第一题

1. $Ax=b$ $A(x+\delta x)=b+\delta b$. where $\begin{bmatrix}610 & 987\\ 987 & 1597\end{bmatrix}=A$

(1) $\|\delta b\|_\infty/\|b\|_\infty$ is very small while $\|\delta x\|_\infty/\|x\|_\infty$ is very big.

对于增广矩阵 $\left[\begin{array}{cc|c}610 & 987 & b_1\\ 987 & 1597 & b_2\end{array}\right] \rightarrow \left[\begin{array}{cc|c}610 & 987 & b_1\\ 0 & \varepsilon & b_2-\frac{987}{610}b_1\end{array}\right]$

其中 $\varepsilon = 1597-987^2/610 \approx 10^{-3}$

解得 $x_1=\left(\frac{987^2}{610^2\varepsilon}+\frac{1}{610}\right)b_1-\frac{987}{610\varepsilon}b_2$.

$x_2=-\frac{987}{610\varepsilon}b_1+\frac{1}{\varepsilon}b_2$

同理有 $\delta x_1=\left(\frac{987^2}{610^2\varepsilon}+\frac{1}{610}\right)\delta b_1-\frac{987}{610\varepsilon}\delta b_2$

$\delta x_2=-\frac{987}{610\varepsilon}\delta b_1+\frac{1}{\varepsilon}\delta b_2$.

希望 $\|\delta x\|_\infty/\|x\|_\infty$ 较大时，希望 $\|x\|_\infty$ 较小，当 $x_1=0$ 时，$b_2=\frac{987}{610}b_1$

当 $x_1=x_2$ 时，$b_2=\left(\frac{987^2}{610\times1597}+\frac{\varepsilon+987}{1597}\right)b_1$ 当 $x_2=0$ 时，$b_2=\left(\frac{987}{610}+\frac{\varepsilon}{987}\right)b_1$.

于是取 $b_2=\left(\frac{987}{610\times1597}+\frac{\varepsilon+987}{1597}\right)b_1$. 为使 $\|b\|_\infty$ 较大，取 $b_1=10^3$，此时 $b_2\approx1618.03$

此时 $x_1=x_2\approx0.626$

再取 $\delta b_1=0.1$, $\delta b_2=1$. 解得 $\delta x_1\approx827.3$ $\delta x_2\approx511.3$

于是 $\frac{\|\delta b\|_\infty}{\|b\|_\infty}=\frac{1}{1618.03}$ 是 10^{-3} 级别. $\frac{\|\delta x\|_\infty}{\|x\|_\infty}=\frac{827.3}{0.626}$ 是 10^3 级别.

(2) $\|\delta b\|_\infty/\|b\|_\infty$ is very large while $\|\delta x\|_\infty/\|x\|_\infty$ is very small.

取 $b_1=b_2=10^3$. 解得 $x_1\approx6\times10^5$. $x_2\approx3.7\times10^5\times(-1)$.

再取 $\delta b_2=\left(\frac{987^2}{610\times1597}+\frac{\varepsilon+987}{1597}\right)\delta b_1$, $\delta b_1=1$. 此时 $\delta b_2\approx1.618$

而 $\delta x_1=\delta x_2\approx6.26\times10^{-4}$

于是 $\frac{\|\delta b\|_\infty}{\|b\|_\infty}=\frac{1.618}{10^3}$ 是 10^{-3} 级别

$\frac{\|\delta x\|_\infty}{\|x\|_\infty}=\frac{6.26\times10^{-4}}{6\times10^5}$ 是 10^{-9} 级别.

第二题

2. Let $Z \in \mathbb{C}^{n\times n}$ and $A = \begin{bmatrix} I_n & Z \\ 0 & I_n \end{bmatrix}$. Find $K_F(A) = \|A\|_F \|A^{-1}\|_F$

显然 $\|A\|_F = (n+n+\|Z\|_F^2)^{1/2}$

$A^{-1} = \begin{bmatrix} I_n & -Z \\ 0 & I_n \end{bmatrix}$. $\|A^{-1}\|_F = (n+n+\|Z\|_F^2)^{1/2}$

故 $K_F(A) = \|A\|_F \|A^{-1}\|_F = 2n + \|Z\|_F^2$.

第三题

3、由于严格对角优势阵在进行高斯消元后仍然是严格对角优势阵，故只需考虑 $a_{kk}^{(k)}$ 作为 $\max\limits_{i,j,k} |a_{ij}^{(k)}|$ 的候选。

对这种矩阵有：$|a_{ii}^{(k)}| > \sum\limits_{\substack{j\neq i \\ j=1}}^{n} |a_{ij}^{(k)}|$，利用 $a_{ij}^{(k+1)} = a_{ij}^{(k)} - a_{kj}^{(k)} \cdot \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ $\forall k = 1, \cdots, n$

$$a_{11}^{(1)} = a_{11}^{(1)}$$
$$a_{22}^{(2)} = a_{22}^{(1)} - \frac{a_{21}^{(1)}}{a_{11}^{(1)}} a_{12}^{(1)}$$
$$a_{33}^{(3)} = a_{33}^{(2)} - \frac{a_{32}^{(2)}}{a_{22}^{(2)}} \cdot a_{23}^{(2)}$$

进一步展开：$a_{33}^{(2)} = a_{33}^{(1)} - \dfrac{a_{31}^{(1)}}{a_{11}^{(1)}} \cdot a_{13}^{(1)}$

$a_{23}^{(2)} = a_{23}^{(1)} - \dfrac{a_{21}^{(1)}}{a_{11}^{(1)}} \cdot a_{13}^{(1)}$ 代入：

$$a_{33}^{(3)} = a_{33}^{(1)} - \left[\frac{a_{31}^{(1)}}{a_{11}^{(1)}} + \frac{a_{32}^{(2)}}{a_{22}^{(2)}} \cdot \frac{a_{21}^{(1)}}{a_{11}^{(1)}}\right] \cdot a_{13}^{(1)} - \frac{a_{32}^{(2)}}{a_{22}^{(2)}} \cdot a_{23}^{(1)}$$
$$< |a_{33}^{(1)}| + \frac{|a_{31}^{(1)}| + |a_{21}^{(1)}|}{|a_{11}^{(1)}|} \cdot |a_{13}^{(1)}| + \frac{|a_{32}^{(2)}|}{|a_{22}^{(2)}|} \cdot |a_{23}^{(1)}|.$$
$$< |a_{33}^{(1)}| + |a_{13}^{(1)}| + |a_{23}^{(1)}|$$

因为 $\dfrac{|a_{32}^{(2)}|}{|a_{22}^{(2)}|} < 1$. $\dfrac{|a_{31}^{(1)}| + |a_{21}^{(1)}|}{|a_{11}^{(1)}|} < 1$

以此类推 $a_{kk}^{(k)} = a_{kk}^{(1)} - \sum_{i=1}^{k-1}\left[\sum_{j=0}^{k-i-1}\alpha^{j}\cdot\frac{a_{(k-j)i}^{(i)}}{a_{ii}^{(i)}}\cdot a_{ik}^{(1)}\right]$.

其中 α 是小于1. 满足 $\alpha \le \sum_{\substack{j\ne i\\ j=1}}^{n}|a_{ji}^{(k)}| / |a_{ii}^{(k)}| < 1$ 的数、

因此 $a_{kk}^{(k)} < |a_{kk}^{(1)}| + \sum_{i=1}^{k-1}\left[\sum_{j=0}^{k-i-1}\frac{|a_{(k-j)i}^{(i)}|}{|a_{ii}^{(i)}|}\cdot a_{ik}^{(1)}\right]$

$< |a_{kk}^{(1)}| + \sum_{i=1}^{k-1}|a_{ik}^{(1)}| = \sum_{i=1}^{k}|a_{ik}| \le \sum_{i=1}^{n}|a_{ik}^{(1)}| < \max_{k}\left\{\sum_{i=1}^{n}|a_{ik}^{(1)}|\right\}$

因为 $\sum_{j=0}^{k-i-1}\frac{|a_{(k-j)i}^{(i)}|}{|a_{ii}^{(i)}|} = \frac{\sum_{j=0}^{k-i-1}|a_{(k-j)i}^{(i)}|}{|a_{ii}^{(i)}|} \le \frac{\sum_{\substack{j\ne i\\ j=1}}^{n}|a_{ji}^{(i)}|}{|a_{ii}^{(i)}|} < 1$.

而 $\|A\|_\infty = \max_{i}\left\{\sum_{j=1}^{n}|a_{ij}^{(1)}|\right\}$.

因此 $\rho = \frac{\max_{i,j,k}|a_{ij}^{(k)}|}{\|A\|_\infty} < \frac{\max_{k}\left\{\sum_{i=1}^{n}|a_{ik}^{(1)}|\right\}}{\max_{i}\left\{\sum_{j=1}^{n}|a_{ij}^{(1)}|\right\}} = \frac{\|A\|_1}{\|A\|_\infty} < 2$.

第四题

又，对于 $\begin{bmatrix} \ddots\ 0 & a'_{kk} & a'_{k(k+1)} & 0 \\ & a_{(k+1)k} & a_{(k+1)(k+1)} & a_{(k+1)(k+2)} \ddots \end{bmatrix}$ 无论是否交换主元，

都保持 $\begin{bmatrix} \ddots\ 0 & a''_{kk} & a''_{k(k+1)} & a''_{k(k+2)} \\ & 0 & a'_{(k+1)(k+1)} & a'_{(k+1)(k+2)} \end{bmatrix}$ 的形式继续进行消元。

估计有界

$$a'_{(k+1)(k+1)} = \begin{cases} a_{(k+1)(k+1)} - \dfrac{a_{(k+1)k}}{a'_{kk}} \cdot a'_{k(k+1)}, & |a'_{kk}| \geq |a_{(k+1)k}| \\ a'_{k(k+1)} - \dfrac{a'_{kk}}{a_{(k+1)k}} \cdot a_{(k+1)(k+1)}, & |a'_{kk}| < |a_{(k+1)k}|. \end{cases}$$

$$a'_{(k+1)(k+2)} = \begin{cases} a_{(k+1)(k+2)}, & |a'_{kk}| \geq |a_{(k+1)k}|. \\ -\dfrac{a'_{kk}}{a_{(k+1)k}} \cdot a_{(k+1)(k+2)}, & |a'_{kk}| < |a_{(k+1)k}|. \end{cases}$$

可知 $|a'_{(k+1)(k+1)}| \leq |a_{(k+1)(k+1)}| + |a'_{k(k+1)}|$，$|a'_{(k+1)(k+2)}| \leq |a_{(k+1)(k+2)}|$，

于是 $|a'_{(k+1)(k+1)}| \leq |a_{(k+1)(k+1)}| + |a_{(k+1)(k+2)}|$，$\rho \leq \dfrac{\max_k\{|a_{kk}| + 2|a_{k(k+1)}| + 0\}}{\|A\|_\infty} < 2$

第五题

程序 T5.py 的运行结果保存在 T5_result.txt 中

两个系数矩阵都是满秩的，因此方程有唯一非零解，可以知道精确解是皆为 1

结果显示，对第一个线性方程组（主对角线为8，两条副对角线为1和6）分别用全选主元、部分选主元和不选主元得出的解相同，且相对精确解的误差较小；对第二个线性方程组（主对角线为6，两条副对角线为1和8），虽然分别用全选主元、部分选主元和不选主元得出的解相同，但是相对精确解的误差极大

可能是因为第二个系数矩阵的条件数过大，这个问题本身是病态的，因此无论如何选用或者不选用高斯消元法都不能得到相对于精确解误差不大的解

文件 T5.py

```python
import numpy as np

def pivoting_gaussian_elimination(A_origin, b_origin):
    A = np.copy(A_origin)
    b = np.copy(b_origin)
    n = len(b)
    Ab = np.hstack([A, b.reshape(-1, 1)])

    for i in range(n):
        max_row = np.argmax(np.abs(A[i:, i])) + i
        if max_row != i:
            A[[i, max_row], :] = A[[max_row, i], :]
        max_col = np.argmax(np.abs(A[i, i:])) + i
        if max_col != i:
            A[:, [i, max_col]] = A[:, [max_col, i]]
        for j in range(i+1, n):
            factor = Ab[j, i] / Ab[i, i]
            Ab[j, i:] -= factor * Ab[i, i:]

    x = np.zeros(n)
    for i in range(n-1, -1, -1):
        x[i] = (Ab[i, -1] - np.dot(Ab[i, i+1:n], x[i+1:n])) / Ab[i, i]

    return x


def partial_pivoting_gaussian_elimination(A_origin, b_origin):
    A = np.copy(A_origin)
    b = np.copy(b_origin)
    n = len(b)
    Ab = np.hstack([A, b.reshape(-1, 1)])

    for i in range(n):
        max_row = np.argmax(np.abs(A[i:, i])) + i
        if max_row != i:
            A[[i, max_row], :] = A[[max_row, i], :]

        for j in range(i+1, n):
            factor = Ab[j, i] / Ab[i, i]
            Ab[j, i:] -= factor * Ab[i, i:]

    x = np.zeros(n)
    for i in range(n-1, -1, -1):
```

```python
        x[i] = (Ab[i, -1] - np.dot(Ab[i, i+1:n], x[i+1:n])) / Ab[i, i]

    return x


def gaussian_elimination(A_origin, b_origin):
    A = np.copy(A_origin)
    b = np.copy(b_origin)
    n = len(b)
    Ab = np.hstack([A, b.reshape(-1, 1)])

    for i in range(n):
        for j in range(i+1, n):
            factor = Ab[j, i] / Ab[i, i]
            Ab[j, i:] -= factor * Ab[i, i:]

    x = np.zeros(n)
    for i in range(n-1, -1, -1):
        x[i] = (Ab[i, -1] - np.dot(Ab[i, i+1:n], x[i+1:n])) / Ab[i, i]

    return x

n = 100

# matric one

A1 = np.zeros((n, n), dtype=np.float32)
for i in range(n-1):
    A1[i, i] = 8
    A1[i, i+1] = 1
    A1[i+1, i] = 6

A1[n-1, n-1] = 8

b1 = np.zeros(n, dtype=np.float32)
for i in range(n):
    b1[i] = 15
b1[0] -= 6
b1[n-1] -= 1

without_sol1 = gaussian_elimination(A1, b1)

with_sol1 = pivoting_gaussian_elimination(A1, b1)
```

```
with_partial_sol1 = partial_pivoting_gaussian_elimination(A1, b1)


# matric two

A2 = np.zeros((n, n), dtype=np.float32)
for i in range(n-1):
    A2[i, i] = 6
    A2[i, i+1] = 1
    A2[i+1, i] = 8

A2[n-1, n-1] = 6

b2 = np.zeros(n, dtype=np.float32)
for i in range(n):
    b2[i] = 15
b2[0] -= 8
b2[n-1] -= 1

without_sol2 = gaussian_elimination(A2, b2)

with_sol2 = pivoting_gaussian_elimination(A2, b2)

with_partial_sol2 = partial_pivoting_gaussian_elimination(A2, b2)

# print

print(with_sol1)
print(with_partial_sol1)
print(without_sol1)

print(with_sol2)
print(with_partial_sol2)
print(without_sol2)
```