Ph 20 Assignment 3

Philip Carr
Friday Section

Part 1

1. The numerical spring modelled using the explicit Euler method

Note: for all plots, $h = 0.1$, and $N = 500$.

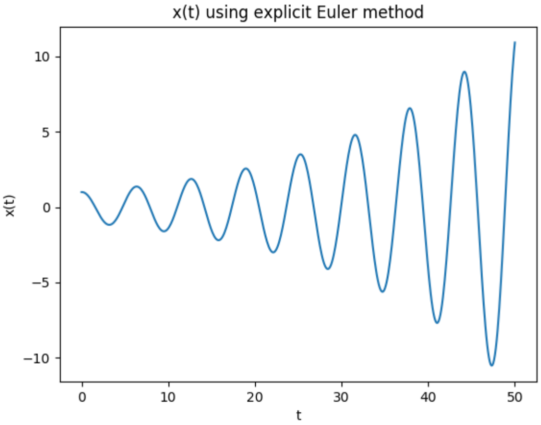

Figure 1: Plot of x(t) for the explicit Euler method.

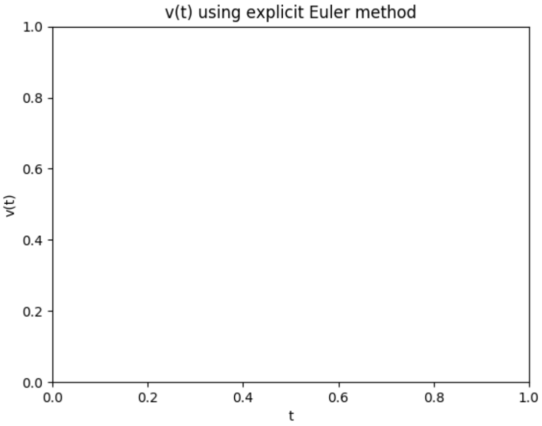

Figure 2: Plot of v(t) for the explicit Euler method.

2. Analytic solution to motion of a mass on a spring

From Newton's second law,

$$F = ma = m\frac{dv}{dt} = m\frac{d^2x}{dt^2}.$$

The force that acts on a mass on a spring is

$$F = ma = -kx.$$

Therefore,

$$F = ma = -kx = m\frac{d^2x}{dt^2} \implies m\frac{d^2x}{dt^2} + kx = 0 \implies \frac{d^2x}{dt^2} + \frac{k}{m}x = 0.$$

Thus, this is system can be represented as simple harmonic motion, where the solution to this equation is

$$x(t) = A\cos(\omega t),$$

where A is the amplitude of the motion of the mass, and ω is the frequency of oscillation. $\omega = \sqrt{\frac{k}{m}}$. Let $A = 1$, and $\frac{k}{m} = 1$ and thus $\omega = 1$. Thus, the solution for $x(t)$ becomes

$$x(t) = \cos(t).$$

Thus,

$$v(t) = -\sin(t)$$

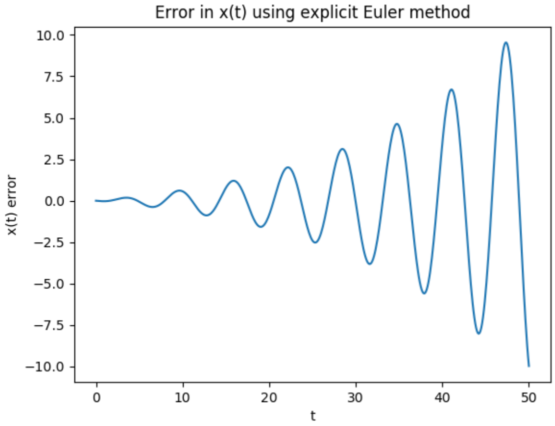

Figure 3: Plot of global error of $x(t)$ for the explicit Euler method.

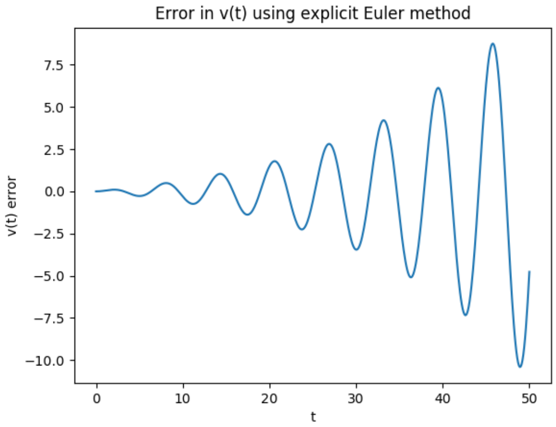

Figure 4: Plot of global error of $v(t)$ for the explicit Euler method.

3. Truncation error

As shown in the two plots below, truncation error is proportional to h for reasonably small values of h.

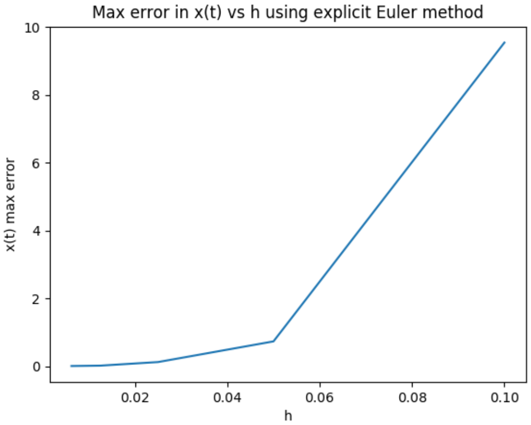

Figure 5: Plot of truncation error of $x(t)$ for the explicit Euler method for $h = h_0,\ h_0/2,\ h_0/4,\ h_0/8,\ h_0/16$, where $h_0 = 0.1$.

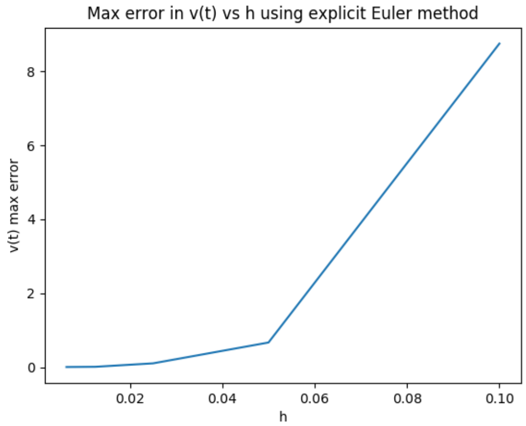

Figure 6: Plot of truncation error of $v(t)$ for the explicit Euler method for $h = h_0,\ h_0/2,\ h_0/4,\ h_0/8,\ h_0/16$, where $h_0 = 0.1$.

4. Total energy

Below is a plot of the normalized total energy $E = x^2 + v^2$ of the explicit Euler method as a function of time. The long-range trend for E is that E increases as time increases. The total energy and the absolute value of the global errors both increase as time goes on.

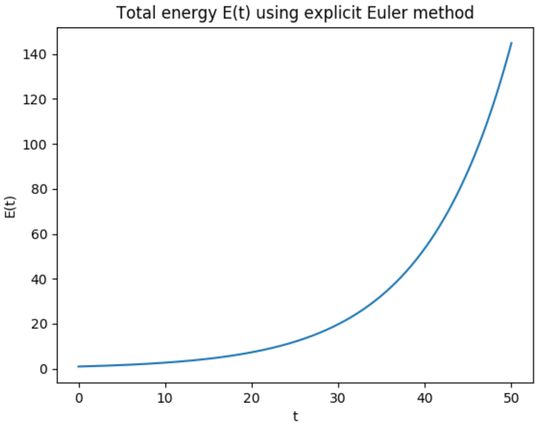

Figure 7: Plot of normalized total energy $E(t)$ for the explicit Euler method.

5. Implicit Euler method

The implicit Euler method uses the system of equations

$$\begin{bmatrix} 1 & -h \\ h & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{i+1} \\ v_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ v_i \end{bmatrix}$$

$$\implies x_{i+1} = x_i - hv_{i+1},\ v_{i+1} = v_i + hx_{i+1}$$

$$\implies x_{i+1} = x_i - h(v_i + hx_{i+1}) \implies x_{i+1} = x_i - hv_i - h^2 x_{i+1} \implies (1+h^2)x_{i+1} = x_i - hv_i$$

$$\implies x_{i+1} = \frac{x_i - hv_i}{1+h^2}.$$

$$v_{i+1} = v_i + hx_{i+1} = v_i + h(x_i - hv_{i+1}) \implies v_{i+1} = v_i + hx_i - h^2 v_{i+1} \implies (1+h^2)v_{i+1} = v_i + hx_i$$

$$\implies v_{i+1} = \frac{v_i + hx_i}{1+h^2}.$$

Below are plots for global errors in $x(t)$ and $v(t)$ for the implicit Euler method. For the implicit Euler method, the global error in $x(t)$ is greater in magnitude than that of the explicit Euler method for t close to 0, but increases in magnitude slower than the explicit Euler method. The global error of $v(t)$ for the implicit Euler method is greater in magnitude than that of the explicit Euler method for t close to 0, and then the magnitude of the implicit Euler method decreases instead of increases as seen in the explicit Euler method. The total normalized energy of the system using the implicit method decreases instead of increases as it does for the explicit Euler method.

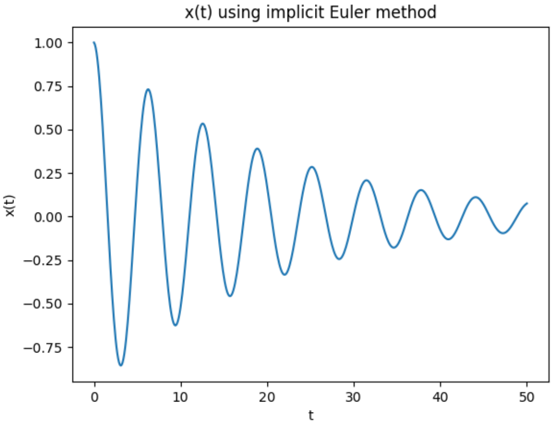

Figure 8: Plot of x(t) for the implicit Euler method.

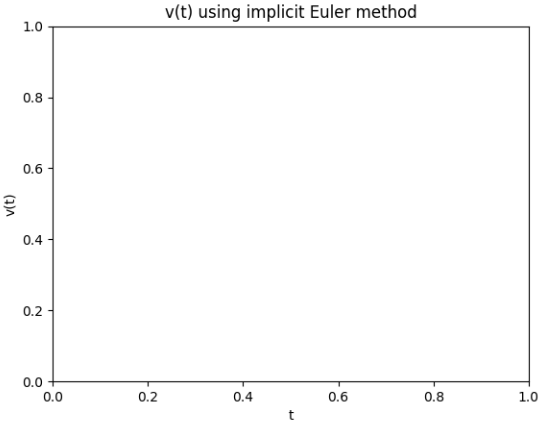

Figure 9: Plot of v(t) for the implicit Euler method.

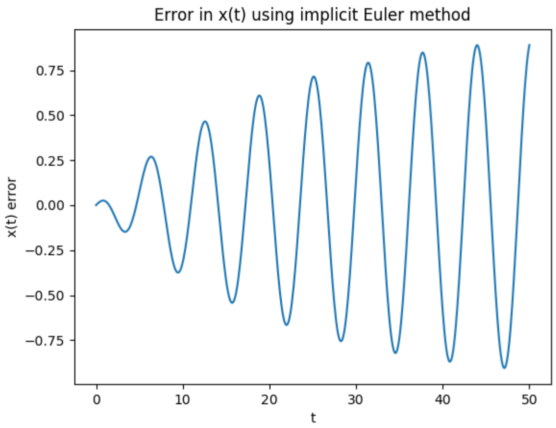

Figure 10: Plot of global error of $x(t)$ for the implicit Euler method.

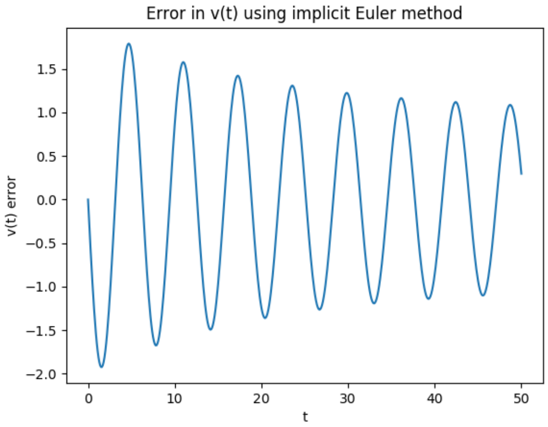

Figure 11: Plot of global error of $v(t)$ for the implicit Euler method.

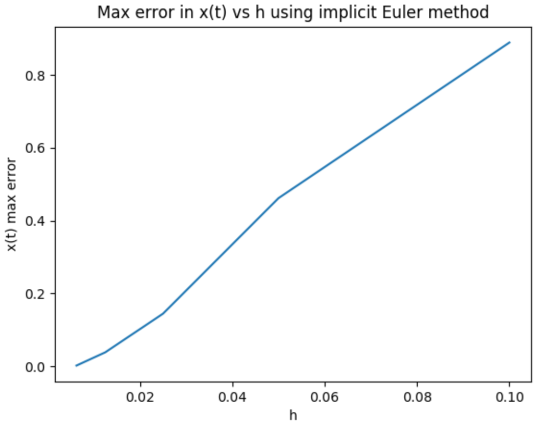

Figure 12: Plot of truncation error of $x(t)$ for the implicit Euler method for $h = h_0,\ h_0/2,\ h_0/4,\ h_0/8,\ h_0/16$, where $h_0 = 0.1$.

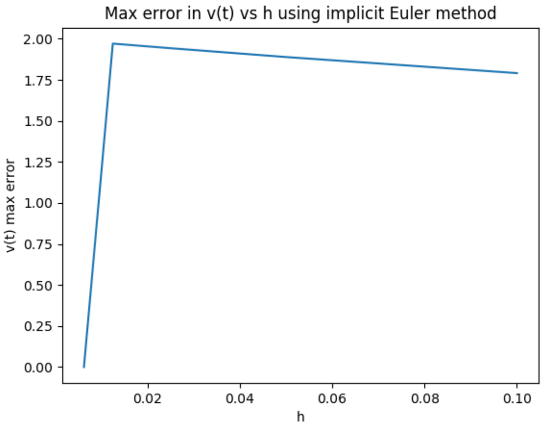

Figure 13: Plot of truncation error of $v(t)$ for the implicit Euler method for $h = h_0,\ h_0/2,\ h_0/4,\ h_0/8,\ h_0/16,$, where $h_0 = 0.1$.

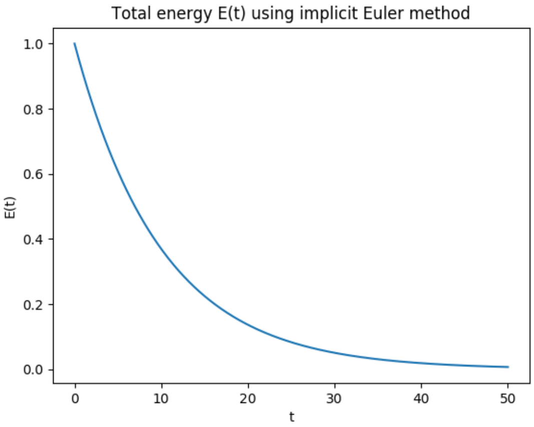

Figure 14: Plot of normalized total energy $E(t)$ for the explicit Euler method.

Part 2

1. Phase-space geometry of trajectories produced by the explicit and implicit Euler methods.

Below are plots of the phase space geometries of the explicit and implicit Euler methods.

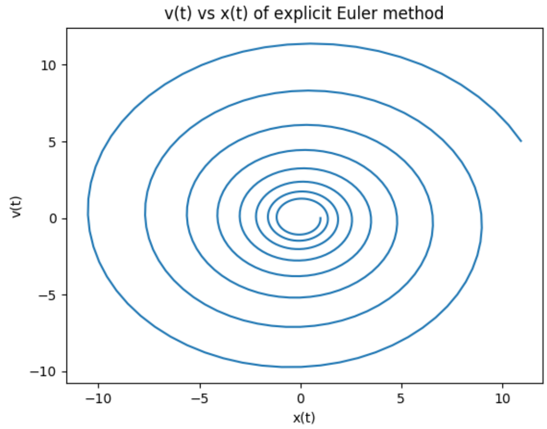

Figure 15: Plot of the phase space geometry for the explicit Euler method.

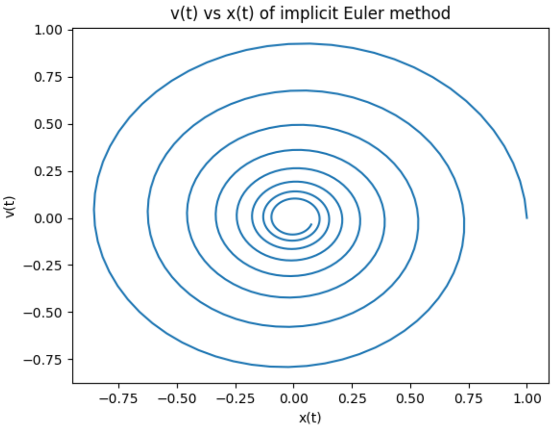

Figure 16: Plot of the phase space geometry for the implicit Euler method.

2. Symplectic Euler method phase-space geometry

Below are plots of the phase-space geometry of the symplectic method alone and a plot comparing the phase-space geometries of all three Euler methods at once.

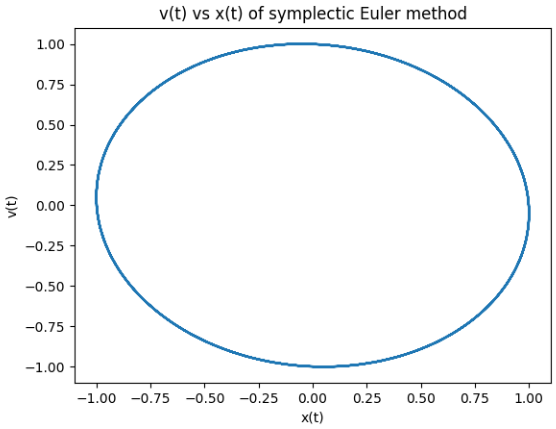

Figure 17: Plot of phase space geometry for the symplectic Euler method.

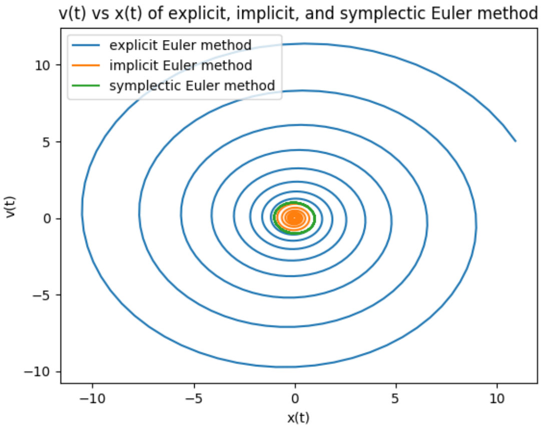

Figure 18: Plot of phase space geometries for the explicit, implicit, and symplectic Euler methods.

3. Total energy obtained with the symplectic Euler method

Below is a plot of the total energy obtained with the symplectic Euler method. $h = 0.1$, and $N = 500$. The deviations from the constant value of total energy are sinusoidal in nature. This evolution relates to what is seen in phase space in the slight distortion of the phase space geometry of the symplectic Euler method from being a perfect circle.

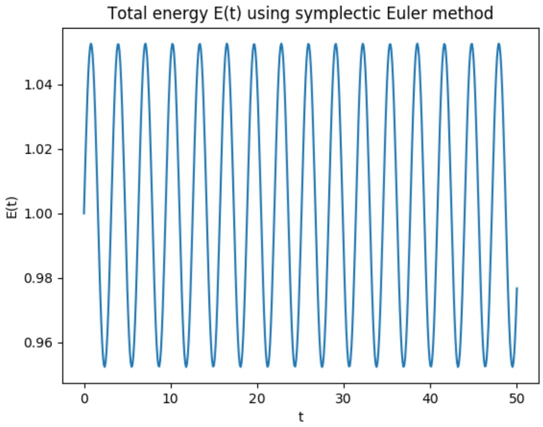

Figure 19: Plot of normalized total energy $E(t)$ for the symplectic Euler method.

Version Control Log

```
commit 0c7c3f5fc89fdcfebf9ad0c3aa92090c81222921
Author: pcarr <pcarr@caltech.edu>
Date:   Mon Jun 12 13:19:01 2017 -0700

    Modified Makefile and ph20_hw3.tex

    Modified Makefile to print git log to correct output file
    (ph_20_assignment_4_git_log.txt)

    Modified ph20_hw3.tex to add sections of Version Control Log, Makefile
    Source Code, ph20_hw3_updated.py Source Code, and Command-line Output

commit dbe311afdd20da707573c1edd83c67e04e22ae11
Author: pcarr <pcarr@caltech.edu>
Date:   Mon Jun 12 13:09:15 2017 -0700

    Modified Makefile to print command-line output to the file commandLineOutput.txt

commit 467296b7fda8e33a3d87442cc2b0f3cd5df18a7a
Author: pcarr <pcarr@caltech.edu>
Date:   Mon Jun 12 04:18:34 2017 -0700

    Modified file to process one plot at a time

commit e8c10908e114fabc162406c0478488b15e522e96
Author: pcarr <pcarr@caltech.edu>
Date:   Mon Jun 12 04:17:34 2017 -0700

    Modified plot title input names

commit 81be5b2077ea6f957307d01e2a33fa1425d96f5c
Author: pcarr <pcarr@caltech.edu>
Date:   Mon Jun 12 04:15:50 2017 -0700
```

```
    Implemented pattern rule for processing python plot outputs

commit cb2f13da6a3b2ff7492b8d313078d0b05e1f7856
Author: pcarr <pcarr@caltech.edu>
Date:   Mon Jun 12 02:20:23 2017 -0700

    Created .tex file to generate completed Ph20 assignment 3 pdf file

commit 14b1c57ca28e9bf05d4ba8862d4b49918aa5eca5
Author: pcarr <pcarr@caltech.edu>
Date:   Mon Jun 12 02:18:57 2017 -0700

    Created Makefile to generate completed Ph20 assignment 3

commit 126e1d9d4259266ff7ad57108c60c2c89f44eee0
Author: pcarr <pcarr@caltech.edu>
Date:   Fri Jun 9 15:53:18 2017 -0700

    Added print statements to main() function

commit 381995b2b44b9c19246164cd55461f8b16af71d0
Author: pcarr <pcarr@caltech.edu>
Date:   Fri Jun 9 15:47:04 2017 -0700

    Ph 20 Homework 3. Numerical solutions for the simple harmonic oscillator
```

Makefile Source Code

```
generate_assignment_3 : ph20_hw3.tex plot_1.png plot_2.png plot_3.png \
plot_4.png plot_5.png plot_6.png plot_7.png plot_8.png plot_9.png plot_10.png \
plot_11.png plot_12.png plot_13.png plot_14.png plot_15.png plot_16.png \
plot_17.png plot_18.png plot_19.png
touch commandLineOutput.txt >> commandLineOutput.txt
git log > ph_20_assignment_4_git_log.txt
cp Makefile makefile_source_code.txt >> commandLineOutput.txt
cp ph20_hw3_updated.py hw3_source_code.txt >> commandLineOutput.txt
pdflatex ph20_hw3.tex >> commandLineOutput.txt

%.png : ph20_hw3_updated.py
touch commandLineOutput.txt >> commandLineOutput.txt
mkdir -p plots >> commandLineOutput.txt
python ph20_hw3_updated.py $*.png >> commandLineOutput.txt

.PHONY : clean
clean :
rm -f ph20_hw3.pdf
rm -f ph20_hw3.aux
rm -f ph20_hw3.log
rm -f ph_20_assignment_4_git_log.txt
rm -f makefile_source_code.txt
rm -f hw3_source_code.txt
rm -f commandLineOutput.txt
rm -f ph20_hw3.synctex.gz
rm -r -f plots
```

ph20_hw3_updated.py Source Code

```
# Philip Carr
# Friday Section
# Ph 20 Homework 3
# 5/2/2017

import numpy as np
import matplotlib.pyplot as plt
import sys

figure_number = 1

# Part 1
# 1.
def plotPositionAndVelocity(xList, vList, tList, numericalType):
    global figure_number
    if figure_number in [1, 8]:
        plt.plot(tList, xList)
        plt.xlabel('t')
        plt.ylabel('x(t)')
        plt.title('x(t) using ' + numericalType + ' Euler method')
        plt.savefig('plots/figure_' + str(figure_number) + '.png')
        plt.plot(tList, vList)
    elif figure_number in [2, 9]:
        plt.xlabel('t')
        plt.ylabel('v(t)')
        plt.title('v(t) using ' + numericalType + ' Euler method')
        plt.savefig('plots/figure_' + str(figure_number) + '.png')

def explicitEuler(h):
    N = 500
    tList = np.zeros(N + 1)
    xList = np.zeros(N + 1)
    vList = np.zeros(N + 1)
    xList[0] = 1
    vList[0] = 0
    for i in range(1, N + 1):
        tList[i] = tList[i-1] + h
        xList[i] = xList[i-1] + h * vList[i-1]
        vList[i] = vList[i-1] - h * xList[i-1]
    return xList, vList, tList

#2.
def plotEulerError(xErrorList, vErrorList, tList, numericalType):
    global figure_number
    if figure_number in [3, 10]:
        plt.plot(tList, xErrorList)
        plt.xlabel('t')
        plt.ylabel('x(t) error')
        plt.title('Error in x(t) using ' + numericalType + ' Euler method')
        plt.savefig('plots/figure_' + str(figure_number) + '.png')
    elif figure_number in [4, 11]:
        plt.plot(tList, vErrorList)
        plt.xlabel('t')
        plt.ylabel('v(t) error')
        plt.title('Error in v(t) using ' + numericalType + ' Euler method')
        plt.savefig('plots/figure_' + str(figure_number) + '.png')

def getEulerError(xList, vList, tList):
    xAnalyticList = tList[:]
    vAnalyticList = tList[:]
    xAnalyticList = np.cos(xAnalyticList)
    vAnalyticList = -np.sin(vAnalyticList)
    xErrorList = xAnalyticList - xList
    vErrorList = vAnalyticList - vList
    return xErrorList, vErrorList

# 3.
def plotTruncationError(numericalType):
    global figure_number
    h = 0.1
    i = 1
    max_xErrorList = np.zeros(5)
    max_vErrorList = np.zeros(5)
    hList = np.zeros(5)
    count = 0
    while i <= 16:
        if numericalType == 'explicit':
            xList, vList, tList = explicitEuler(h)
        elif numericalType == 'implicit':
            xList, vList, tList = implicitEuler(h)
        else:
            raise ValueError('numericalType needed!')
        xErrorList, vErrorList = getEulerError(xList, vList, tList)
        hList[count] = h
        max_xErrorList[count] = max(xErrorList)
        max_vErrorList[count] = max(vErrorList)
        i *= 2
        h /= 2
        count += 1
    if figure_number in [5, 12]:
        plt.plot(hList, max_xErrorList)
        plt.xlabel('h')
        plt.ylabel('x(t) max error')
        plt.title('Max error in x(t) vs h using ' + numericalType + ' Euler method')
        plt.savefig('plots/figure_' + str(figure_number) + '.png')
    if figure_number in [6, 13]:
        plt.plot(hList, max_vErrorList)
        plt.xlabel('h')
        plt.ylabel('v(t) max error')
        plt.title('Max error in v(t) vs h using ' + numericalType + ' Euler method')
        plt.savefig('plots/figure_' + str(figure_number) + '.png')

# 4.
```

```
def plotTotalEnergy(xList, vList, tList, numericalType):
    global figure_number
    totalEnergyList = np.power(xList, 2) + np.power(vList, 2)
    plt.plot(tList, totalEnergyList)
    plt.xlabel('t')
    plt.ylabel('E(t)')
    plt.title('Total energy E(t) using ' + numericalType + ' Euler method')
    plt.savefig('plots/figure_' + str(figure_number) + '.png')

# 5.
def implicitEuler(h):
    N = 500
    tList = np.zeros(N + 1)
    xList = np.zeros(N + 1)
    vList = np.zeros(N + 1)
    xList[0] = 1
    vList[0] = 0
    for i in range(1, N + 1):
        tList[i] = tList[i-1] + h
        xList[i] = (xList[i-1] - h * vList[i-1]) / (1 + np.power(h, 2))
        vList[i] = (vList[i-1] + h * xList[i-1]) / (1 + np.power(h, 2))
    return xList, vList, tList

# Part 2
# 1.
def phaseSpacePlot(xList, vList, numericalType):
    global figure_number
    plt.plot(xList, vList)
    plt.xlabel('x(t)')
    plt.ylabel('v(t)')
    plt.title('v(t) vs x(t) of ' + numericalType + ' Euler method')
    plt.savefig('plots/figure_' + str(figure_number) + '.png')

# 2.
def phaseSpacePlotAllThreeEulerMethods(allLists):
    global figure_number
    plt.plot(allLists[0], allLists[1], label='explicit Euler method')
    plt.plot(allLists[3], allLists[4], label='implicit Euler method')
    plt.plot(allLists[6], allLists[7], label='symplectic Euler method')
    plt.xlabel('x(t)')
    plt.ylabel('v(t)')
    plt.title('v(t) vs x(t) of explicit, implicit, and symplectic Euler method')
    plt.legend()
    plt.savefig('plots/figure_' + str(figure_number) + '.png')

def symplecticEuler(h):
    N = 500
    tList = np.zeros(N + 1)
    xList = np.zeros(N + 1)
    vList = np.zeros(N + 1)
    xList[0] = 1
    vList[0] = 0
    for i in range(1, N + 1):
        tList[i] = tList[i-1] + h
        xList[i] = xList[i-1] + h * vList[i-1]
        vList[i] = vList[i-1] - h * xList[i]
    return xList, vList, tList

def main():
    global figure_number
    figure_number = int(sys.argv[1][sys.argv[1].index(’_’)+1:-4])
    h = 0.1
    xListE, vListE, tListE = explicitEuler(h)
    xListI, vListI, tListI = implicitEuler(h)
    xListS, vListS, tListS = symplecticEuler(h)
    xErrorListE, vErrorListE = getEulerError(xListE, vListE, tListE)
    xErrorListI, vErrorListI = getEulerError(xListI, vListI, tListI)
    if figure_number in [1, 2]:
        plotPositionAndVelocity(xListE, vListE, tListE, ’explicit’)
    elif figure_number in [3, 4]:
        plotEulerError(xErrorListE, vErrorListE, tListE, ’explicit’)
    elif figure_number in [5, 6]:
        plotTruncationError(’explicit’)
    elif figure_number in [7]:
        plotTotalEnergy(xListE, vListE, tListE, ’explicit’)
    elif figure_number in [8, 9]:
        plotPositionAndVelocity(xListI, vListI, tListI, ’implicit’)
    elif figure_number in [10, 11]:
        plotEulerError(xErrorListI, vErrorListI, tListI, ’implicit’)
    elif figure_number in [12, 13]:
        plotTruncationError(’implicit’)
    elif figure_number in [14]:
        plotTotalEnergy(xListI, vListI, tListI, ’implicit’)
    elif figure_number in [15]:
        phaseSpacePlot(xListE, vListE, ’explicit’)
    elif figure_number in [16]:
        phaseSpacePlot(xListI, vListI, ’implicit’)
    elif figure_number in [17]:
        phaseSpacePlot(xListS, vListS, ’symplectic’)
    elif figure_number in [18]:
        allLists = [xListE, vListE, tListE, xListI, vListI, tListI
                    , xListS, vListS, tListS]
        phaseSpacePlotAllThreeEulerMethods(allLists)
    elif figure_number in [19]:
        plotTotalEnergy(xListS, vListS, tListS, ’symplectic’)

if __name__ == ’__main__’: main()
```

Command-line Output

```
This is pdfTeX, Version 3.14159265-2.6-1.40.17 (TeX Live 2016/Debian) (preloaded format=pdflatex)
 restricted \write18 enabled.
entering extended mode
(./ph20_hw3.tex
LaTeX2e <2017/01/01> patch level 3
Babel <3.9r> and hyphenation patterns for 83 language(s) loaded.
(/usr/share/texlive/texmf-dist/tex/latex/base/article.cls
Document Class: article 2014/09/29 v1.4h Standard LaTeX document class
(/usr/share/texlive/texmf-dist/tex/latex/base/size10.clo))
(/usr/share/texlive/texmf-dist/tex/latex/base/inputenc.sty
(/usr/share/texlive/texmf-dist/tex/latex/base/utf8.def
(/usr/share/texlive/texmf-dist/tex/latex/base/t1enc.dfu)
(/usr/share/texlive/texmf-dist/tex/latex/base/ot1enc.dfu)
(/usr/share/texlive/texmf-dist/tex/latex/base/omsenc.dfu)))
(/usr/share/texlive/texmf-dist/tex/latex/graphics/graphicx.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics/keyval.sty)
(/usr/share/texlive/texmf-dist/tex/latex/graphics/graphics.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics/trig.sty)
(/usr/share/texlive/texmf-dist/tex/latex/graphics-cfg/graphics.cfg)
(/usr/share/texlive/texmf-dist/tex/latex/graphics-def/pdftex.def
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/infwarerr.sty)
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ltxcmds.sty))))
(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/amssymb.sty
(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/amsfonts.sty))
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsmath.sty
For additional information on amsmath, use the ‘?’ option.
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amstext.sty
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsgen.sty))
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsbsy.sty)
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsopn.sty))
(/usr/share/texlive/texmf-dist/tex/latex/float/float.sty)
(/usr/share/texlive/texmf-dist/tex/latex/caption/caption.sty
(/usr/share/texlive/texmf-dist/tex/latex/caption/caption3.sty))
(/usr/share/texlive/texmf-dist/tex/latex/tools/verbatim.sty)
(/usr/share/texlive/texmf-dist/tex/latex/geometry/geometry.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ifpdf.sty)
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ifvtex.sty)
(/usr/share/texlive/texmf-dist/tex/generic/ifxetex/ifxetex.sty))
No file ph20_hw3.aux.
(/usr/share/texlive/texmf-dist/tex/context/base/mkii/supp-pdf.mkii
[Loading MPS to PDF converter (version 2006.09.02).]
) (/usr/share/texlive/texmf-dist/tex/generic/oberdiek/pdftexcmds.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ifluatex.sty))
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/epstopdf-base.sty
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/grfext.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/kvdefinekeys.sty))
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/kvoptions.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/kvsetkeys.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/etexcmds.sty)))
(/usr/share/texlive/texmf-dist/tex/latex/latexconfig/epstopdf-sys.cfg))
*geometry* driver: auto-detecting
*geometry* detected driver: pdftex
(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/umsa.fd)
(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/umsb.fd)
<plots/figure_1.png, id=1, 462.528pt x 346.896pt> <use plots/figure_1.png>
<plots/figure_2.png, id=3, 462.528pt x 346.896pt> <use plots/figure_2.png>
[1{/var/lib/texmf/fonts/map/pdftex/updmap/pdftex.map} <./plots/figure_1.png> <.
/plots/figure_2.png>] <plots/figure_3.png, id=18, 462.528pt x 346.896pt>
<use plots/figure_3.png> <plots/figure_4.png, id=19, 462.528pt x 346.896pt>
<use plots/figure_4.png> [2 <./plots/figure_3.png>]
<plots/figure_5.png, id=28, 462.528pt x 346.896pt> <use plots/figure_5.png>
```

```
<plots/figure_6.png, id=29, 462.528pt x 346.896pt> <use plots/figure_6.png>
[3 <./plots/figure_4.png> <./plots/figure_5.png>]
<plots/figure_7.png, id=36, 462.528pt x 346.896pt> <use plots/figure_7.png>
[4 <./plots/figure_6.png> <./plots/figure_7.png>]
<plots/figure_8.png, id=43, 462.528pt x 346.896pt> <use plots/figure_8.png>
<plots/figure_9.png, id=44, 462.528pt x 346.896pt> <use plots/figure_9.png>
<plots/figure_10.png, id=45, 462.528pt x 346.896pt> <use plots/figure_10.png>
[5 <./plots/figure_8.png> <./plots/figure_9.png>]
<plots/figure_11.png, id=51, 462.528pt x 346.896pt> <use plots/figure_11.png>
<plots/figure_12.png, id=52, 462.528pt x 346.896pt> <use plots/figure_12.png>
Underfull \hbox (badness 5726) in paragraph at lines 139--139
[]\OT1/cmr/bx/n/9 Figure 12: |\OT1/cmr/m/n/9 Plot of trun-ca-tion er-ror of $\O
ML/cmm/m/it/9 x\OT1/cmr/m/n/9 (\OML/cmm/m/it/9 t\OT1/cmr/m/n/9 )$ for the im-pl
icit Eu-ler method for $\OML/cmm/m/it/9 h \OT1/cmr/m/n/9 =
<plots/figure_13.png, id=53, 462.528pt x 346.896pt> <use plots/figure_13.png>
Underfull \hbox (badness 5802) in paragraph at lines 145--145
[]\OT1/cmr/bx/n/9 Figure 13: |\OT1/cmr/m/n/9 Plot of trun-ca-tion er-ror of $\O
ML/cmm/m/it/9 v\OT1/cmr/m/n/9 (\OML/cmm/m/it/9 t\OT1/cmr/m/n/9 )$ for the im-pl
icit Eu-ler method for $\OML/cmm/m/it/9 h \OT1/cmr/m/n/9 =
[6 <./plots/figure_10.png> <./plots/figure_11.png> <./plots/figure_12.png>]
<plots/figure_14.png, id=60, 462.528pt x 346.896pt> <use plots/figure_14.png>
<plots/figure_15.png, id=61, 462.528pt x 346.896pt> <use plots/figure_15.png>
[7 <./plots/figure_13.png> <./plots/figure_14.png>]
<plots/figure_16.png, id=68, 462.528pt x 346.896pt> <use plots/figure_16.png>
<plots/figure_17.png, id=69, 462.528pt x 346.896pt> <use plots/figure_17.png>
[8 <./plots/figure_15.png> <./plots/figure_16.png>]
<plots/figure_18.png, id=75, 462.528pt x 346.896pt> <use plots/figure_18.png>
<plots/figure_19.png, id=76, 462.528pt x 346.896pt> <use plots/figure_19.png>
[9 <./plots/figure_17.png> <./plots/figure_18.png>]
Overfull \hbox (7.37616pt too wide) in paragraph at lines 196--196
[]    \OT1/cmtt/m/n/10 Modified Makefile to print command-line output to the fi
le commandLineOutput.txt[]
[10 <./plots/figure_19.png>] [11] [12]
Overfull \hbox (7.37616pt too wide) in paragraph at lines 202--202
[]        \OT1/cmtt/m/n/10 plt.title('Max error in x(t) vs h using ' + numerica
lType + ' Euler method')[]

Overfull \hbox (7.37616pt too wide) in paragraph at lines 202--202
[]        \OT1/cmtt/m/n/10 plt.title('Max error in v(t) vs h using ' + numerica
lType + ' Euler method')[]
[13] [14]
Overfull \hbox (75.62556pt too wide) in paragraph at lines 205--205
[]\OT1/cmtt/m/n/10 This is pdfTeX, Version 3.14159265-2.6-1.40.17 (TeX Live 201
6/Debian) (preloaded format=pdflatex)
[15] [16]
```