be the advance operator, then the equation can be written as

$$A^2 x_k - 2.25 A x_k + 0.5 x_k = 0$$

The characteristic equation is

$$\lambda^2 - 2.25\lambda + 0.5 = 0$$

$$\lambda_1 = 2, \lambda_2 = \frac{1}{4}$$

The general solution to the difference equation is

$$x_k = c_1 \cdot 2^k + c_2 \cdot \left(\frac{1}{4}\right)^k$$

The absolute value of the first term increases and the second term decreases as k grows larger. For the particular initial condition specified in this problem,

$$c_1 = 0, \quad c_2 = \frac{4}{3},$$

there is no contribution from the first term, therefore the sequence converges to 0. However this initial condition is very unstable, as c_1 would become non-zero even for the slightest perturbations resulted from machine errors. Then the sequence no longer converges to 0, and the first term would explode when k grows larger, which explains the unexpected behavior described above.

CX 4640 Homework 2

Wenqi He

September 25, 2017

2.7

(a)

$$\begin{aligned}\det A &= \det \begin{bmatrix} 1 & 1+\epsilon \\ 1-\epsilon & 1 \end{bmatrix} \\ &= 1 - (1-\epsilon)(1+\epsilon) \\ &= 1 - (1-\epsilon^2) \\ &= \epsilon^2\end{aligned}$$

(b)

The smallest non-negative number representable in a normalized single-precision system is 1×2^{-126}. Therefore the computed result for determinant would be zero if

$$\epsilon^2 < 2^{-126}$$
$$|\epsilon| < 2^{-63}$$
$$-2^{-63} < \epsilon < 2^{63}$$

In a double-precision system,

$$\epsilon^2 < 2^{-1022}$$
$$-2^{-511} < \epsilon < 2^{511}$$

(c)

$$A = \begin{bmatrix} 1 & 1+\epsilon \\ 1-\epsilon & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1-\epsilon & 1 \end{bmatrix} \begin{bmatrix} 1 & 1+\epsilon \\ 0 & \epsilon^2 \end{bmatrix}$$

(d)

$$\det U = 1 \cdot \epsilon^2 = \epsilon^2$$

The matrix is singular when the computed value of ϵ^2 equals zero, that is, ϵ^2 is smaller than the smallest representable number. So the answer is the same as (b).

2.21

$$\begin{aligned} \boldsymbol{x} &= B^{-1}(2A+I)(C^{-1}+A)\boldsymbol{b} \\ B\boldsymbol{x} &= (2A+I)(C^{-1}+A)\boldsymbol{b} \\ &= 2AC^{-1}\boldsymbol{b} + 2A^2\boldsymbol{b} + C^{-1}\boldsymbol{b} + A\boldsymbol{b} \end{aligned}$$

Let $\boldsymbol{y} = C^{-1}\boldsymbol{b}$, then

$$C\boldsymbol{y} = \boldsymbol{b}$$

$$B\boldsymbol{x} = 2A\boldsymbol{y} + 2A^2\boldsymbol{b} + \boldsymbol{y} + A\boldsymbol{b}$$

Using Gaussian Elimination, one can solve the first equation for $\boldsymbol{y}$, and then solve the second one for $\boldsymbol{x}$ without computing the inverses of B and C.

The MATLAB code is included in `SolveForX.m`

2.26

(a)

One can compute the inverse of A using Sherman-Morrison formula:

$$\begin{aligned} A^{-1} &= (I - uv^T)^{-1} \\ &= I^{-1} + I^{-1}u(1 - v^T I^{-1}u)^{-1}v^T I^{-1} \\ &= I + u(1 - v^T u)^{-1}v^T, \end{aligned}$$

provided that $1 - v^T u$ is invertible, that is,

$$v^T u \neq 1$$

(b)

From (a), $\sigma = -(1 - v^T u)^{-1}$

(c)

Yes.

$$M_k = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -m_{k+1} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -m_n & 0 & \cdots & 1 \end{bmatrix} = I - \begin{bmatrix} 0 \\ \vdots \\ m_{k+1} \\ \vdots \\ m_n \end{bmatrix} \boldsymbol{e_k}^T,$$

$$u = \begin{bmatrix} 0 \\ \vdots \\ m_{k+1} \\ \vdots \\ m_n \end{bmatrix}, \quad v = \boldsymbol{e}_k, \quad \sigma = -\left(1 - \boldsymbol{e}_k{}^T \begin{bmatrix} 0 \\ \vdots \\ m_{k+1} \\ \vdots \\ m_n \end{bmatrix} \right)^{-1} = -(1-0)^{-1} = -1$$

2.4

$\|A_1^{-1}\| = 0.7097, \quad \text{cond}(A_1) = 12.7742, \quad \|A_2^{-1}\| = 1.6393e{+}04, \quad \text{cond}(A_2) = 4.0163e{+}06$

Using the first approach, the estimations are:

$\|A_1^{-1}\| = 0.6226, \quad \text{cond}(A_1) = 11.2071, \quad \|A_2^{-1}\| = 1.3082e{+}04, \quad \text{cond}(A_2) = 3.2051e{+}06$

Using the second approach (The values might vary):

$\|A_1^{-1}\| = 0.6013, \quad \text{cond}(A_1) = 11.1528, \quad \|A_2^{-1}\| = 4.4131e{+}03, \quad \text{cond}(A_2) = 1.9666e{+}06$

Apparently, the first approach is more accurate.

5

(Run `CreateTable` to reproduce the result.)

n	relative error	condition number
2	2.8951e-16	19.281
3	3.4571e-15	524.06
4	9.1928e-14	15514
5	8.4248e-14	4.7661e+05
6	1.0052e-10	1.4951e+07
7	3.3589e-09	4.7537e+08
8	1.1118e-08	1.5258e+10
9	3.2899e-06	4.9315e+11
10	0.00010306	1.6025e+13
11	0.0024921	5.2202e+14
12	0.068797	1.6212e+16

CX 4640 Homework 3

Wenqi He

October 15, 2017

3.5

(a)

The vector is obviously within $span(A)$ and therefore cannot be the residue.

(b)

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \end{bmatrix} \neq 0$$

The vector is not orthogonal to $span(A)$, so it cannot be the residue.

(c)

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = 0$$

The vector is orthogonal to $span(A)$, therefore it is a possible value for r

3.28

(a)

For all $i \neq j$,

$$P_i P_j = q_i q_i^T q_j q_j^T = q_i (q_i^T q_j) q_j^T = 0$$

Base case:

$$(I - P_2)(I - P_1) = I - P_1 - P_2 + P_2 P_1 = I - P_2 - P_1$$

Inductive Step:
Suppose $(I - P_n)(I - P_{n-1}) \cdots (I - P_1) = I - P_n - \cdots - P_1$, then

$$\begin{aligned}
&(I - P_{n+1})(I - P_n)(I - P_{n-1}) \cdots (I - P_1) \\
&= (I - P_{n+1})(I - P_n - \cdots - P_1) \\
&= (I - P_n - \cdots - P_1) - P_{n+1}(I - P_n - \cdots - P_1) \\
&= I - P_{n+1} - P_n - \cdots - P_1 + \sum_{i=1}^{n} (P_{n+1} P_i) \\
&= I - P_{n+1} - P_n - \cdots - P_1
\end{aligned}$$

(b)

In the classical Gram-Schmidt procedure, during the i-th iteration, $P_i a_k$ is subtracted away, therefore the process is equivalent to

$$\begin{aligned}
q_k &= v_{k-2} - P_{k-1} a_k \\
&= v_{k-3} - P_{k-2} a_k - P_{k-1} a_k \\
&\cdots \\
&= a_k - P_1 a_k - \cdots - P_{k-1} a_k \\
&= (I - (P_1 + \cdots + P_{k-1})) a_k
\end{aligned}$$

(c)

In the modified Gram-Schmidt procedure, during the i-th iteration, $P_i v_{i-1}$ is subtracted away, where v_{i-1} is the intermediate result from last iteration, therefore the process is equivalent to

$$\begin{aligned}
q_k &= (I - P_{k-1}) v_{k-2} \\
&= (1 - P_{k-1})(1 - P_{k-2}) v_{k-3} \\
&\cdots \\
&= (I - P_{k-1}) \cdots (1 - P_1) a_k
\end{aligned}$$

(d)

It's already shown in (a) that (b) and (c) are equivalent. As for (d), take $m = 2$

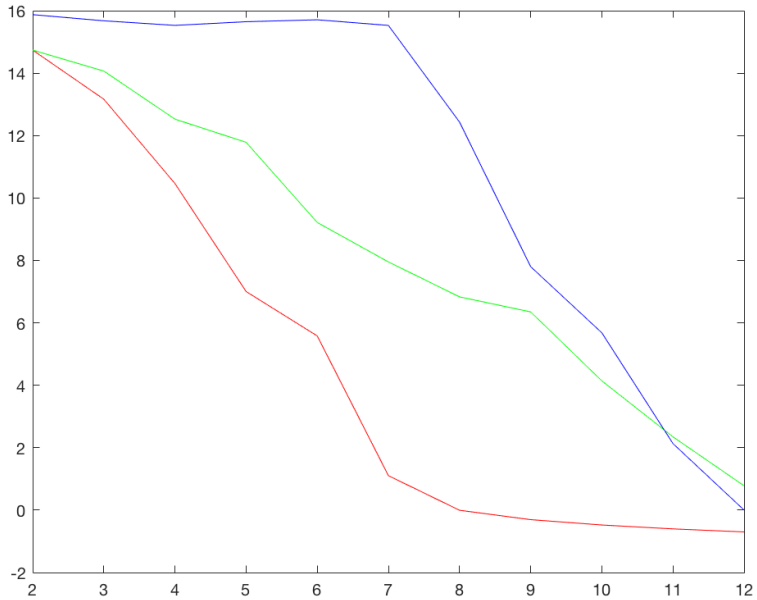
$$\begin{aligned}
&(I - (P_1 + \cdots + P_{k-1}))^2 \\
&= \left(I - \sum_{i=1}^{k-1} P_i\right)\left(I - \sum_{i=1}^{k-1} P_i\right) \\
&= I - \sum_{i=1}^{k-1} P_i - \left(\sum_{i=1}^{k-1} P_i - \sum_{i=1}^{k-1}\sum_{j=1}^{k-1} P_i P_j\right) \\
&= I - \sum_{i=1}^{k-1} P_i - \left(\sum_{i=1}^{k-1} P_i - \sum_{i=1}^{k-1} P_i^2\right)
\end{aligned}$$

Since $P_i^2 = q_i q_i^T q_i q_i^T = q_i (q_i^T q_i) q_i^T = q_i q_i^T = P_i$, the last term evaluates to zero

$$(I - (P_1 + \cdots + P_{k-1}))^2 = I - \sum_{i=1}^{k-1} P_i = I - (P_1 + \cdots + P_{k-1})$$

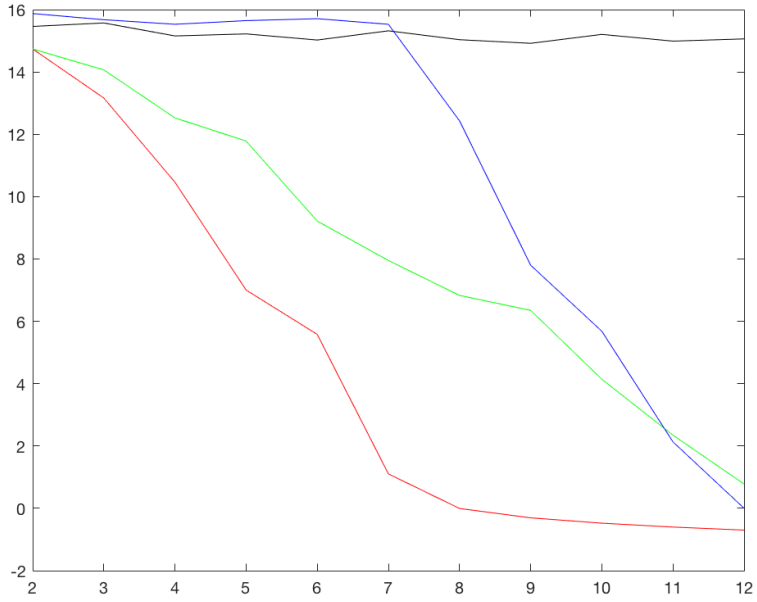
Therefore all three methods are equivalent.

3.12



All three methods become less and less accurate as n grows. Classical Gram-

Schmidt is the least accurate, but applying it twice makes it more accurate than modified Gram-Schmidt, although it takes twice as much time.



In contrast with the three variations of Gram-Schmidt procedure, the Householder method remains accurate as n grows.

4.3

(a)

$$(1-\lambda)^2 - 4$$

(b)

$$\lambda_1 = -1, \quad \lambda_2 = 3$$

(c)

Same as (b)

(d)

For $\lambda_1 = -1$

$$\begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} x = 0, \quad x_{\lambda_1} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

For $\lambda_2 = 3$

$$\begin{bmatrix} -2 & 4 \\ 1 & -2 \end{bmatrix} x = 0, \quad x_{\lambda_2} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

(e)

$$Ax_0 = \begin{bmatrix} 1 & 4 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

(f)

It will converge to $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$, the eigenvector corresponding to the dominant eigenvalue $\lambda = 3$.

(g)

$$\lambda \approx \frac{x^T A x}{x^T x} = \frac{\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}}{\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}} = \frac{7}{2} = 3.5$$

(h)

The eigenvalues of A^{-1} are

$$\lambda_1' = \frac{1}{\lambda_1} = -1, \quad \lambda_2' = \frac{1}{\lambda_2} = 1/3$$

The inverse iteration would converge to the eigenvector corresponding to the dominant eigenvalue λ_1', which is

$$x_{\lambda_1'} = x_{\lambda_1} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

(i)

The eigenvalues of $A - \sigma I$ are

$$\lambda_1' = \lambda_1 - \sigma = -3, \quad \lambda_2' = \lambda_2 - \sigma = 1$$

The eigenvalues of $(A - \sigma I)^{-1}$ are

$$\lambda_1'' = \frac{1}{\lambda_1'} = -\frac{1}{3}, \quad \lambda_2'' = \frac{1}{\lambda_2''} = 1$$

The dominant eigenvalue of $(A - \sigma I)^{-1}$, λ_2'', would be obtained from the shifted inverse iteration, which translates to

$$\lambda_2 = 3$$

of the original matrix A

(j)

Since A is not symmetric, it would just converge to triangular form.

4.24

(a)

Since the matrix is of rank one, the columns must be multiples of some real vector u, that is:

$$A = \begin{bmatrix} u \cdot v_1 & u \cdot v_2 & \cdots & u \cdot v_n \end{bmatrix} = u \begin{bmatrix} v_1 & v_2 & \cdots, v_n \end{bmatrix} = uv^T$$

(b)

Multiply A by u on the right:

$$Au = (uv^T)u = u(v^T u) = u(u^T v)$$

Therefore, $u^T v$ is an eigenvalue corresponding to eigenvector u

(c)

The other eigenvalue is 0.
proof: By rank-nullity theorem, A has nullity of $n-1$. For any vector x in the null space, $Ax = 0 = 0x$, which means that 0 is an eigenvalue with multiplicity $n-1$. By spectral theorem, there are no more eigenvalues.

(d)

It only takes 1 iteration.
proof: Any vector can be expressed as a linear combination of the eigenvectors:

$$x = \sum_i c_i x_i = cu + \sum_{x_i \in Null(A)} c_i x_i,$$

After the first iteration:

$$\begin{aligned} Ax &= A\left(cu + \sum_{x_i \in Null(A)} c_i x_i\right) \\ &= A(cu) + 0 = (c\lambda)u \end{aligned}$$

4.12

(a)

$$x^{(3)} = A^3 x^{(0)} = \begin{bmatrix} 0.8 & 0.2 & 0.1 \\ 0.1 & 0.7 & 0.3 \\ 0.1 & 0.1 & 0.6 \end{bmatrix}^3 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.587 \\ 0.238 \\ 0.175 \end{bmatrix}$$

(b)

The long-term value must satisfies

$$Ax^{(\infty)} = x^{(\infty)}$$

$$(A - I)x^{(\infty)} = 0$$

$$\begin{bmatrix} -0.2 & 0.2 & 0.1 \\ 0.1 & -0.3 & 0.3 \\ 0.1 & 0.1 & -0.4 \end{bmatrix} x^{(\infty)} = 0$$

$$x_1^{(\infty)} = 2.25x_3, \quad x_2^{(\infty)} = 1.75x_3$$

$$x^{(\infty)} = \frac{1}{2.25 + 1.75 + 1} \begin{bmatrix} 2.25 \\ 1.75 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.45 \\ 0.35 \\ 0.2 \end{bmatrix}$$

(c)

No. The equation above has only one solution, therefore all transitions will converge to the same vector regardless of the initial condition.

(d)

Using MATALB to evaluate A^k when k is large:

$$A^{100} = \begin{bmatrix} 0.45 & 0.45 & 0.45 \\ 0.35 & 0.35 & 0.35 \\ 0.2 & 0.2 & 0.2 \end{bmatrix}, \quad A^{200} = \begin{bmatrix} 0.45 & 0.45 & 0.45 \\ 0.35 & 0.35 & 0.35 \\ 0.2 & 0.2 & 0.2 \end{bmatrix}$$

From the result, it can be inferred that

$$\lim_{k \to \infty} A^k = \begin{bmatrix} 0.45 & 0.45 & 0.45 \\ 0.35 & 0.35 & 0.35 \\ 0.2 & 0.2 & 0.2 \end{bmatrix}$$

The rank of this matrix is 1.

(e)

The eigenvalues and corresponding eigenvectors (unnormalized) of A are:

$$\lambda_1 = 1, \quad v_1 = \begin{bmatrix} -0.7448 \\ -0.5793 \\ -0.3310 \end{bmatrix}$$

$$\lambda_2 = 0.6, \quad v_2 = \begin{bmatrix} -0.7071 \\ 0.7071 \\ 0 \end{bmatrix}$$

$$\lambda_3 = 0.5, \quad v_3 = \begin{bmatrix} 0.4082 \\ -0.8165 \\ 0.4082 \end{bmatrix}$$

The transitions are equivalent to applying power iteration to the initial vector, which converges to to the normalized eigenvector corresponding to the dominant eigenvalue $\lambda_1 = 1$:

$$v_1 = \frac{1}{-0.7448 - 0.5793 - 0.3310} \begin{bmatrix} -0.7448 \\ -0.5793 \\ -0.3310 \end{bmatrix} = \begin{bmatrix} 0.45 \\ 0.35 \\ 0.2 \end{bmatrix}$$

Therefore, the long-term effect of the Markov chain is just converting any initial vectors to v_1, that is:

$$\lim_{k \to \infty} A^k = \begin{bmatrix} v_1 & v_1 & v_1 \end{bmatrix}$$

(f)

No. If the Markov chain doesn't have a steady-state, that is, if the equation in (b) has no solution, then 1 is not a eigenvalue of A.

(g)

A probability distribution vector that satisfies $Ax = x$ is simply a normalized eigenvector corresponding to eigenvalue 1 .

(h)

By directly solving $(A - I)x = 0$ and then normalize the result.

CX 4640 Homework 4

Wenqi He

November 5, 2017

1

(a)

$$g_1'(x) = 1 - 2x$$

The iteration is locally convergent when $|1 - 2x| < 1$, or equivalently, $0 < x < 1$, Conversely, when $x \leq 0$ or $x \geq 1$, the iteration is locally divergent.

(b)

$$g_2'(x) = 1 - \frac{2}{3}x$$

Following the same reasoning as (a), when $0 < x < 3$ the iteration is locally convergent and when $x \leq 0$ or $x \geq 3$, the iteration is locally divergent.

(c)

$$f'(x) = 2x$$

The function is given by

$$g(x) = x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - y}{2x} = \frac{1}{2}\left(x + \frac{y}{x}\right)$$

2

(a)

The iteration function of this scheme is

$$g(x) = x - f(x)/d$$

It's locally convergent when

$$|g'(x^*)| = |1 - f'(x^*)/d| < 1$$

(b)

$$g'(x) = 1 - f'(x)/d$$

$$x_{n+1} = g(x_n) = r + g'(r)(x_n - r) + \frac{g''(c)}{2}(x_n - r)^2$$

$$x_{n+1} - r = g'(r)(x_n - r) + \frac{g''(c)}{2}(x_n - r)^2$$

$$\frac{x_{n+1} - r}{x_n - r} = g'(r) + \frac{g''(c)}{2}(x_n - r)$$

$$\lim_{n\to\infty} \frac{x_{n+1} - r}{x_n - r} = g'(r) + 0 = g'(r) = 1 - f'(r)/d$$

In general, the convergence rate is $1 - f'(r)/d$, where r is the fixed point

(c)

To achieve quadratic convergence, we can set $g'(r) = 0$.
Proof: when $g'(r) = 0$,

$$\begin{aligned} x_{n+1} &= g(x_n) \\ &= r + g'(r)(x_n - r) + \frac{g''(r)}{2}(x_n - r)^2 + \frac{g'''(c)}{6}(x_n - r)^3 \\ &= r + \frac{g''(r)}{2}(x_n - r)^2 + \frac{g'''(c)}{6}(x_n - r)^3 \end{aligned}$$

$$\frac{x_{n+1} - r}{(x_n - r)^2} = \frac{g''(r)}{2} + \frac{g'''(c)}{6}(x_n - r)$$

$$\lim_{n\to\infty} \frac{x_{n+1} - r}{(x_n - r)^2} = \frac{g''(r)}{2}$$

Therefore d needs to satisfy

$$g'(r) = 1 - f'(r)/d = 0$$

$$d = f'(r)$$

3

(a)

(1)

$$\begin{aligned} LHS &= R_{k+1} \\ &= I - AX_{k+1} \\ &= I - A(X_k + X_k(I - AX_k)) \\ &= I - A(2X_k - X_kAX_k) \\ &= I - 2AX_k + (AX_k)^2 \end{aligned}$$

$$\begin{aligned} RHS &= R_k^2 \\ &= (I - AX_k)^2 \\ &= I - 2AX_k + (AX_k)^2 \end{aligned}$$

Therefore

$$R_{k+1} = R_k^2$$

(2)

$$\begin{aligned} LHS &= E_{k+1} \\ &= A^{-1} - X_{k+1} \\ &= A^{-1} - (X_k + X_k(I - AX_k)) \\ &= A^{-1} - (2X_k - X_kAX_k) \\ &= A^{-1} - 2X_k + X_kAX_k) \end{aligned}$$

$$\begin{aligned} RHS &= E_kAE_k \\ &= (A^{-1} - X_k)A(A^{-1} - X_k) \\ &= (A^{-1} - X_k)(I - AX_k) \\ &= A^{-1} - 2X_k + X_kAX_k \end{aligned}$$

Therefore

$$E_{k+1} = E_kAE_k$$

(b)

See `MyInverse.m`

4

$$f'(x) = 2n \cdot x^{2n-1}$$

The iteration scheme is

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^{2n} - a^n}{2n \cdot x_k^{2n-1}}$$

For $n = 1$,

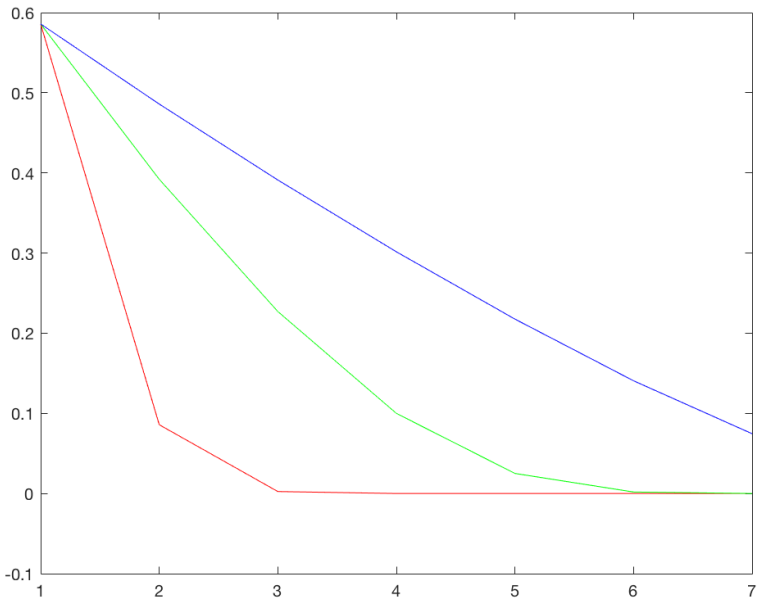
$$x_{k+1} = x_k - \frac{x_k^2 - a}{2x_k}$$

For $n = 5$,

$$x_{k+1} = x_k - \frac{x_k^{10} - a^5}{10x_k^9}$$

For $n = 10$,

$$x_{k+1} = x_k t - \frac{x_k^{20} - a^{10}}{20x_k^{19}}$$



The smaller n is, the faster the iteration converges.

5

See `Newton.m`

6

See `Symmetrize.m`, `Diagonalize.m` and `MySVD.m`

CX 4640 Homework 5

Wenqi He

December 4, 2017

1

1.1 first-order optimality condition

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} 2x_1 - 2 \\ 2x_2 \\ -2x_3 + 4 \end{pmatrix}, \quad \nabla f(\boldsymbol{x}^*) = \begin{pmatrix} 2 \times 2.5 - 2 \\ 2 \times -1.5 \\ -2 \times -1 + 4 \end{pmatrix} = \begin{pmatrix} 3 \\ -3 \\ 6 \end{pmatrix}$$

$$\boldsymbol{J}_g^T = \nabla g = \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}$$

The first-order optimality condition is

$$\nabla f(\boldsymbol{x}^*) + \boldsymbol{J}_g^T(\boldsymbol{x}^*)\boldsymbol{\lambda} = \begin{pmatrix} 3 \\ -3 \\ 6 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix} \lambda = 0$$

$\lambda^* = -3$ satisfies the condition.

1.2 second-order optimality condition

$$\boldsymbol{H}_f = \boldsymbol{J}_{\nabla f} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{pmatrix}$$

$$\boldsymbol{H}_g = \boldsymbol{J}_{\nabla g} = \boldsymbol{0}$$

$$\boldsymbol{B}(\boldsymbol{x}^*, \lambda^*) = \boldsymbol{H}_f + \lambda \boldsymbol{H}_g = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{pmatrix} + \boldsymbol{0} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{pmatrix}$$

Now find the null space of $\boldsymbol{J}_g$:

$$\boldsymbol{J}_g \boldsymbol{x} = \begin{pmatrix} 1 & -1 & 2 \end{pmatrix} \boldsymbol{x} = 0$$

The solution is $x_1 = x_2 - 2x_3$, where x_2 and x_3 are free variables. Now we can construct a $\boldsymbol{Z}$ whose column space is the null space of $\boldsymbol{J}_g$:

$$\boldsymbol{Z} = \begin{pmatrix} 1 & -2 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\boldsymbol{B}^T\boldsymbol{Z}\boldsymbol{B} = \begin{pmatrix} 1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} 1 & -2 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 4 & -4 \\ -4 & 6 \end{pmatrix}$$

$$\boldsymbol{v}^T\boldsymbol{B}^T\boldsymbol{Z}\boldsymbol{B}\boldsymbol{v} = 4v_1^2 - 8v_1v_2 + 6v_2^2 = (2v_1 - 2v_2)^2 + 2v_2^2 \geq 0$$

Since $\boldsymbol{B}^T\boldsymbol{Z}\boldsymbol{B}$ is positive definite, the point $(\boldsymbol{x}^*, \lambda^*)$ satisfies the second-order optimality condition.

2

(a)

$$\begin{aligned} \nabla f(\boldsymbol{x}) &= \frac{1}{2}\partial^k(x_iA_j^ix^j) - \partial^k(x_ib^i) + \partial^k c \\ &= \frac{1}{2}\big((\partial^k x_i)A_j^ix^j + x_iA_j^i(\partial^k x_j)\big) - (\partial^k x_i)b^i \\ &= \frac{1}{2}\big((\partial^k x_i)A_j^ix^j + (\partial^k x_j)A_i^jx^i\big) - (\partial^k x_i)b^i \\ &= \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \end{aligned}$$

$$\boldsymbol{H}_f(\boldsymbol{x}) = \boldsymbol{J}_{\nabla f} = \partial_k(A_j^ix^j) + \partial_k b^i = A_j^i(\partial_k x^j) = \boldsymbol{A}$$

Using Newton's method:

$$\boldsymbol{H}_f(\boldsymbol{x}_0)\boldsymbol{s}_0 = -\nabla f(\boldsymbol{x}_0)$$

$$\boldsymbol{A}\boldsymbol{s}_0 = -\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{b}$$

After the first iteration:

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{s}_0$$

$$\begin{aligned} \nabla f(\boldsymbol{x}_1) &= \boldsymbol{A}\boldsymbol{x}_1 - \boldsymbol{b} = \boldsymbol{A}(\boldsymbol{x}_0 + \boldsymbol{s}_0) - \boldsymbol{b} \\ &= \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{A}\boldsymbol{s}_0 - \boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_0 - \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{b} - \boldsymbol{b} \\ &= \boldsymbol{0} \end{aligned}$$

(b)

Using the steepest descent method,

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - \alpha \nabla f(\boldsymbol{x}_0),$$

where α minimizes $f(\boldsymbol{x})$ along the direction of negative gradient. From (a),

$$\nabla f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$$

The fact that $\boldsymbol{x}^*$ is the solution means that

$$\nabla f(\boldsymbol{x}^*) = \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{b} = \boldsymbol{0}$$

The fact that $\boldsymbol{x}_0 - \boldsymbol{x}^*$ is an eigenvector of $\boldsymbol{A}$ means that there exists some λ such that

$$\boldsymbol{A}(\boldsymbol{x}_0 - \boldsymbol{x}^*) = \lambda(\boldsymbol{x}_0 - \boldsymbol{x}^*)$$
$$\boldsymbol{A}\boldsymbol{x}_0 = \lambda(\boldsymbol{x}_0 - \boldsymbol{x}^*) + \boldsymbol{A}\boldsymbol{x}^*$$

Plug this result in the update function:

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - \alpha(\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{b})$$
$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - \alpha\big(\lambda(\boldsymbol{x}_0 - \boldsymbol{x}^*) + \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{b}\big)$$
$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - \alpha\lambda(\boldsymbol{x}_0 - \boldsymbol{x}^*)$$

When $\alpha = \lambda^{-1}$, f reaches a critical point,

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - (\boldsymbol{x}_0 - \boldsymbol{x}^*) = \boldsymbol{x}^*$$
$$\nabla f(\boldsymbol{x}_1) = \boldsymbol{0}$$

Therefore the method with the given starting point converges in one iteration.

3

(a)

$$\nabla f = \begin{pmatrix} 2x \\ 2y \end{pmatrix}, \quad \nabla g = \begin{pmatrix} -3(x-1)^2 \\ 2y \end{pmatrix}$$

Using the method of Lagrange multiplier, we need to solve the equation:

$$\nabla f + \lambda \nabla g = \boldsymbol{0}$$

Or:

$$\begin{aligned} 2x - 3\lambda(x-1)^2 &= 0 \\ 2y + 2\lambda y &= 0 \\ y^2 - (x-1)^3 &= 0 \end{aligned}$$

From the second equation, we must have either $y = 0$ or $\lambda = -1$. If $y = 0$, from the third equation we can conclude that $x = 1$, however this result does not satisfy the first equation. On the other hand, if $\lambda = -1$, the first equation becomes

$$3x^2 - 4x + 3 = 0,$$

which does not have real solutions. Therefore this problem cannot be solved using Lagrange multipliers.

(b)

The penalty function is

$$\phi_\rho(x, y) = x^2 + y^2 + \frac{1}{2}\rho\left(y^2 - (x-1)^3\right)^2$$

$$\nabla\phi_\rho(x, y) = \begin{pmatrix} 2x - 3\rho(y^2 - (x-1)^3)(x-1)^2 \\ 2y + 2\rho(y^2 - (x-1)^3)y \end{pmatrix} = \mathbf{0}$$

From the second equation, if $y \neq 0$

$$\rho(y^2 - (x-1)^3) = -1$$

The first equation $2x + 3(x-1)^2 = 0$ does not have real solutions, therefore y must be 0. Then we have:

$$2x + 3\rho(x-1)^5 = 0$$

$$\lim_{\rho\to\infty}(x-1)^5 = \lim_{\rho\to\infty} -\frac{2x}{3\rho} = 0$$

$$x = 1$$

So $(1, 0)$ is the minimizer.

4

(a)

There are 5 edges, so there must be 5 vertices. They are:

$$\left(\frac{12}{11}, \frac{6}{11}\right), \left(\frac{6}{5}, 0\right), \left(\frac{6}{7}, \frac{6}{7}\right), \left(0, \frac{3}{2}\right), (0, 0)$$

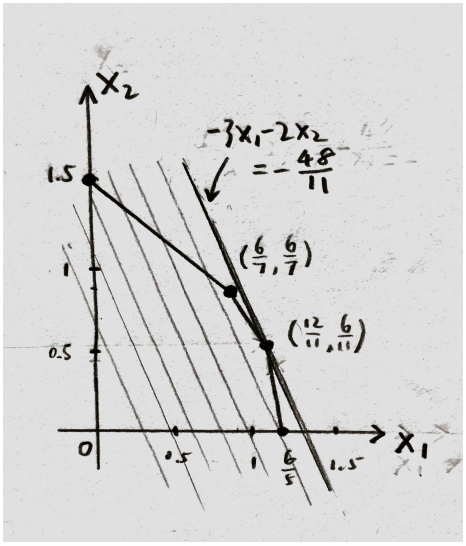
(b)

$$f(\frac{12}{11}, \frac{6}{11}) = -\frac{48}{11}, \quad f(\frac{6}{5}, 0) = -\frac{18}{5}$$

$$f(\frac{6}{7}, \frac{6}{7}) = -\frac{30}{7}, \quad f(0, \frac{3}{2}) = -3, \quad f(0, 0) = 0$$

From the above results, we can see that $(\frac{12}{11}, \frac{6}{11})$ is the minimizer.

(c)



5

(a)

See `mygn.m`
Using Gauss-Newton method, the least squares solution is

$$x_1 = 14.3766$$

$$x_2 = -1.5139$$

.

(b)

See `linear.m`
The result obtained by linear least squares method is

$$x_1 = 8.6350$$

$$x_2 = -1.0967$$

which is different from that of part (a) because the objective here is to minimize

$$\sum_i \big(\log(y_i) - \log(x_1) - x_2 t_i\big)^2 = \sum_i \log^2\left(\frac{y_i}{x_1 e^{x_2 t_i}}\right) = \sum_i \log^2\left(\frac{y_i}{f(t_i, \boldsymbol{x})}\right)$$

whereas Gauss-Newton method in part (a) minimizes

$$\sum_i \big(y_i - f(t_i, \boldsymbol{x})\big)^2$$

6

(a)

See `ueig.m`

(b)

See `ceig.m`

7

(a)

See `qlopt.m`

(b)

See `qnopt.m`

(c)

See `nnopt.m`