Numerical HW #9

(A)

Brandon Finley HW #9

A

$A = \begin{pmatrix} 100 & 99 \\ 99 & 98 \end{pmatrix} \quad A^{-1} = \begin{pmatrix} -98 & 99 \\ 99 & -100 \end{pmatrix}$

i) $AA^{-1} = \begin{pmatrix} 100 & 99 \\ 99 & 98 \end{pmatrix}\begin{pmatrix} -98 & 99 \\ 99 & -100 \end{pmatrix}$

$= \begin{pmatrix} 100\cdot -98 + 99^2 & \underbrace{-98\cdot 99 + 99\cdot 98}_{0} \\ \underbrace{100\cdot 99 + 99\cdot -100}_{0} & 99^2 - 98\cdot 100 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$

ii) $\text{cond}(A)_p \quad p = 1, 2, \infty$

$\text{cond}(A)_1 = \|A\|_1 \|A^{-1}\|_1 = (100+99)(100+99) = 199^2 = \boxed{39601}$

$\text{cond}(A)_\infty = \|A\|_\infty \|A^{-1}\|_\infty = (100+99)(99+100) = 199^2 = \boxed{39601}$

$\text{cond}(A)_2 = \sqrt{\rho_0(A^TA)} \Rightarrow A^TA = \begin{pmatrix} 100 & 99 \\ 99 & 98 \end{pmatrix}\begin{pmatrix} 100 & 99 \\ 99 & 98 \end{pmatrix} = \begin{pmatrix} 10000 & 9801 \\ 9801 & 9604 \end{pmatrix}$ $\rho_0(A^TA)$

$\Rightarrow \begin{pmatrix} 10000-\lambda & 9801 \\ 9801 & 9604-\lambda \end{pmatrix} \Rightarrow (10{,}000-\lambda)(9604-\lambda) - 9801^2 = 0$

$\Rightarrow \lambda_{1,2} = 1.96045e^4, -9.99e^{-2}$

$\Rightarrow \sqrt{1.96045e^4} = \boxed{140.0178}$

iii) $b = (1, 1)^T, \quad \tilde{b} = (0.9, 1.1)^T$

$Ax = b \Rightarrow x = A^{-1}b = \begin{pmatrix} -98 & 99 \\ 99 & -100 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \boxed{\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \vec{x}}$

$A\tilde{x} = \tilde{b} \Rightarrow \tilde{x} = A^{-1}\tilde{b} = \begin{pmatrix} -98 & 99 \\ 99 & -100 \end{pmatrix}\begin{pmatrix} 0.9 \\ 1.1 \end{pmatrix} = \begin{pmatrix} 20.7 \\ -20.9 \end{pmatrix} = \tilde{x}$

$\Rightarrow \Delta x = (\tilde{x} - \vec{x}) = \begin{pmatrix} 19.7 \\ -19.9 \end{pmatrix} \Rightarrow$ About 1900% off !! ☹

It appears that the relative change to x is 20 while the change in b is 0.1

$\Rightarrow$ This is a <u>massive</u> difference

(B)

```
import numpy as np

def GuassJacobi(A, b, G, true):

    temp = np.zeros((A.shape[0]), dtype = float)
    g = np.copy(G)
    count = 0
    error_x = 0
    while error_x == 0:
        for i in range(0, A.shape[0]):
            s = customDot(A[i,:], g, i)
            temp[i] = (1/A[i][i])*(b[i] - s)
        count += 1
        g = np.copy(temp)
        print('X_' + str(count) + ':', g)
        error_x = error(g, true)
    return g, count

def GuassSeidel(A, b, G, true):

    g = np.copy(G)
    count = 0
    error_x = 0
    while error_x == 0:
        for i in range(0, A.shape[0]):
            s = customDot(A[i, :], g, i)
            g[i] = (1/A[i][i])*(b[i] - s)
        count += 1
        print('X_' + str(count) + ':', g)
        error_x = error(g, true)
    return g, count

def SOR(A, b, G, w, true):

    g = np.copy(G)
    count = 0
    error_x = 0
    while error_x == 0:
        for i in range(0, A.shape[0]):
            s = customDot(A[i, :], g, i)
            g[i] = w*(1/A[i][i])*(b[i] - s) + g[i]*(1 - w)
        count += 1
        print('X_' + str(count) + ':', g)
```

```python
        error_x = error(g, true)
    return g, count

def customDot(a, g, k):
    s = 0
    for i in range(0, a.shape[0]):
        if i != k:
            s += a[i]*g[i]

    return s
def error(pred, true):
    valid = 1
    for i in range(0, pred.shape[0]):
        if abs(pred[i] - true[i]) > 10e-8:
            return 0
    return valid
def main():
    # Problem to Solve

    A = np.array([ \
        [4, -1, 0, -1, 0, 0], \
        [-1, 4, -1, 0, -1, 0], \
        [0, -1, 4, 0, 0, -1], \
        [-1, 0, 0, 4, -1, 0], \
        [0, -1, 0, -1, 4, -1], \
        [0, 0, -1, 0, -1, 4]])
    b = np.array([2, 1, 2, 2, 1, 2]).T
    x0 = np.array([0, 0, 0, 0, 0, 0], dtype=float)
    true = np.ones((6, 1))

    # Using Jacobi, Seidel, and SOR
    print('\nJacobi:')
    test, count = GuassJacobi(A, b, x0, true)
    print('Iterations:', count)

    print('\nSeidel:')
    test, count = GuassSeidel(A, b, x0, true)
    print('Iterations:', count)

    w = 1.113
    print('\nSOR:')
    test, count = SOR(A, b, x0, w, true)
    print('Iterations:', count)
if __name__ == "__main__":
    main()
```

```
Output:
Jacobi:
X_1: [0.5  0.25 0.5  0.5  0.25 0.5 ]
X_2: [0.6875 0.5625 0.6875 0.6875 0.5625 0.6875]
X_3: [0.8125   0.734375 0.8125   0.8125   0.734375 0.8125  ]
X_4: [0.88671875 0.83984375 0.88671875 0.88671875 0.83984375 0.88671875]
X_5: [0.93164062 0.90332031 0.93164062 0.93164062 0.90332031 0.93164062]
X_6: [0.95874023 0.94165039 0.95874023 0.95874023 0.94165039 0.95874023]
X_7: [0.97509766 0.96478271 0.97509766 0.97509766 0.96478271 0.97509766]
X_8: [0.98497009 0.97874451 0.98497009 0.98497009 0.97874451 0.98497009]
X_9: [0.99092865 0.98717117 0.99092865 0.99092865 0.98717117 0.99092865]
X_10: [0.99452496 0.99225712 0.99452496 0.99452496 0.99225712 0.99452496]
X_11: [0.99669552 0.99532676 0.99669552 0.99669552 0.99532676 0.99669552]
X_12: [0.99800557 0.99717945 0.99800557 0.99800557 0.99717945 0.99800557]
X_13: [0.99879625 0.99829765 0.99879625 0.99879625 0.99829765 0.99879625]
X_14: [0.99927348 0.99897254 0.99927348 0.99927348 0.99897254 0.99927348]
X_15: [0.9995615  0.99937987 0.9995615  0.9995615  0.99937987 0.9995615 ]
X_16: [0.99973534 0.99962572 0.99973534 0.99973534 0.99962572 0.99973534]
X_17: [0.99984027 0.9997741  0.99984027 0.99984027 0.9997741  0.99984027]
X_18: [0.99990359 0.99986366 0.99990359 0.99990359 0.99986366 0.99990359]
X_19: [0.99994181 0.99991771 0.99994181 0.99994181 0.99991771 0.99994181]
X_20: [0.99996488 0.99995033 0.99996488 0.99996488 0.99995033 0.99996488]
X_21: [0.9999788  0.99997002 0.9999788  0.9999788  0.99997002 0.9999788 ]
X_22: [0.99998721 0.99998191 0.99998721 0.99998721 0.99998191 0.99998721]
X_23: [0.99999228 0.99998908 0.99999228 0.99999228 0.99998908 0.99999228]
X_24: [0.99999534 0.99999341 0.99999534 0.99999534 0.99999341 0.99999534]
X_25: [0.99999719 0.99999602 0.99999719 0.99999719 0.99999602 0.99999719]
X_26: [0.9999983 0.9999976 0.9999983 0.9999983 0.9999976 0.9999983]
X_27: [0.99999898 0.99999855 0.99999898 0.99999898 0.99999855 0.99999898]
X_28: [0.99999938 0.99999913 0.99999938 0.99999938 0.99999913 0.99999938]
X_29: [0.99999963 0.99999947 0.99999963 0.99999963 0.99999947 0.99999963]
X_30: [0.99999977 0.99999968 0.99999977 0.99999977 0.99999968 0.99999977]
X_31: [0.99999986 0.99999981 0.99999986 0.99999986 0.99999981 0.99999986]
X_32: [0.99999992 0.99999988 0.99999992 0.99999992 0.99999988 0.99999992]
X_33: [0.99999995 0.99999993 0.99999995 0.99999995 0.99999993 0.99999995]
Iterations: 33

Seidel:
X_1: [0.5       0.375     0.59375   0.625     0.5       0.7734375]
X_2: [0.75       0.7109375  0.87109375 0.8125     0.82421875 0.92382812]
X_3: [0.88085938 0.89404297 0.95446777 0.92626953 0.93603516 0.97262573]
X_4: [0.95507812 0.96139526 0.98350525 0.97277832 0.97669983 0.99005127]
X_5: [0.9835434  0.98593712 0.9939971  0.99006081 0.9915123  0.99637735]
X_6: [0.99399948 0.99487722 0.99781364 0.99637794 0.99690813 0.99868044]
X_7: [0.99781379 0.99813389 0.99920358 0.99868048 0.9988737  0.99951932]
```

```
X_8: [0.99920359 0.99932022 0.99970989 0.99951932 0.99958972 0.9998249 ]
X_9: [0.99970989 0.99975237 0.99989432 0.9998249  0.99985054 0.99993622]
X_10: [0.99989432 0.99990979 0.9999615  0.99993622 0.99994556 0.99997676]
X_11: [0.9999615  0.99996714 0.99998598 0.99997676 0.99998017 0.99999154]
X_12: [0.99998598 0.99998803 0.99999489 0.99999154 0.99999278 0.99999692]
X_13: [0.99999489 0.99999564 0.99999814 0.99999692 0.99999737 0.99999888]
X_14: [0.99999814 0.99999841 0.99999932 0.99999888 0.99999904 0.99999959]
X_15: [0.99999932 0.99999942 0.99999975 0.99999959 0.99999965 0.99999985]
X_16: [0.99999975 0.99999979 0.99999991 0.99999985 0.99999987 0.99999995]
X_17: [0.99999991 0.99999992 0.99999997 0.99999995 0.99999995 0.99999998]
Iterations: 17

SOR:
X_1: [0.5565     0.43309612 0.677009   0.71134612 0.59669106 0.91090704]
X_2: [0.81205656 0.80967191 0.95874908 0.86810191 0.93112436 0.97942479]
X_3: [0.93157818 0.97182599 0.99109688 0.97670147 0.98773566 0.99643515]
X_4: [0.99340943 0.99545999 0.99875088 0.99738635 0.99840345 0.99961102]
X_5: [0.99875423 0.99937458 0.99985889 0.99950447 0.99976027 0.99993799]
X_6: [0.99982887 0.99991709 0.99997562 0.99994167 0.99997053 0.99999202]
X_7: [0.99998004 0.99998883 0.99999743 0.99999284 0.99999601 0.99999908]
X_8: [0.99999716 0.99999864 0.99999966 0.99999891 0.99999951 0.99999987]
X_9: [0.99999964 0.99999982 0.99999995 0.99999989 0.99999994 0.99999998]
X_10: [0.99999996 0.99999998 1.         0.99999998 0.99999999 1.        ]
Iterations: 10
```