APMTH 205 midterm
Louis Baum
November 15, 2016

Problem 1

For this problem the code can be found in the accompanying file prob1.py

Using the iterative approach I have found R,P (rounded for clarity)

$$R_{it} = \begin{bmatrix} .969 & .241 & .055 & -.011 \\ -.245 & .961 & .112 & .055 \\ -.020 & -.131 & -.961 & .241 \\ .030 & -.020 & -.245 & .969 \end{bmatrix}$$

$$P_{it} = \begin{bmatrix} 15.5 & 3.83 & .642 & .796 \\ * & 17.3 & 2.23 & .800 \\ * & * & 15.9 & 4.05 \\ * & * & * & 17.5 \end{bmatrix}$$

P is symmetric

R and P calculated with the SVD method are equivalent to the precision depicted here. To determine the difference I calculated the Frobenius norm between the matrices:

$$\left|\left|P_{it} - P_{svd}\right|\right|_{Fro} = 2.302 * 10^{-14}$$

$$\left|\left|R_{it} - R_{svd}\right|\right|_{Fro} = 7.384 * 10^{-16}$$

Both of these values are small indicating that these methods produce R,P matricies with small differences.

Problem 2

We will first expand $f(x-3\Delta x)$, $f(x-2\Delta x)$, $f(x)$, $f(x+2\Delta x)$, $f(x+3\Delta x)$ as Tailor series to 4th order.

$$
\begin{aligned}
f(x-3\Delta x) &= f(x) \text{ - } 3\Delta x f'(x) + \tfrac{9}{2}(\Delta x)^2 f''(x) \text{-} \tfrac{9}{2}(\Delta x)^3 f'''(x) + \tfrac{27}{8}(\Delta x)^4 f''''(x) \\
f(x-2\Delta x) &= f(x) \text{ - } 2\Delta x f'(x) + 2(\Delta x)^2 f''(x) \text{-} \tfrac{4}{3}(\Delta x)^3 f'''(x) + \tfrac{2}{3}(\Delta x)^4 f''''(x) \\
f(x) &= f(x) \\
f(x+2\Delta x) &= f(x) + 2\Delta x f'(x) + 2(\Delta x)^2 f''(x) + \tfrac{4}{3}(\Delta x)^3 f'''(x) + \tfrac{2}{3}(\Delta x)^4 f''''(x) \\
f(x+3\Delta x) &= f(x) + 3\Delta x f'(x) + \tfrac{9}{2}(\Delta x)^2 f''(x) + \tfrac{9}{1}(\Delta x)^3 f'''(x) + \tfrac{27}{8}(\Delta x)^4 f''''(x)
\end{aligned}
$$

Now we will look for a linear combination of these elements such that the $f'(x)$ has coefficient 1 and there are no restrictions on $f^5(x)$ coefficients

Let us define a vector X and b such that:

$$
X = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \epsilon \end{bmatrix} \qquad b = \begin{bmatrix} \text{f(x)} \\ \text{f'(x)} \\ \text{f''(x)} \\ \text{f'''(x)} \\ \text{f''''(x)} \end{bmatrix}
$$

(the vector b is made of the coefficients of the derivatives)

We set up a vector equation AX =b where A is given by the coefficients of the Taylor expansion and b correcponds to the coefficients of the derivatives above.

$$
\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -3 & -2 & 0 & 2 & 3 \\ \frac{9}{2} & 2 & 0 & 2 & \frac{9}{2} \\ -\frac{9}{2} & -\frac{4}{3} & 0 & \frac{4}{3} & \frac{9}{2} \\ \frac{27}{8} & \frac{2}{3} & 0 & \frac{2}{3} & \frac{27}{8} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \epsilon \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

solving this expression we determine

$$
\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \epsilon \end{bmatrix} = \begin{bmatrix} \frac{2}{15} \\ -\frac{9}{20} \\ 0 \\ \frac{9}{20} \\ -\frac{2}{15} \end{bmatrix}
$$

with the above coefficients the next leading order term is of order at least $(\Delta x)^4$ thus we have a 4th order accurate finite difference approximation for the derivative of $f(x)$.

Problem 3

a)

$$F(\mathbf{X})=\begin{bmatrix} \frac{\partial x}{\partial X} & \frac{\partial x}{\partial Y} \\ \\ \frac{\partial y}{\partial X} & \frac{\partial y}{\partial Y} \end{bmatrix}$$

$$F(\mathbf{X})=$$
$$\begin{bmatrix} 1+0.2Cos(3Y)Cos(X)-0.1Sin(X+Y) & -0.3Sin(X)Sin(3Y)-0.1Sin(X+Y) \\ \\ -0.3Sin(3X+2Y) & 1-0.2Sin(3X+2Y) \end{bmatrix}$$

b)

I have implemented a script that will determine the zero of one of the three scaler fields to within 10^{-5}. This script is 'findzero(Xvec,i)' it takes an initial guess "Xvec" and which of the scaler fields you are interested in i = 0,1,2. it then calculates G and grad G in the neigborhood of that point and ammends the guess until the difference between the kth and k+1th step is smaller than 10^{-6}.

c)

To determine the sufficient amount of locations I will calculate the number of distinct critical points as I increase the number of starting points. Once the number of critical points is the same for increase numbers of initial guesses I will assume I found them all.

For G1 we obtained the following information.

# of Guesses	Critical Points
4	1
9	4
25	15
64	41
100	44
169	44

Thus 100 initial guesses were enough to find all of the critical points.

For G2

# of Guesses	Critical Points
4	1
25	12
64	35
100	36
144	36

again 100 initial guesses were enough to find all of the critical points of G2

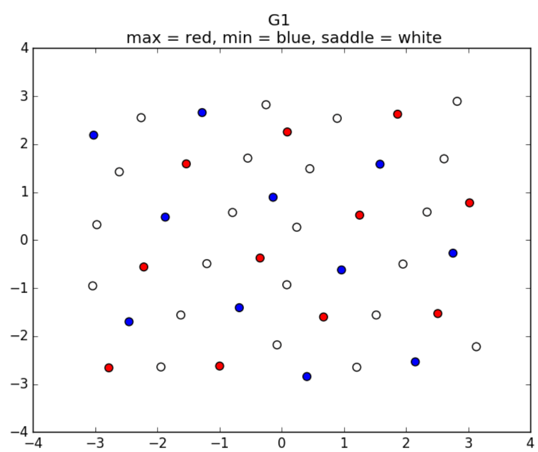
For G3

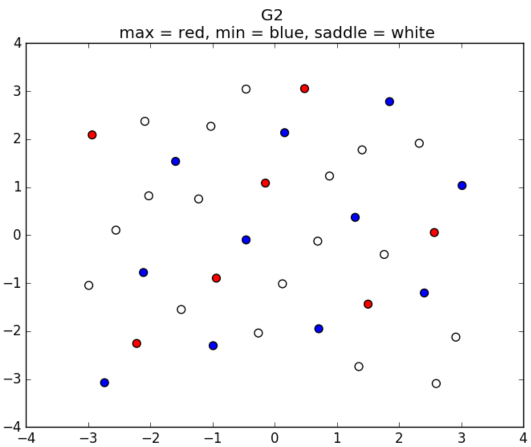
# of Guesses	Critical Points
64	43
100	62
169	85
225	97
289	100
400	109
625	122
784	122

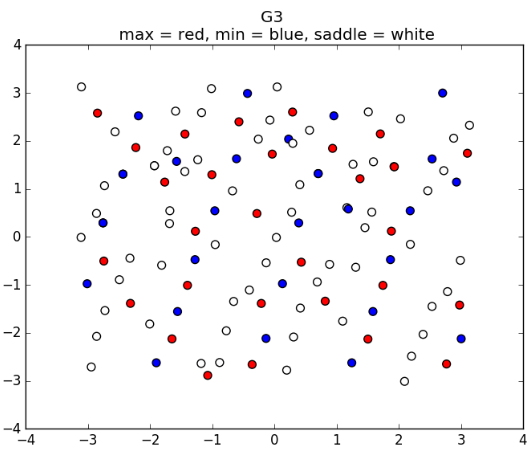
It took 625 initial guesses to find all of the critical points of G3.

I determined the type of critical point by looking at the hessian matrix and determining if the eigenvalues were positive, negative or both. if the eigenvalues are all positive the critical point is a maximum, if the eigenvalues are all negative, the critical point is a minimum and if the eigenvalues are both the critical point is a saddle point.

Here are depictions of the critical points of G1, G2 and G3







d)

Looking at the critical points and points along the border:
The location of the Global Max of G1 is (-1.86923, 0.47745)
The location of the Global Max of G2 is (3.14159, 0.98580)
The location of the Global Max of G3 is (1.20979, -2.65028)

The location of the Global Min of G1 is (1.2562, 0.51924)
The location of the Global Min of G2 is (-0.13526, 0.8913)
The location of the Global Min of G3 is (0.999127, 3.141592)

Problem 3

a)

To derive an Adams-Bashforth scheme using the relation:

$$y(t_{k+1}) = y_k + \int_{t_k}^{t_{k+1}} f(t,y)dt$$

we must evaluate the integral using a linear interpolant:

$$f(t,y) = f_{k-1} + (t - t_{k-1})\frac{f_k - f_{k-1}}{t_k - t_{k-1}}$$

$$\int_{t_k}^{t_{k+1}} f_{k-1} + (t - t_{k-1})\frac{f_k - f_{k-1}}{t_k - t_{k-1}}dt$$

$$\frac{h^2 + 2hh_{old}}{2h_{old}}f_k - \frac{h^2}{2h_{old}}f_{k-1}$$

So we have found α and β

$$y(t_{k+1}) = y_k + \frac{h^2 + 2hh_{old}}{2h_{old}}f_k - \frac{h^2}{2h_{old}}f_{k-1}$$

b)

We will solve the differential equation:

$$\begin{bmatrix} u \\ v \end{bmatrix}' = \begin{bmatrix} 0 & 7\pi\cos(\mathrm{t}) \\ -7\pi\ \cos(\mathrm{t}) & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

With initial conditions

$$\begin{bmatrix} u(0) \\ v(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

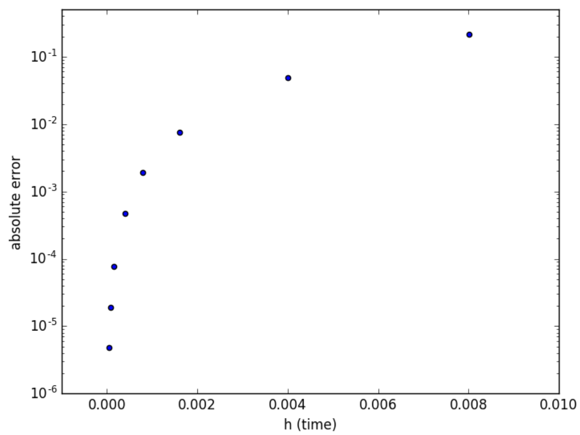
This can be solved analytically (yay Mathematica) for:

$$\begin{aligned} u(t) &= Sin(7*\pi\mathrm{Sin(t)}) \\ v(t) &= Cos(7*\pi\mathrm{Sin(t)}) \end{aligned}$$

c)

The file prob4c.py contains the script.

Absolute Error vs Step size (h)



d)

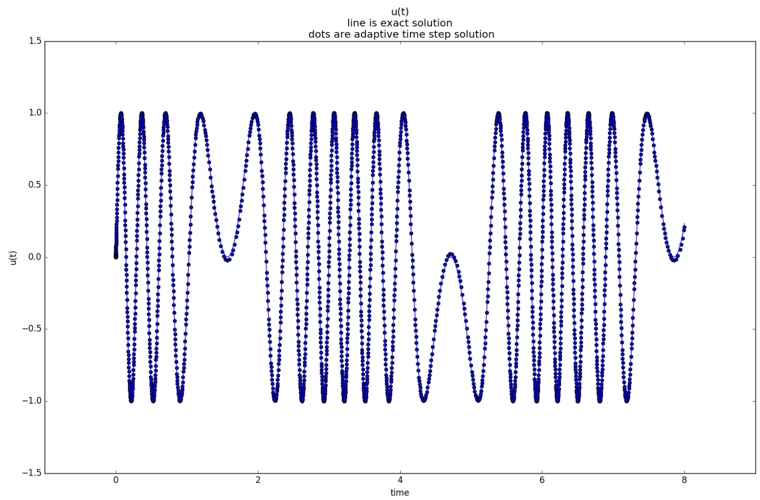
By the Runge-Kutta method using the Butcher table below:
$\widetilde{y}(t_{k+1}) = y(t_k)$ + h*($\frac{K1}{6} + \frac{2K2}{3} + \frac{K3}{6}$)

K1 = f(t,y_k)

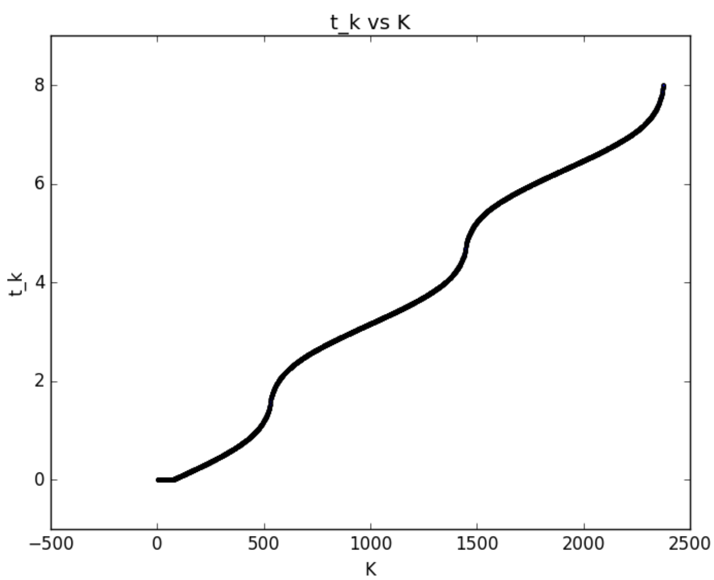
K2 = f(t+$\frac{h}{2}$, y_k + K1$\frac{h}{2}$)

K3 = f(t+h,y_k - K1*h +2 K2*h)

u(t)
both exact and numeric solution



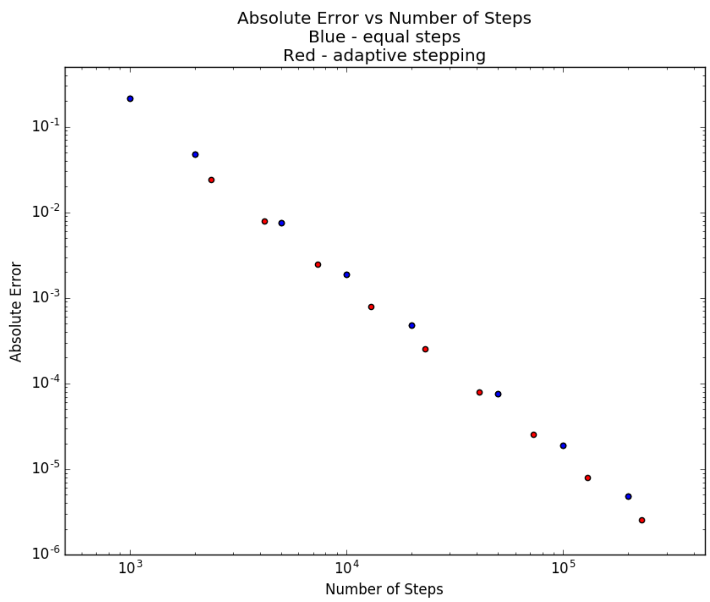
t_k vs k



Thresh	Number of Steps	Absolute Error
$10^{-1.5}$	2376	$2.408*10^{-2}$
10^{-2}	4155	$7.933*10^{-3}$
$10^{-2.5}$	7336	$2.474*10^{-3}$
10^{-3}	12980	$8.003*10^{-4}$
$10^{-3.5}$	23039	$2.513*10^{-4}$
10^{-4}	40931	$7.940*10^{-5}$
$10^{-4.5}$	72737	$2.5159*10^{-5}$
10^{-5}	129304	$7.9427*10^{-6}$
$10^{-5.5}$	229924	$2.5164*10^{-6}$
10^{-6}	408747	$7.9577*10^{-7}$

e)

Comparison c) and d)



The graph shows that the adaptive time step algorithm achieves lower absolute error for fewer steps. Thus the adaptive time step algorithm is more efficient. To quantify the increase in efficiency we can look at the number of steps it requires to achieve the same level of accuracy. We look at the 3rd cluster of points and see that to achieve 10^{-2} accuracy we need ≈ 4200 steps for the adaptive algorithm and ≈ 5000 steps for the equally spaced algorithm. This means that the adaptive time step method is $\approx 20\%$ more efficient.