Math 374

Final Exam

Name: Gianluca Crescenzo

Exercise 2. In class, we saw the Jacobi and Gauss Seidel iterative methods for solving a linear system of the form $A\vec{x} = \vec{b}$. These methods partitioned the matrix A as $A = L+D+U$, where D is a diagonal matrix, U is the part of A above the diagonal, and L is the part of A below the diagonal. The Jacobi method was derived as:

$$\overrightarrow{x_{n+1}} = D^{-1}\left(\vec{b} - (U+L)\vec{x_n}\right).$$

The Gauss-Seidel method was derived as:

$$\overrightarrow{x_{n+1}} = (D+L)^{-1}\left(\vec{b} - U\vec{x_n}\right).$$

(a) Prove, using your knowledge of Linear Algebra, that if A has a 0 on the diagonal, neither method above will produce a solution without somehow rearranging the rows of A and $\vec{b}$.

(b) Compose a new iterative method similar to the Jacobi or Gauss-Seidel method which can find a solution of $A\vec{x} = \vec{b}$ without rearranging rows.

(c) Use your new method to determine a solution to the linear system $A\vec{x} = \vec{b}$, where:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 0 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix}.$$

Continue to iterate until your result shows four decimal digit stability. How many iterations were required?

Solution. (a) Note that D is a diagonal matrix and $D+L$ is a lower triangular matrix. The determinant of a diagonal matrix is the product of its diagonal. Likewise, the determinant of a triangular matrix is the product of its diagonal. Whence $\det(D) = \det(D+L) = 0$. Since both matrices have a determinant of 0, neither are invertible. Thus, neither the Gauss-Seidel nor Jacobi method will produce a solution.

(b) Let $(L+D+U)\vec{x} = \vec{b}$. We can express this as $(L+U)\vec{x} = \vec{b} - D\vec{x}$. Thus our iterative method is $\overrightarrow{x_{n+1}} = (L+U)^{-1}\left(\vec{b} - D\vec{x_n}\right)$.

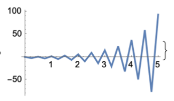
(c) We use the following Mathematica code:

```
In[1]:= Clear["Global`*"]
        it=100;
        A={{1,2,3},{4,0,6},{7,8,9}};
        b={{4},{2},{1}};
        x0 = {{0},{0},{0}};

        JD=DiagonalMatrix[Diagonal[A]] ;
        JL=LowerTriangularize[A,-1];
        JU=UpperTriangularize[A,1];

        x=Table[0,{it+1},{Length[b]}];
        x[[1]]=x0;
        For[n=1,n⩽it,n++,
          x[[n+1]]=N[PseudoInverse[JL + JU].(b-(JD).x[[n]]),100000];
          If[Max[Abs[x[[n+1]]-x[[n]]]]<0.0001,Break[];];
          xLastit=n+1;
        ];
        table=Table[{
          n-1,
          NumberForm[x[[n]],{Infinity,8},ExponentFunction→(Null&),
            NumberPadding→{"","0"}]
        },{n,1,xLastit}];
        TableForm[Prepend[table,{"n","Method"}]]

Out[14]//TableForm=
              n    Method
              0    {{0},{0},{0}}
              1    {{-0.73333333},{0.76666667},{0.82222222}}
              2    {{-1.42222222},{0.44444444},{1.28148148}}
              3    {{-1.88148148},{0.32962963},{1.58765432}}
              4    {{-2.18765432},{0.25308642},{1.79176955}}
              5    {{-2.39176955},{0.20205761},{1.92784636}}
              6    {{-2.52784636},{0.16803841},{2.01856424}}
              7    {{-2.61856424},{0.14535894},{2.07904283}}
              8    {{-2.67904283},{0.13023929},{2.11936189}}
              9    {{-2.71936189},{0.12015953},{2.14624126}}
              10   {{-2.74624126},{0.11343969},{2.16416084}}
              11   {{-2.76416084},{0.10895979},{2.17610723}}
              12   {{-2.77610723},{0.10597319},{2.18407148}}
              13   {{-2.78407148},{0.10398213},{2.18938099}}
              14   {{-2.78938099},{0.10265475},{2.19292066}}
              15   {{-2.79292066},{0.10176984},{2.19528044}}
              16   {{-2.79528044},{0.10117989},{2.19685363}}
              17   {{-2.79685363},{0.10078659},{2.19790242}}
              18   {{-2.79790242},{0.10052440},{2.19860161}}
              19   {{-2.79860161},{0.10034960},{2.19906774}}
              20   {{-2.79906774},{0.10023306},{2.19937849}}
              21   {{-2.79937849},{0.10015538},{2.19958566}}
              22   {{-2.79958566},{0.10010358},{2.19972378}}
              23   {{-2.79972378},{0.10006906},{2.19981585}}
```

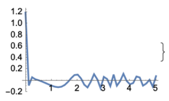
We can see that 23 iterations were required.

Exercise 3. Use the Euler method to determine a numerical solution to the initial value problem: $\frac{dy}{dx} = -9y + \cos(e^x)$, $y(0) = 1.2$ over the interval $0 \leqslant x \leqslant 5$. Choose one (1) value of Δx for which the method is unstable, and one (1) value of Δx for which the method is stable. Print your solution curves for each case.

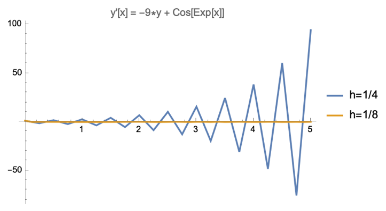
Solution. The following Mathematica code shows that $\Delta x = \frac{1}{4}$ gives an unstable result, while $\Delta x = \frac{1}{8}$ gives a stable result.

```
In[140]:= Clear["Global`*"]
          EulersMethod[f_,leftEndpoint_,rightEndpoint_,deltaX_,
              initialCondition_]:=
            Module[{a,b,h,x,y,yVals,n,points,plot},
             a=leftEndpoint;
             b=rightEndpoint;
             h=deltaX;
             x=Table[x,{x,a,b,h}];
             y=Table[0,{Length[x]}];
             y[[1]]=initialCondition;
             For[n=1,n<Length[x],n++,
              y[[n+1]]=N[y[[n]]+h*f[x[[n]],y[[n]]],10];];
             points=Transpose[{x,y}];
             plot=ListLinePlot[points,PlotRange→All];
             {MatrixForm[points],plot,points}
          ];

          f[x_,y_]:=-9*y + Cos[Exp[x]];
          p1=EulersMethod[f,0,5,1/4,1.2];
          p2=EulersMethod[f,0,5,1/8,1.2];
          {p1[[1]],p1[[2]]}
          {p2[[1]],p2[[2]]}
          ListLinePlot[{p1[[3]],p2[[3]]},
            PlotLabel→"y'[x] = -9*y + Cos[Exp[x]]",PlotRange→All,
            PlotLegends→{"h=1/4","h=1/8"}]

Out[140]=
{ 0      1.2
  1/4   -1.364924424
  1/2    1.77686965
  3/4   -2.240548589
  1      2.670823981
  5/4   -3.566463454
  3/2    4.223129194
  7/4   -5.336076229
  2      6.885975969
  9/4   -8.4953809
  5/2    10.36972142
  11/4  -12.73034685
  3      15.66346689
  13/4  -19.49718493
  7/2    24.56934672
  15/4  -30.74377686
  4      38.45706158
  17/4  -48.1639924
  9/2    60.3420848
  19/4  -75.54343661
  5      94.23097953 ,
```


```
}
```

```
{ 0      1.2
  1/8   -0.08246221177
  1/4    0.0632840513
  3/8    0.02744655409
  1/2    0.01101246157
  5/8   -0.01110732069
  3/4   -0.03524693333
  7/8   -0.06052501117
  1     -0.08451356758
  9/8   -0.1034025434
  5/4   -0.1118393191
  11/8  -0.1034951473
  3/2   -0.07293445491
  13/8  -0.01946556145
  7/4    0.04717210837
  15/8   0.1020438278
  2      0.1087317345
  17/8   0.04245306341
  9/4   -0.06729900836
  19/8  -0.1163399749
  5/2   -0.01572381684
  21/8   0.1178679435
  11/4   0.02607835614
  23/8  -0.1279931275
  3      0.0699941038
  25/8   0.03232508145
  13/4  -0.09388847244
  27/8   0.1106688387
  7/2   -0.08654908505
  29/8  -0.005228096655
  15/4   0.1237576108
  31/8  -0.001799451462
  4     -0.06101396323
  33/8  -0.03870597127
  17/4   0.07611357659
  35/8   0.05903294986
  9/2   -0.08507809187
  37/8  -0.04728054395
  19/4   0.01826339045
  39/8  -0.1014410423
  5      0.08305996525 ,
```


```
}
```



Exercise 4. Single but Exciting Questions!

(a) Let $T(4n) = 3T(4n-4)$. Determine $T(n)$ explicitly.

(b) Express the number -219.1963 as a 64 bit double precision number in IEEE 754 format.

(c) List the eigenvalues and eigenvectors for the matrix:

$$A = \begin{bmatrix} 3 & 8 & 0 \\ 2 & 3 & 8 \\ 0 & 2 & 3 \end{bmatrix}$$

using only an appropriate formula, and without computing the characteristic polynomial or solving any linear systems.

Solution. (a) Let $k = 4n$. Observe that:

$$\begin{aligned} T(4n) &= T(k) \\ &= 3T(k-4) \\ &= 3^2T(k-8) \\ &= 3^3T(k-16) \\ &\vdots \end{aligned}$$

Inductively, $T(k) = 3^{\frac{k}{4}}T(0)$ for $k \equiv 0 \pmod 4$.

(b) The following Mathematica code will express -219.1963 as a 64 bit double precision number.

```
In[140]:= Clear["Global`*"]
          Binary[totalBits_,val_]:=
            Module[{sbit=,ebits=,mbits=,n,j,c,q,r,counter,digit,s,
              eBits,mBits,bias},
             Switch[totalBits,
              16,{eBits=5;mBits=10;bias=15;},
              32,{eBits=8;mBits=23;bias=127;},
              64,{eBits=11;mBits=52;bias=1023;},
              128,{eBits=15;mBits=112;bias=16383;},
              256,{eBits=19;mBits=236;bias=262143;},
              _,Return[unsupported size]];
            Clear[j,n,c,q,r,counter,digit,s];
            n=Abs[val];
            If[val<0,sbit=1,sbit=0];
            ebits=;
            mbits=;
            If[n<2,e=bias;
              Do[{q,r}=QuotientRemainder[e,2];
               ebits=ToString[r]<>ebits;
               e=q;,{eBits}];
              f=n-1;
              Do[s=2 f;
               digit=Floor[s];
               mbits=mbits<>ToString[digit];
               f=s-digit;,{mBits}];,counter=0;
```

```
        While[n⩾2,n=n/2;
         counter++;];
        e=counter+bias;
        Do[{q,r}=QuotientRemainder[e,2];
         ebits=ToString[r]<>ebits;
         e=q;,{eBits}];
        f=n-1;
        Do[s=2 f;
         digit=Floor[s];
         mbits=mbits<>ToString[digit];
         f=s-digit;,{mBits}];];
        sbit <>ebits<> mbits]
    Binary[64,-219.1963]

Out[142]= 1100000001101011011001100100100000010110111100000000011010001110
```

(c) The eigenvalues of A are given by:

$$\lambda_h(A) = 3 + 4\cos\left(\frac{h\pi}{4}\right), \quad h = 1, 2, 3.$$

The corresponding eigenvector of $\lambda_h(A)$ is given by:

$$\begin{bmatrix} \left(\frac{1}{4}\right)^{\frac{1}{2}} \sin\left(\frac{h\pi}{4}\right) \\ \left(\frac{1}{4}\right) \sin\left(\frac{2h\pi}{4}\right) \\ \left(\frac{1}{4}\right)^{\frac{3}{2}} \sin\left(\frac{3h\pi}{4}\right) \end{bmatrix}$$

Exercise 5. Consider the two equations $x^3 + y^2 = 2$ and $2x^2 + \sin(y) = 3$.

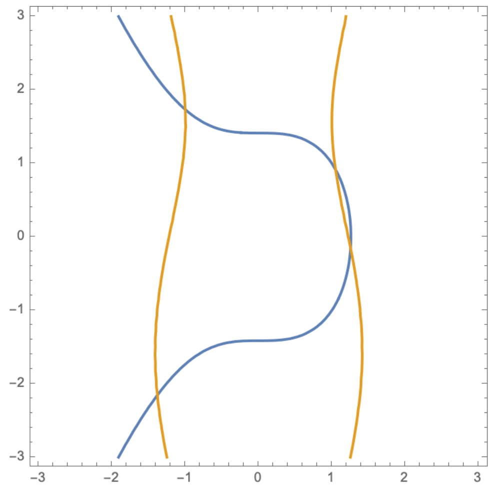
(a) Research the Mathematica command "ContourPlot". Then plot both of these equations in the same plot on the interval $-3 \leqslant x, y \leqslant 3$. Turn in your plot.

(b) If your graph is correct, you should see four points of intersection between the two equations. Using your graph to "suggest" good initial guesses, use an iterative method to determine the four points of intersection $(x_1, y_1), ..., (x_4, y_4)$, stable to 4 decimal places.

Solution. (a) The Mathematica cell:

```
In[146]:= ContourPlot[{x^3+y^2==2,2 x^2+Sin[y]==3},{x,-3,3},{y,-3,3}]
```

outputs the following graph:



(b) The following Mathematica code determines all four points of intersection.

```
In[147]:= Clear["Global`*"]
          NewtonsMethod[f1_,f2_,x0_,y0_,tol_:0.0001,maxIter_:20]:=Module[{
               xVals=ConstantArray[0.,maxIter+1],yVals=ConstantArray[0.,maxIter+1],
               delta,n,F,J,sol},F[x_,y_]={{f1[x,y]},{f2[x,y]}};
              J[x_,y_]={{D[f1[x,y],x],D[f1[x,y],y]},
                 {D[f2[x,y],x],D[f2[x,y],y]}};
              xVals[[1]]=N[x0];
              yVals[[1]]=N[y0];
              For[n=1,n⩽maxIter,n++,
               delta=Inverse[J[xVals[[n]],yVals[[n]]]].F[xVals[[n]],yVals[[n]]];
               xVals[[n+1]]=xVals[[n]]-delta[[1,1]];
               yVals[[n+1]]=yVals[[n]]-delta[[2,1]];
               If[Max[Abs[{xVals[[n+1]],yVals[[n+1]]}-{xVals[[n]],yVals[[n]]}]]<tol,
                xVals=Take[xVals,n+1];
                yVals=Take[yVals,n+1];
                Break[]];
              ];
              sol=N[Transpose[{xVals,yVals}]];
              table=
               Table[{n-1,NumberForm[sol[[n]],{Infinity, 10},ExponentFunction→(Null&),
                     NumberPadding→{,0}]},{n,Length@sol}];
              TableForm[Prepend[table,{n,"(x,y)"}]]]

          f1[x_,y_]=x^3+y^2-2;
          f2[x_,y_]=2 x^2+Sin[y]-3;
          NewtonsMethod[f1,f2,1,1]
          NewtonsMethod[f1,f2,-1,-2]
          NewtonsMethod[f1,f2,-1,2]
          NewtonsMethod[f1,f2,1,-1]
```

Out[-120]//TableForm=
```
n (x,y)
0 {1.0000000000,1.0000000000}
1 {1.0497026813,0.9254459781}
2 {1.0502192250,0.9174500191}
3 {1.0502341737,0.9173877601}
```

Out[-119]//TableForm=
```
n (x,y)
0 {-1.0000000000,-2.0000000000}
1 {-1.4669022999,-2.1001767249}
2 {-1.3871062327,-2.1550829268}
3 {-1.3843773549,-2.1571122515}
4 {-1.3843741591,-2.1571142160}
```

Out[-118]//TableForm=
```
n (x,y)
0 {-1.0000000000,2.0000000000}
1 {-0.9963843454,1.7472882591}
2 {-1.0033616158,1.7349710779}
3 {-1.0033556981,1.7349642401}
```

Out[-117]//TableForm=
```
n (x,y)
0 {1.0000000000,-1.0000000000}
1 {1.3828061119,-0.4257908322}
2 {1.2718843155,-0.2036910745}
3 {1.2557034631,-0.1534251740}
4 {1.2551061728,-0.1511568631}
5 {1.2551050824,-0.1511524435}
```