Ph 20
Problem Set 3

Carson Adams

Part 1

Exercise 1

See Figure 1. Notice that the amplitude of the oscillations increase for both position and velocity (while these should remain constant for SHM). However, it does appear that these quantities remain out of phase (that is, when $x = 0$, the velocity is at an extrema).

Exercise 2

We solve the differential equation with general initial conditions t_0, x_0, v_0 and arrive at

$$x(t) = x_0 \cos(t - t_0) + v_0 \sin(t - t_0) \tag{1}$$

$$v(t) = v_0 \cos(t - t_0) - x_0 \sin(t - t_0) \tag{2}$$

For our chosen initial conditions of $\{t_0 = 0, x_0 = 1, v_0 = 0\}$, this gives $x(t) = x_0 \cos(t)$ and $v(t) = -x_0 \sin(t)$.

In Figure 1, we plot the global error $x(t_i) - x_i(t_i)$ and $v(t_i) - v_i(t_i)$ of our numerical estimate from the analytic solution above.

Exercise 3

For small step size $h \lesssim 0.5$, the truncation error is approximately linear in h (where the truncation error is defined as the maximum value of $|x(t_i) - x_i|$). See Figure 1.

Exercise 4

Numerically, the total energy in the system grows in time and tends towards infinity (see Figure 2). This agrees with the evolution of the global errors which show that v^2 and x^2 are increasingly overestimated (see Figure 1).

Exercise 5

Solving the implicit Euler equations,

$$x_{i+1} = \frac{x_i + hv_i}{1 + h^2} \tag{3}$$

$$v_{i+1} = \frac{v_i - hx_i}{1 + h^2} \tag{4}$$

We observe that this method systematically underestimates the position and velocity of oscillator. Unlike the explicit Euler method, the global errors produced by the implicit method do not grow without bound (as the oscillator trends towards zero motion). Similarly the total energy approaches zero over time.

Part 2

Exercises 1 & 2

The phase-space geometries of the explicit and implicit Euler methods are not closed shapes. The explicit solution spirals *out* while the implicit solution spirals *in*, as expected. However, the symplectic method is indeed a closed loop. Comparing it to the true circular trajectory, it appears that this estimate 'wobbles'; that is, the symplectic phase-space trajectory appears to be some closed ellipse. Physically, this conservation of phase-space volume is much more 'realistic'.

Exercise 3

The total energy estimated by the symplectic Euler method does not grow without bound or dissipate to zero, but instead oscillates around the true constant energy at twice the frequency of the oscillator itself. This is exactly as expected from the phase-space diagram which shows the symplectic trajectory deviating 'in' and 'out' twice per cycle (i.e. the ellipse crosses the true circular path a total of four times per full motion).

Assignment 4

Makefile

```
all : plot pdf
.PHONY : all

.PHONY : plot
plot : img/

img/ : Ph20_Set_3.ipynb
rm -rf $@
mkdir $@
jupyter-2.7 nbconvert --execute --allow-errors --to notebook --inplace Ph20_Set_3.ipynb

Ph20_Set_3.py : Ph20_Set_3.ipynb
jupyter nbconvert --to script Ph20_Set_3.ipynb

.PHONY : pdf
pdf : Ph20_Set_3.pdf
Ph20_Set_3.pdf : Ph20_Set_3.tex git.log Ph20_Set_3.py
pdflatex $<
pdflatex $<
rm -f *.log
rm -f *.aux
rm -f *.py

git.log :
git log > git.log

Ph20_Set_3.py : Ph20_Set_3.ipynb
jupyter-2.7 nbconvert --to script  Ph20_Set_3.ipynb

.PHONY : clean
```

```
clean :
rm -rf img
rm -r *.pdf
rm -f *.py
```

Version control

```
commit e98a68cdd5fc132a3aebb350e69005daea94ca15
Author: CarsonAdams <carsonadams@me.com>
Date:   Wed Nov 8 02:44:30 2017 -0800

    Trivial change

commit dcf61c303bc4c1334c3d5749a1969922ba0726bf
Author: CarsonAdams <carsonadams@me.com>
Date:   Wed Nov 8 02:34:11 2017 -0800

    Post .gitignore changes

commit 617f7cd6fe6ec4fbc68a522421da0330916a8261
Author: CarsonAdams <carsonadams@me.com>
Date:   Wed Nov 8 02:33:51 2017 -0800

    Pre .gitignore changes

commit 06a29be82cc58077b0bb76555613b6951e548206
Author: CarsonAdams <carsonadams@me.com>
Date:   Wed Nov 8 02:31:51 2017 -0800

    Post .gitignore changes

commit 8d0ff1b3c5e104cc1bb59fa85cd7d47d672776d4
Author: CarsonAdams <carsonadams@me.com>
Date:   Wed Nov 8 02:22:39 2017 -0800

    first commit
```

Source

```
# coding: utf-8

# In[1]:


# TRIVIAL CHANGE
# YES MORE CHANGE
import numpy as np
import matplotlib.pyplot as plt
get_ipython().magic(u'matplotlib inline')
import matplotlib.ticker as mtick
from matplotlib import rc
```

```
rc('font', **{'family': 'serif', 'serif': ['Times New Roman'], 'size': 14})
rc('text', usetex=True)


# In[2]:


# Exercise 1.1
def eulerExp(x0=1., v0=0., t0=0., P=5, h=0.1):
    t = np.arange(t0, t0+P*2*np.pi+h, h)
    N = len(t)
    x = np.zeros(N)
    v = np.zeros(N)
    x[0], v[0] = x0, v0
    for i in range(N-1):
        x[i+1] = x[i] + h*v[i]
        v[i+1] = v[i] - h*x[i]
    return (t, x, v)


# In[3]:


# Exercise 1.1
t, x, v = eulerExp()
plt.plot(t, x, label='x')
plt.plot(t, v, label='v')
plt.legend()
plt.title('Explicit Euler: SHM')
plt.xlabel('Time')
plt.savefig('img/p1_1.pdf')


# In[4]:


# Analytic SHM
def SHM(x0=1., v0=0., t0=0., P=5, h=0.1):
    t = np.arange(t0, t0+P*2*np.pi+h, h)
    x = x0*np.cos(t-t0) + v0*np.sin(t-t0)
    v = v0*np.cos(t-t0) - x0*np.sin(t-t0)
    return (t, x, v)


# In[5]:


# Exercise 1.2
t, x, v = eulerExp()
T, X, V = SHM()
plt.plot(t, X-x, label='$x_{\\text{true}}-x$')
```

```
plt.plot(t, V-v, label='$v_{\\text{true}}-v$')
plt.legend()
plt.title('Explicit Euler: Global error')
plt.xlabel('Time')
plt.savefig('img/p1_2.pdf')


# In[6]:


# Exercise 1.3
h0 = 0.1
h = h0/np.power(2, np.arange(5))
err = np.zeros(np.size(h))
for i in range(len(h)):
    err[i] = np.max(np.abs(SHM(h=h[i])[1]-eulerExp(h=h[i])[1]))
plt.plot(h, err)
plt.title('Explicit Euler: Truncation error')
plt.xlabel('Step size (h)')
plt.ylabel('Error')
plt.savefig('img/p1_3.pdf')


# In[7]:


# Exercise 1.4
t, x, v = eulerExp()
E = np.power(x, 2) + np.power(v, 2)
plt.plot(t, E)
plt.title('Explicit Euler: Energy')
plt.xlabel('Time')
plt.ylabel('Energy')
plt.savefig('img/p1_4.pdf')


# In[8]:


# Exercise 1.5
def eulerImp(x0=1., v0=0., t0=0., P=5, h=0.1):
    t = np.arange(t0, t0+P*2*np.pi+h, h)
    N = len(t)
    x = np.zeros(N)
    v = np.zeros(N)
    x[0], v[0] = x0, v0
    for i in range(N-1):
        x[i+1] = (x[i] + h*v[i])/(1. + h**2)
        v[i+1] = (v[i] - h*x[i])/(1. + h**2)
    return (t, x, v)
```

```
# In[9]:


# Exercise 1.5
# Solution
t, x, v = eulerImp()
plt.plot(t, x, label='x')
plt.plot(t, v, label='v')
plt.legend()
plt.title('Implicit Euler: SHM')
plt.xlabel('Time')
plt.savefig('img/p1_5_1.pdf')


# In[10]:


# Exercise 1.5
# Global error
t, x, v = eulerImp()
T, X, V = SHM()
plt.plot(t, X-x, label='$x_{\\text{true}}-x$')
plt.plot(t, V-v, label='$v_{\\text{true}}-v$')
plt.legend()
plt.title('Implicit Euler: Global error')
plt.xlabel('Time')
plt.savefig('img/p1_5_2.pdf')


# In[11]:


# Exercise 1.5
# Truncation error
h0 = 0.1
h = h0/np.power(2, np.arange(5))
err = np.zeros(np.size(h))
for i in range(len(h)):
    err[i] = np.max(np.abs(SHM(h=h[i])[1]-eulerImp(h=h[i])[1]))
plt.plot(h, err)
plt.title('Implicit Euler: Truncation error')
plt.xlabel('Step size (h)')
plt.ylabel('Error')
plt.savefig('img/p1_5_3.pdf')


# In[12]:


# Exercise 1.5
```

```
# Energy
t, x, v = eulerImp()
E = np.power(x, 2) + np.power(v, 2)
plt.plot(t, E)
plt.title('Implicit Euler: Energy')
plt.xlabel('Time')
plt.ylabel('Energy')
plt.savefig('img/p1_5_4.pdf')


# In[13]:


# Exercise 2.1
plt.figure(figsize=(5,5))
t, x, v = eulerImp()
T, X, V = eulerExp()
plt.plot(x, v, label='Implicit')
plt.plot(X, V, label='Explicit')
plt.legend()
plt.title('Phase space: Euler methods')
plt.xlabel('Position')
plt.ylabel('Velocity')
plt.axis('equal')
plt.savefig('img/p2_1.pdf')


# In[14]:


# Exercise 2.2
def eulerSym(x0=1., v0=0., t0=0., P=5, h=0.1):
    t = np.arange(t0, t0+P*2*np.pi+h, h)
    N = len(t)
    x = np.zeros(N)
    v = np.zeros(N)
    x[0], v[0] = x0, v0
    for i in range(N-1):
        x[i+1] = x[i] + h*v[i]
        v[i+1] = v[i] - h*x[i+1]
    return (t, x, v)


# In[15]:


# Exercise 2.2
plt.figure(figsize=(5,5))
t, x, v = eulerImp()
T, X, V = eulerExp()
ts, xs, vs = eulerSym()
```

```
ta, xa, va = SHM()
plt.plot(x, v, label='Implicit')
plt.plot(X, V, label='Explicit')
plt.plot(xs, vs, label='Symplectic')
plt.plot(xa, va, label='Analytic')
plt.scatter(1, 0, marker='.', zorder=4, color='r', s=150)
plt.legend()
plt.title('Phase space: Euler methods')
plt.xlabel('Position')
plt.ylabel('Velocity')
plt.axis('equal')
plt.savefig('img/p2_2.pdf')


# In[16]:


# Exercise 2.3
# Energy
t, x, v = eulerSym()
E = np.power(x, 2) + np.power(v, 2)
plt.plot(t, E)
plt.title('Symplectic Euler: Energy')
plt.xlabel('Time')
plt.ylabel('Energy')
plt.savefig('img/p2_3.pdf')


# In[17]:
```

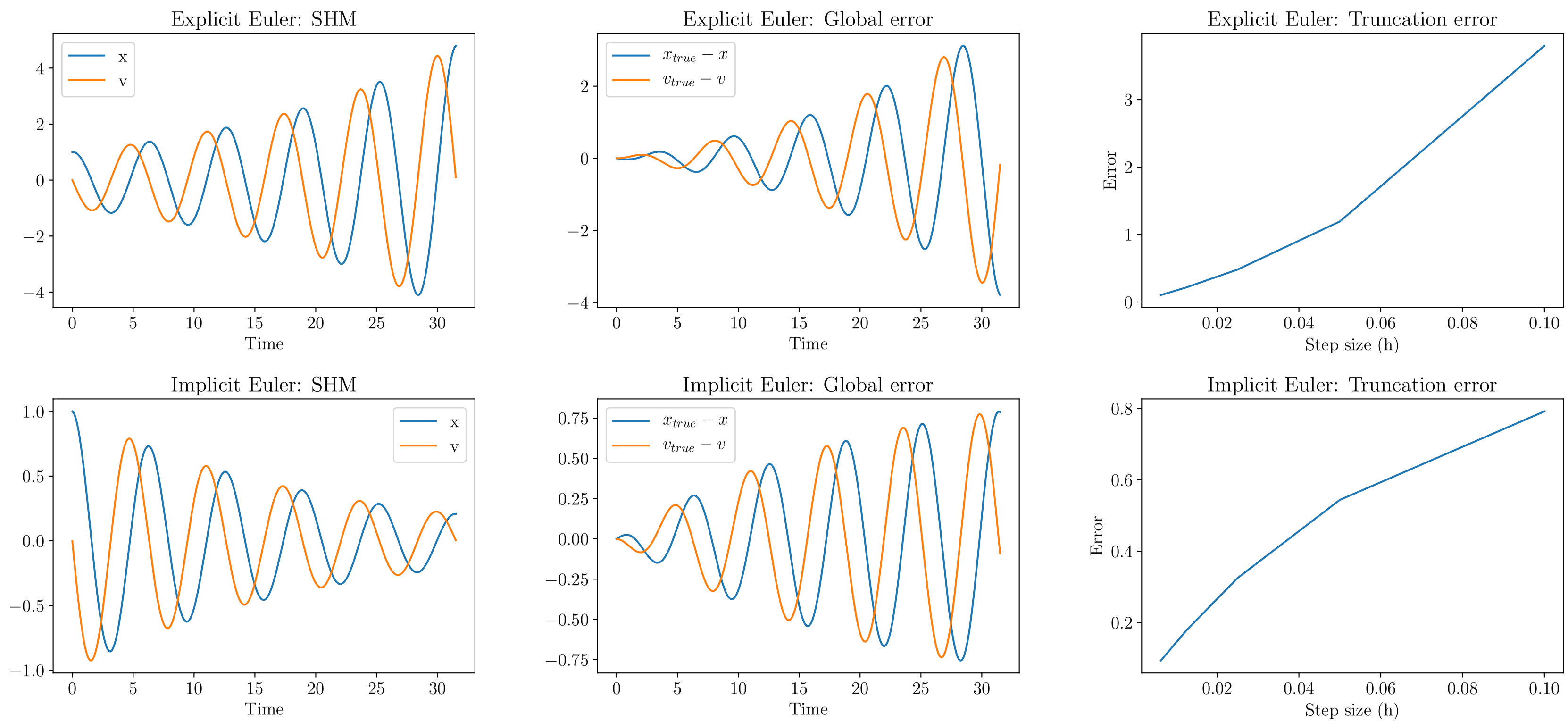

Figure 1: Euler method approximations of simple harmonic motion with $x_0 = 1$ and $v_0 = 0$ using a step size of $h = 0.1$ unless otherwise specified. The explicit Euler method increasingly overestimates velocity and position over time, while the implicit Euler method underestimates. The global error of the explicit method grows without bound, while the global error of the implicit method approaches the analytic solution as the approximate motion damps to stationary.

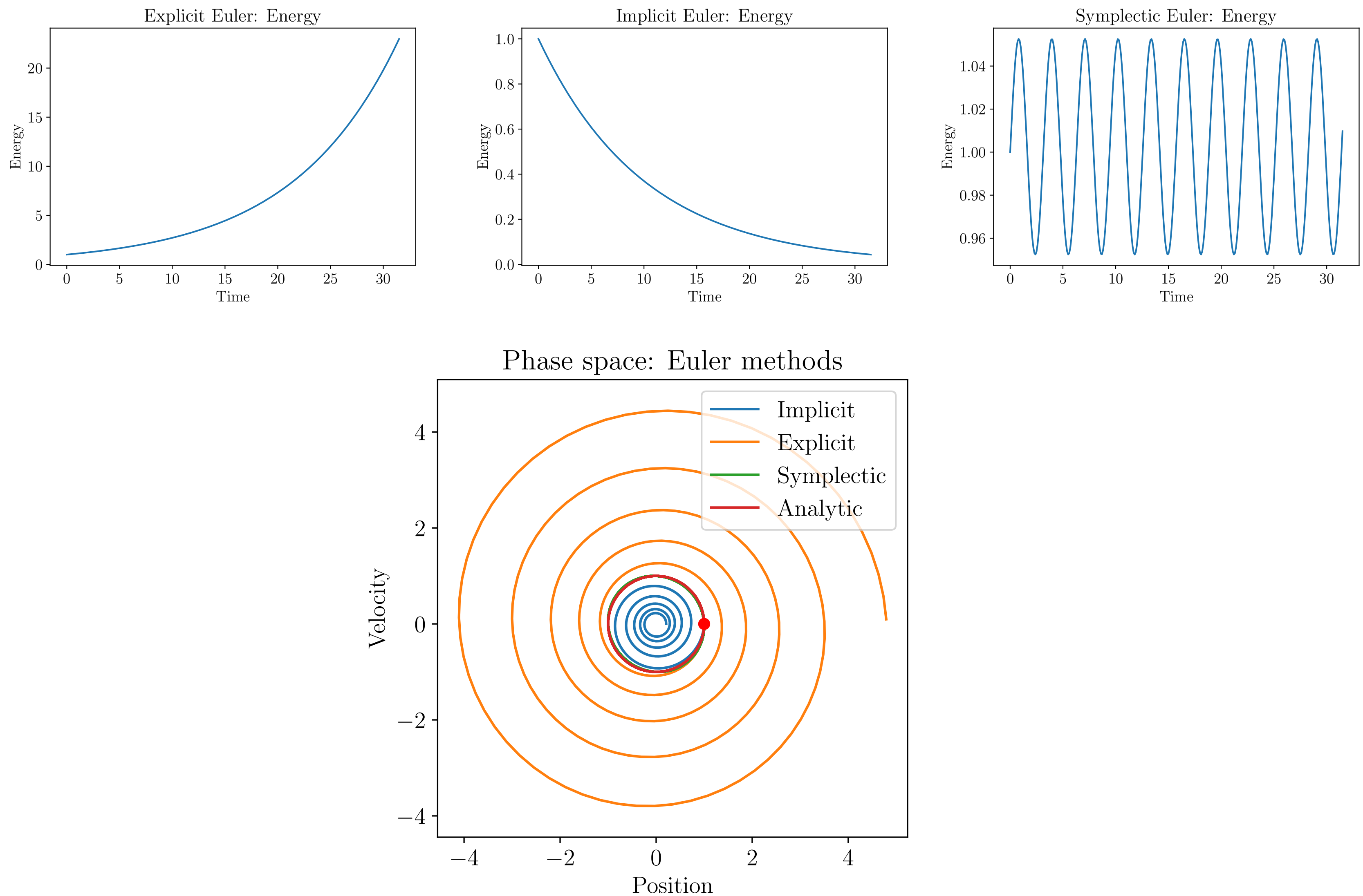

Figure 2: Euler method approximations of simple harmonic motion with $x_0 = 1$ and $v_0 = 0$ using a step size of $h = 0.1$. Plotted over 5 periods of oscillation. The explicit method accumulates energy and the implicit method dissipates energy; this is obvious in the phase-space diagram as well. The symplectic method does not conserve energy strictly, but the time-averaged energy is conserved. Unlike the other methods, the symplectic Euler phase-space trajectory is closed.