kpp_vs_kppa

Unknown Author

April 10, 2015

In [14]:
```
kpp_file_1 = '0-dim_box_model_Exa2Green/kpp-2.2.1.dat'
kpp_file_2 = '0-dim_box_model_Exa2Green/box_model.dat_OpenMP'
```

In [15]:
```
%matplotlib inline
import re
from itertools import cycle
from pylab import *
from matplotlib.markers import MarkerStyle
import matplotlib.pyplot as plt

ATOL = 1.0e-2
RTOL = 1.0e-2
EPS = 2.2204460492503131E-016
REGEX = re.compile('^([+\-]?)([0-9.]+)e?([+\-])([0-9.]+)$')
def convert(s):
    """
    Converts a number in Fortran E24.16 format to a Python float
    """
    m = re.search(REGEX, s)
    if m:
        s = ''.join([m.group(1), m.group(2), 'e', m.group(3), m.group(4)])
    try:
        fval = float(s)
    except ValueError:
        print '================> %s' % s
        fval = 0.0
    if fval < EPS:
        return 0.0
    else:
        return fval

def read_datfile(fname, tstart, cstart):
    """
    Read data from fname beginning on line tstart with concentration data beginning in
    Returns a tuple: (time, concentrations)
    Time data:
    [t0 t1 ... tN]
    Concentration data:
    [ [SPC_0(t0) SPC_1(t0) ... SPC_N(t0)]
    [SPC_0(t1) SPC_1(t1) ... SPC_N(t1)]
    : : :
    [SPC_0(tN) SPC_1(tN) ... SPC_N(tN)] ]
    """
    t = []
    c = []
    with open(fname, 'r') as f:
        while tstart:
            f.readline()
            tstart -= 1
        for line in f:
            parts = line.split()
            t.append(convert(parts[0]))
            c.append([convert(x) for x in parts[cstart:]])
    return t, c

def plot_dat(data, xlabel='Time', ylabel='Conc', names=None, titles=None):
    """
    Draw a plot of data read from read_datfile
    """
    lines = ['-', '--', '-.', ':']
    markers = MarkerStyle.filled_markers
    linecycler = cycle(lines)
    markercycler = cycle(markers)
    datastyles = ['%s%s' % (linecycler.next(), markercycler.next()) for _ in data]
    ndat = len(data)
    nspec = len(data[0][1][0])
    x = data[0][0]
    for i in xrange(0, nspec):
        fig, ax = plt.subplots()
        for j, dat in enumerate(data):
            t, c = dat
            y = [ct[i] for ct in c]
            style = datastyles[j]
            if names:
                label = '%s' % names[j]
            else:
                label = '%d' % j
            ax.plot(x, y, style, label=label)
        if ndat > 1:
            ax.legend(loc=2)
        ax.set_xlabel(xlabel)
        ax.set_ylabel(ylabel)
        if titles:
            ax.set_title(titles[i])
        else:
            ax.set_title('Species %d' % i)
        show()

def scaled_err(x, y):
    if x or y:
        return abs(x-y)/max(x, y)
    elif x == y:
        return 0.0
    else:
        return float('inf')

def calc_err(d0, d1):
    c0 = d0[1]
    c1 = d1[1]
    err = []
    nsteps = len(c0)
    nspec = len(c0[0])
    sigPow = 0.0
    errPow = 0.0
    errCount = 0.0
    for i in xrange(0, nsteps):
        e = []
        for j in xrange(0, nspec):
            x = c0[i][j]
            y = c1[i][j]
            sigPow += x*x
            errPow += (x-y)*(x-y)
            serr = scaled_err(x,y)
            if serr > RTOL:
                print '%g > %g: %g, %g' % (serr, RTOL, x, y)
                errCount += 1
            e.append(serr)
        err.append(e)
    if errPow > 0:
        snr = 20 * log10(sigPow / errPow)
    else:
        snr = float('inf')
    print 'SNR: %fdb' % snr
    if errCount:
        print '%d samples with relative error > %g' % (errCount, RTOL)
    return d1[0], err
```

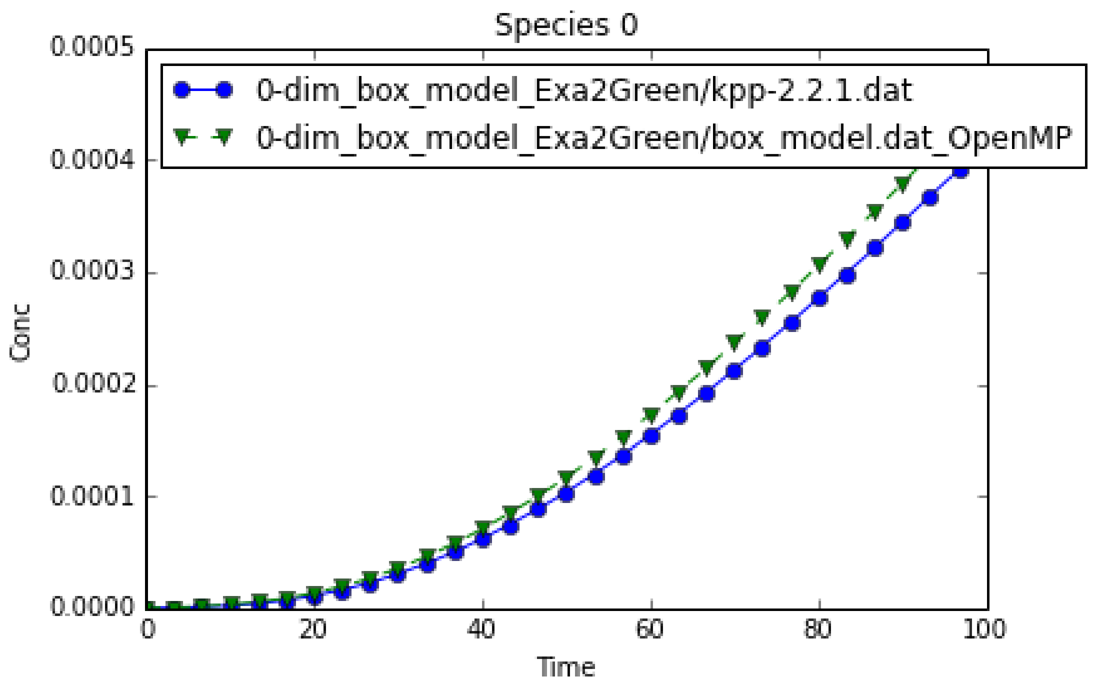
In [16]:
```
kpp_dat_1 = read_datfile(kpp_file_1, 0, 1)
kpp_dat_2 = read_datfile(kpp_file_2, 0, 1)
```

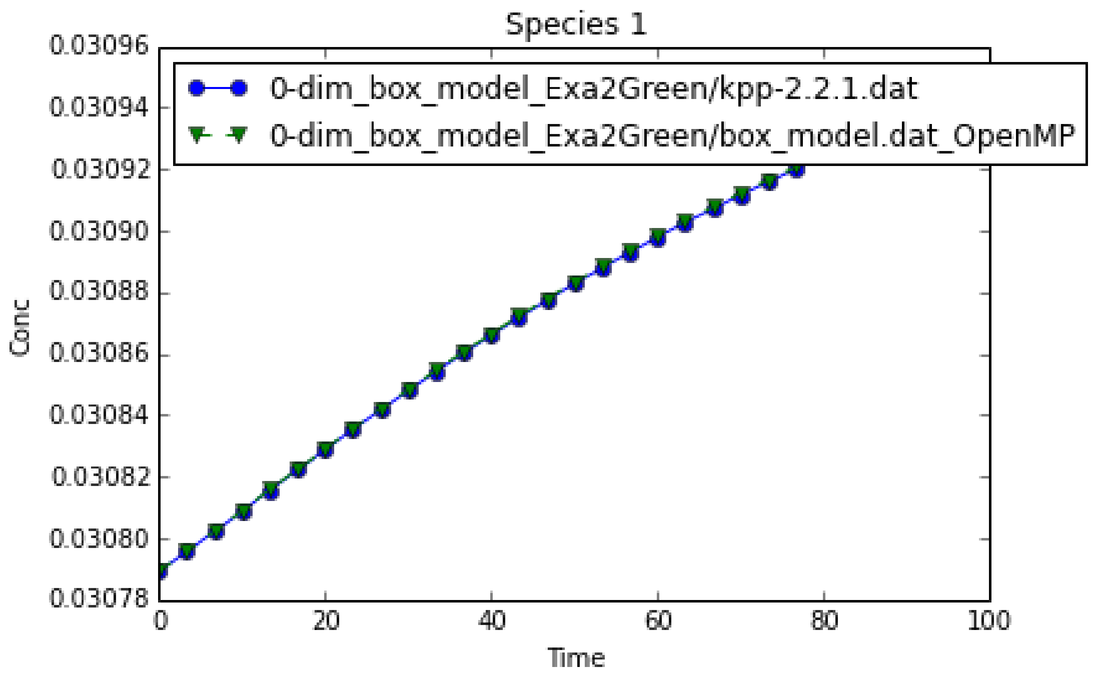
In [17]:
```
err_dat = calc_err(kpp_dat_1, kpp_dat_2)
```

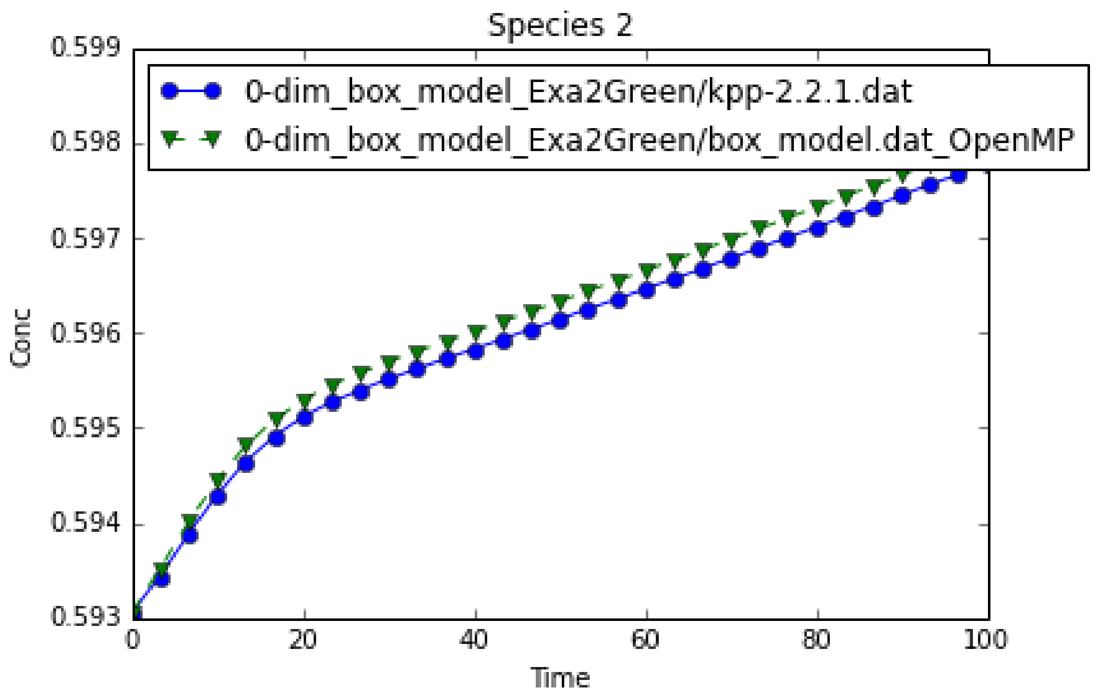
```
0.361836 > 0.01: 3.47092e-07, 5.43892e-07
0.0499489 > 0.01: 2.58437e-06, 2.45528e-06
0.314789 > 0.01: 1.03217e-06, 1.50635e-06
0.0531104 > 0.01: 1.38604e-06, 1.31242e-06
0.272678 > 0.01: 2.191e-06, 3.01242e-06
0.0506626 > 0.01: 8.57078e-07, 8.13656e-07
0.233603 > 0.01: 4.08409e-06, 5.32894e-06
0.0517952 > 0.01: 6.24647e-07, 5.92293e-07
0.202608 > 0.01: 6.93453e-06, 8.69651e-06
0.0523877 > 0.01: 5.20619e-07, 4.93345e-07
0.180151 > 0.01: 1.09141e-05, 1.33123e-05
0.0522227 > 0.01: 4.70318e-07, 4.45757e-07
0.163511 > 0.01: 1.6137e-05, 1.92914e-05
0.0528041 > 0.01: 4.42854e-07, 4.1947e-07
0.151125 > 0.01: 2.26559e-05, 2.66893e-05
0.0529358 > 0.01: 4.24034e-07, 4.01588e-07
0.14169 > 0.01: 3.04836e-05, 3.55159e-05
0.0530632 > 0.01: 4.08384e-07, 3.86714e-07
0.134312 > 0.01: 3.9607e-05, 4.57521e-05
0.0531989 > 0.01: 3.93746e-07, 3.72799e-07
0.12839 > 0.01: 4.99971e-05, 5.73618e-05
0.0533419 > 0.01: 3.7939e-07, 3.59152e-07
0.123516 > 0.01: 6.16146e-05, 7.02975e-05
0.0534896 > 0.01: 3.65142e-07, 3.45611e-07
0.119416 > 0.01: 7.44131e-05, 8.45043e-05
0.0536402 > 0.01: 3.51028e-07, 3.32199e-07
0.115898 > 0.01: 8.83411e-05, 9.99218e-05
0.0537925 > 0.01: 3.37132e-07, 3.18997e-07
0.112827 > 0.01: 0.000103342, 0.000116485
0.0539458 > 0.01: 3.2355e-07, 3.06095e-07
0.110106 > 0.01: 0.000119358, 0.000134126
0.0540999 > 0.01: 3.10367e-07, 2.93576e-07
0.107666 > 0.01: 0.000136326, 0.000152774
0.0542544 > 0.01: 2.97657e-07, 2.81508e-07
0.105452 > 0.01: 0.000154182, 0.000172358
0.0544092 > 0.01: 2.85477e-07, 2.69945e-07
0.103427 > 0.01: 0.000172862, 0.000192803
0.054564 > 0.01: 2.7387e-07, 2.58927e-07
0.101559 > 0.01: 0.0001923, 0.000214037
0.0547186 > 0.01: 2.62864e-07, 2.4848e-07
0.0998251 > 0.01: 0.00021243, 0.000235987
0.0548728 > 0.01: 2.52472e-07, 2.38618e-07
```

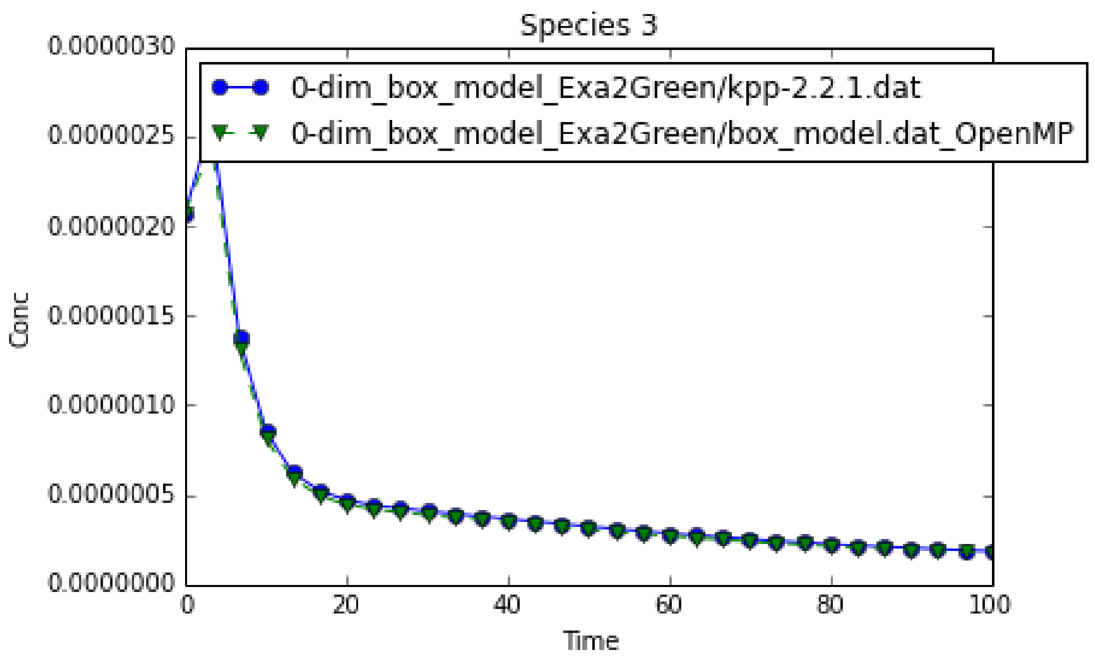
```
0.0982071 > 0.01: 0.000233186, 0.000258581
0.0550263 > 0.01: 2.42698e-07, 2.29343e-07
0.0966904 > 0.01: 0.000254505, 0.000281747
0.0551789 > 0.01: 2.33534e-07, 2.20648e-07
0.0952632 > 0.01: 0.000276322, 0.000305417
0.0553305 > 0.01: 2.24964e-07, 2.12517e-07
0.0939161 > 0.01: 0.000298577, 0.000329525
0.0554807 > 0.01: 2.16965e-07, 2.04927e-07
0.0926412 > 0.01: 0.000321209, 0.000354004
0.0556293 > 0.01: 2.09507e-07, 1.97852e-07
0.0914321 > 0.01: 0.00034416, 0.000378794
0.0557762 > 0.01: 2.02558e-07, 1.9126e-07
0.0902833 > 0.01: 0.000367375, 0.000403835
0.0559211 > 0.01: 1.96083e-07, 1.85118e-07
0.0891902 > 0.01: 0.000390801, 0.00042907
0.0560639 > 0.01: 1.90043e-07, 1.79389e-07
0.0881488 > 0.01: 0.000414386, 0.000454444
0.0562045 > 0.01: 1.84402e-07, 1.74038e-07
SNR: 195.332820db
60 samples with relative error > 0.01
```

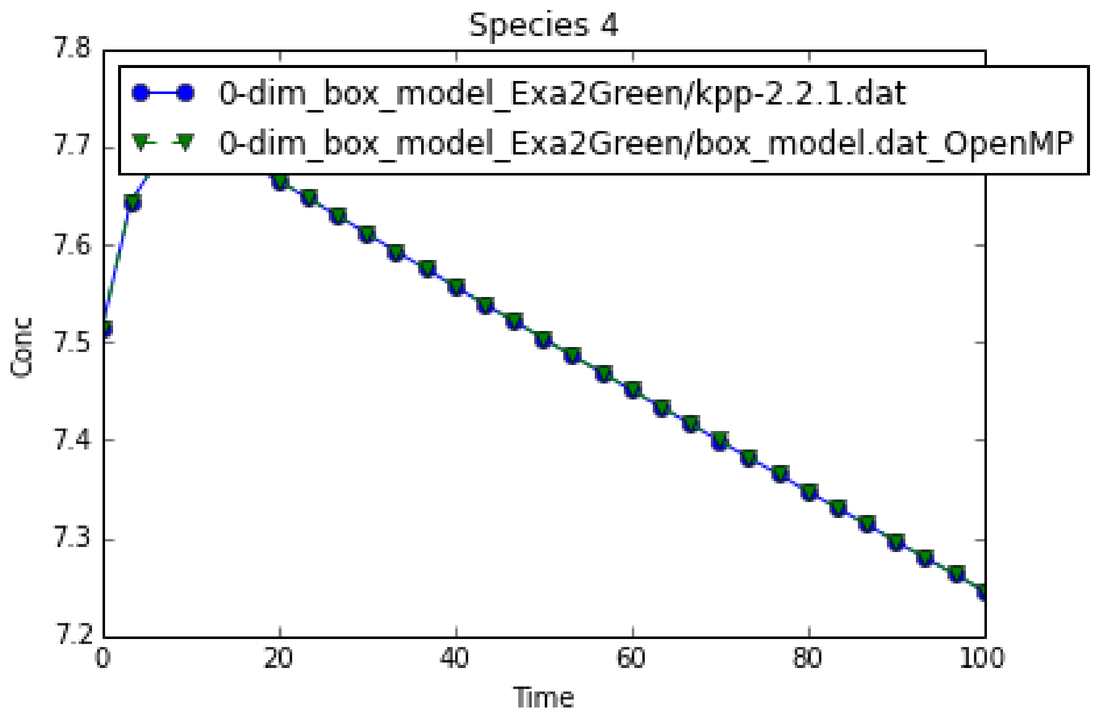
In [18]:
```
plot_dat([kpp_dat_1, kpp_dat_2], names=[kpp_file_1, kpp_file_2], titles=None)
```

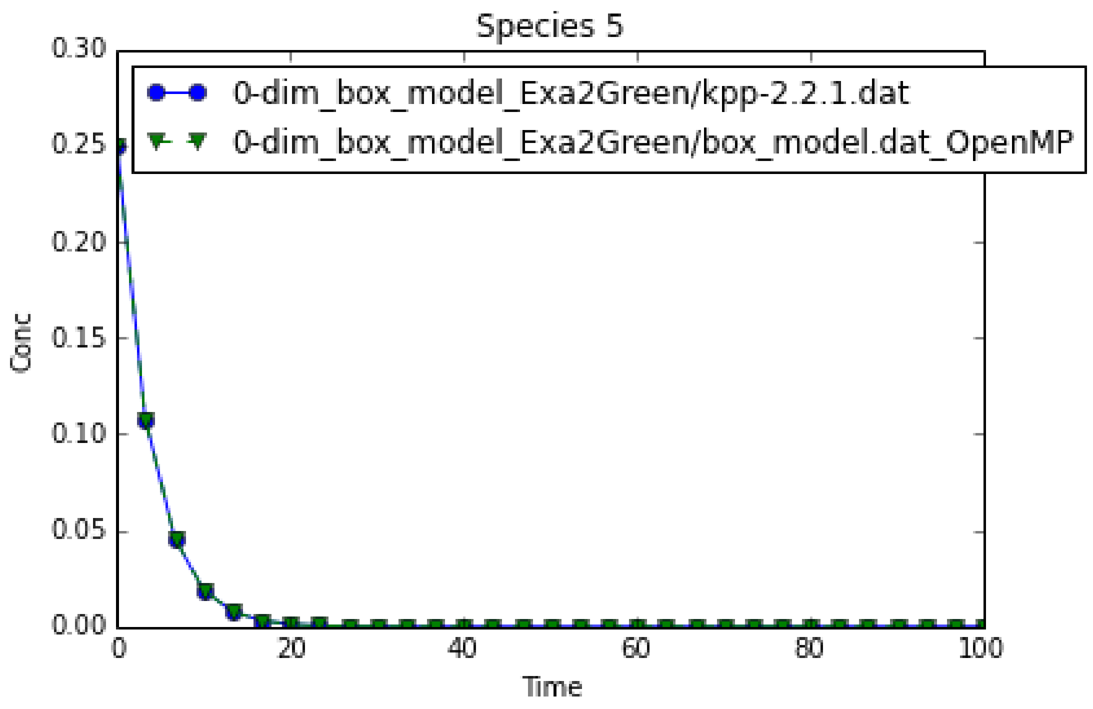


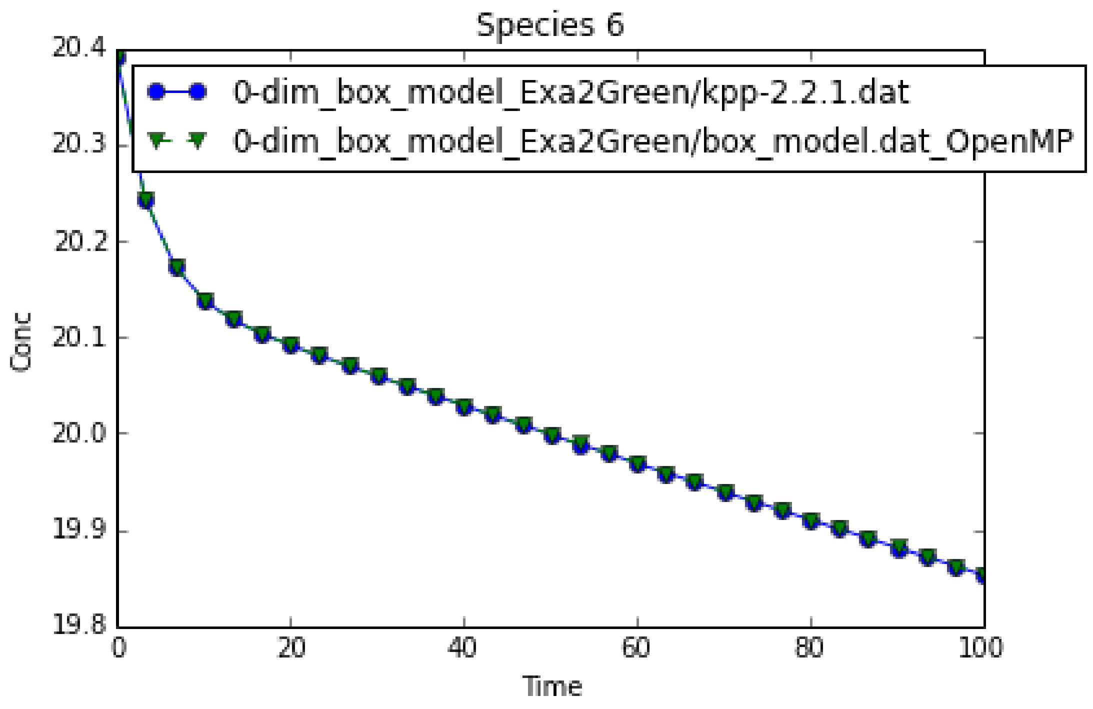

Species 1
0-dim_box_model_Exa2Green/kpp-2.2.1.dat
0-dim_box_model_Exa2Green/box_model.dat_OpenMP
Conc
Time
0.03096
0.03094
0.03092
0.03090
0.03088
0.03086
0.03084
0.03082
0.03080
0.03078
0
20
40
60
80
100

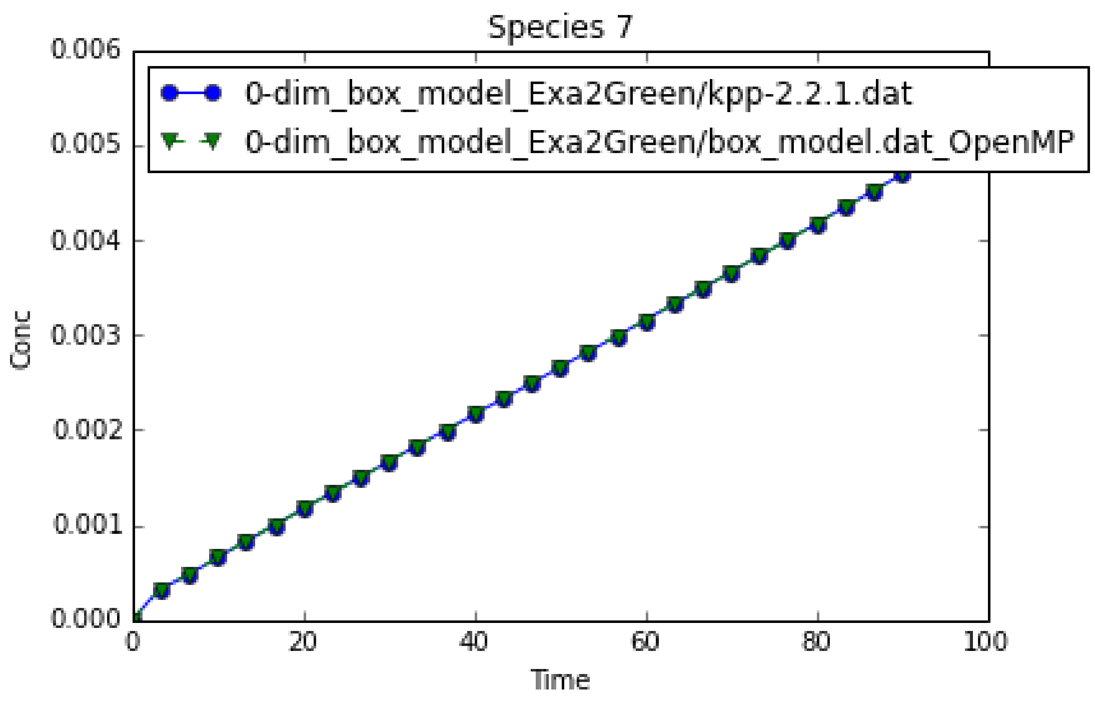

Species 2
0-dim_box_model_Exa2Green/kpp-2.2.1.dat
0-dim_box_model_Exa2Green/box_model.dat_OpenMP
Conc
Time
0.599
0.598
0.597
0.596
0.595
0.594
0.593
0
20
40
60
80
100


Species 3
0.0000030
0.0000025
0.0000020
0.0000015
0.0000010
0.0000005
0.0000000
Conc
0
20
40
60
80
100
Time
0-dim_box_model_Exa2Green/kpp-2.2.1.dat
0-dim_box_model_Exa2Green/box_model.dat_OpenMP


Species 4
7.8
7.7
7.6
7.5
7.4
7.3
7.2
Conc
0
20
40
60
80
100
Time
0-dim_box_model_Exa2Green/kpp-2.2.1.dat
0-dim_box_model_Exa2Green/box_model.dat_OpenMP

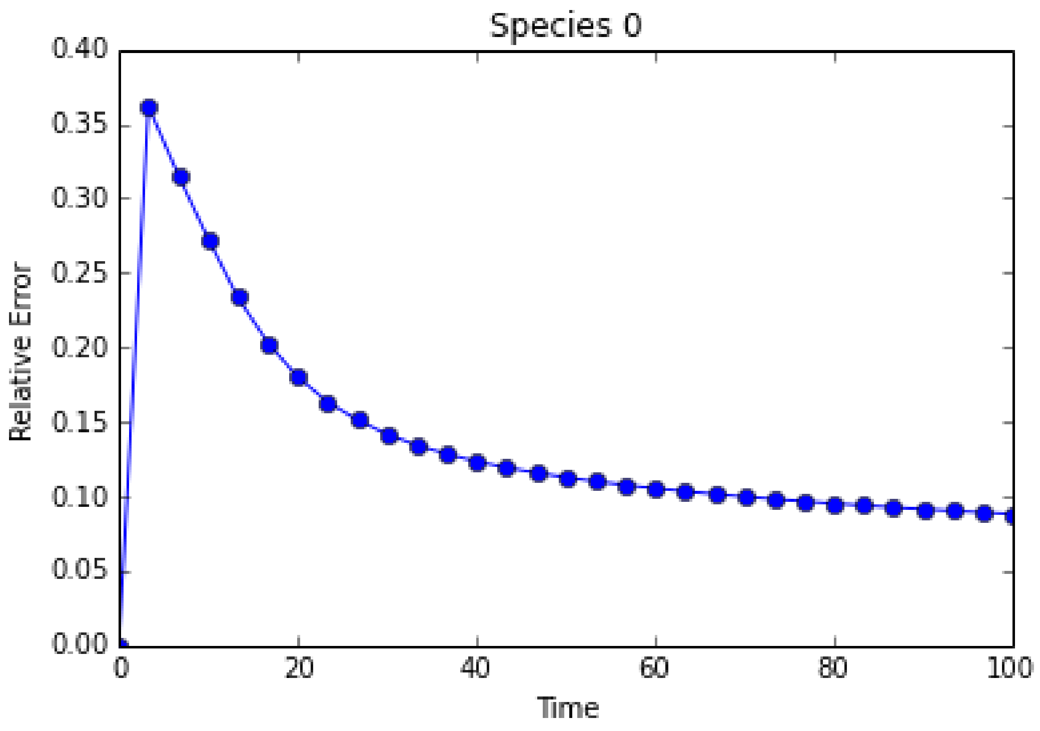

Species 5
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Conc
0
20
40
60
80
100
Time
0-dim_box_model_Exa2Green/kpp-2.2.1.dat
0-dim_box_model_Exa2Green/box_model.dat_OpenMP

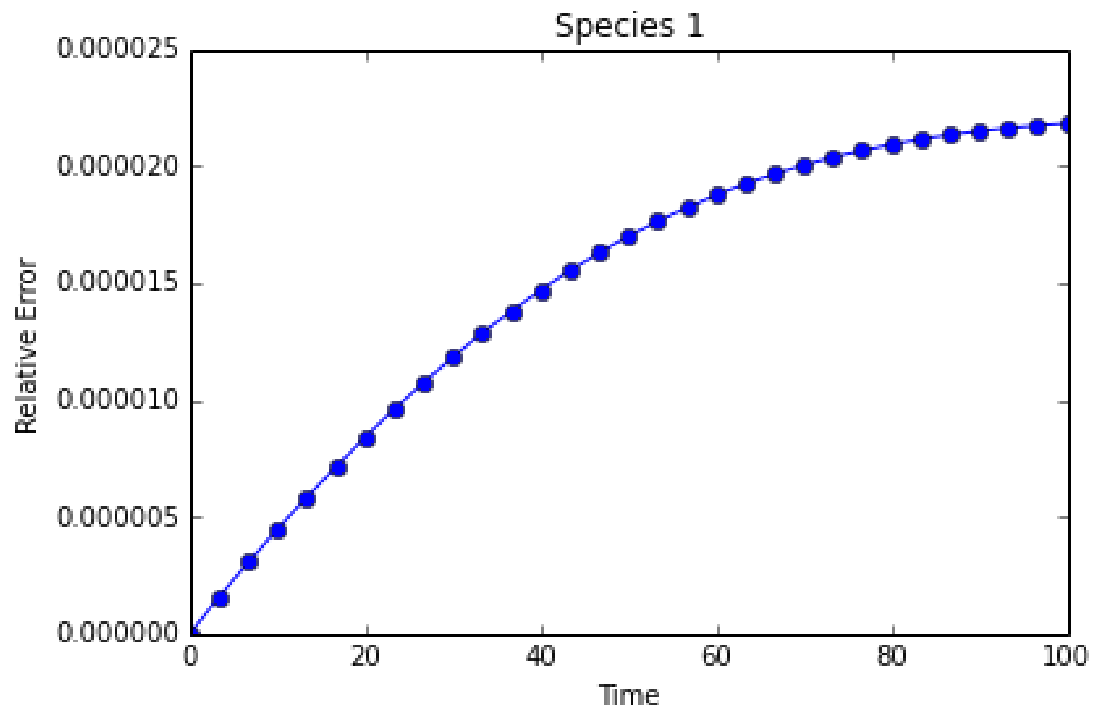

Species 6
20.4
20.3
20.2
20.1
20.0
19.9
19.8
Conc
0
20
40
60
80
100
Time
0-dim_box_model_Exa2Green/kpp-2.2.1.dat
0-dim_box_model_Exa2Green/box_model.dat_OpenMP

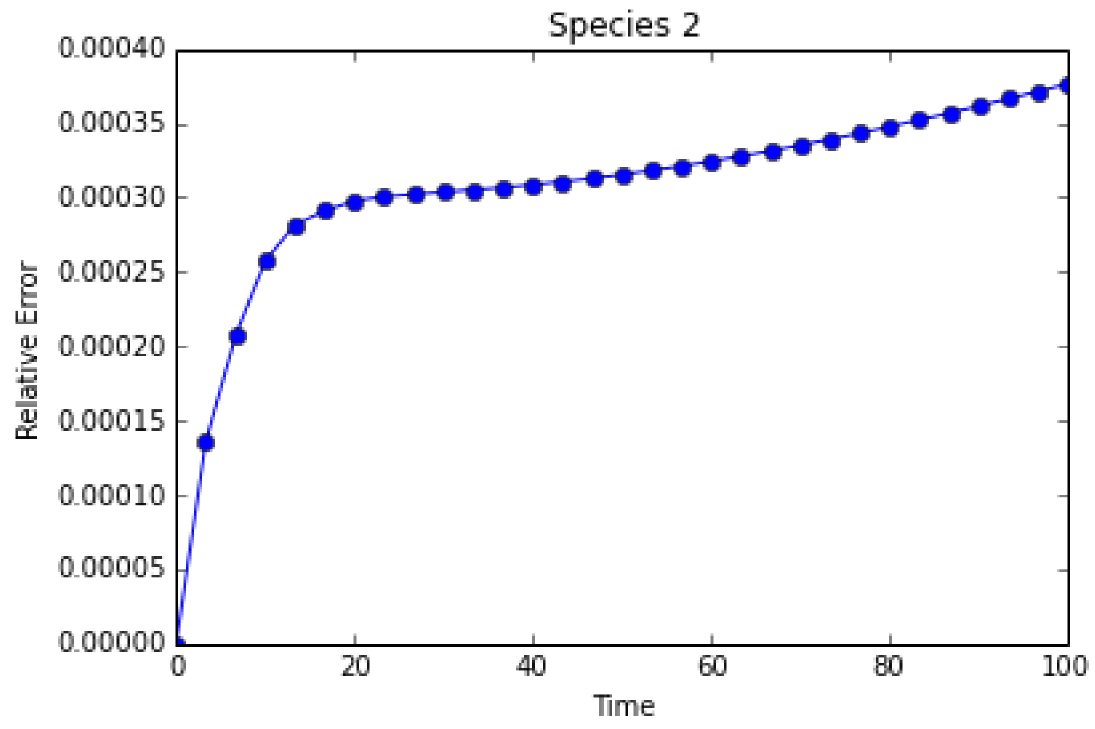


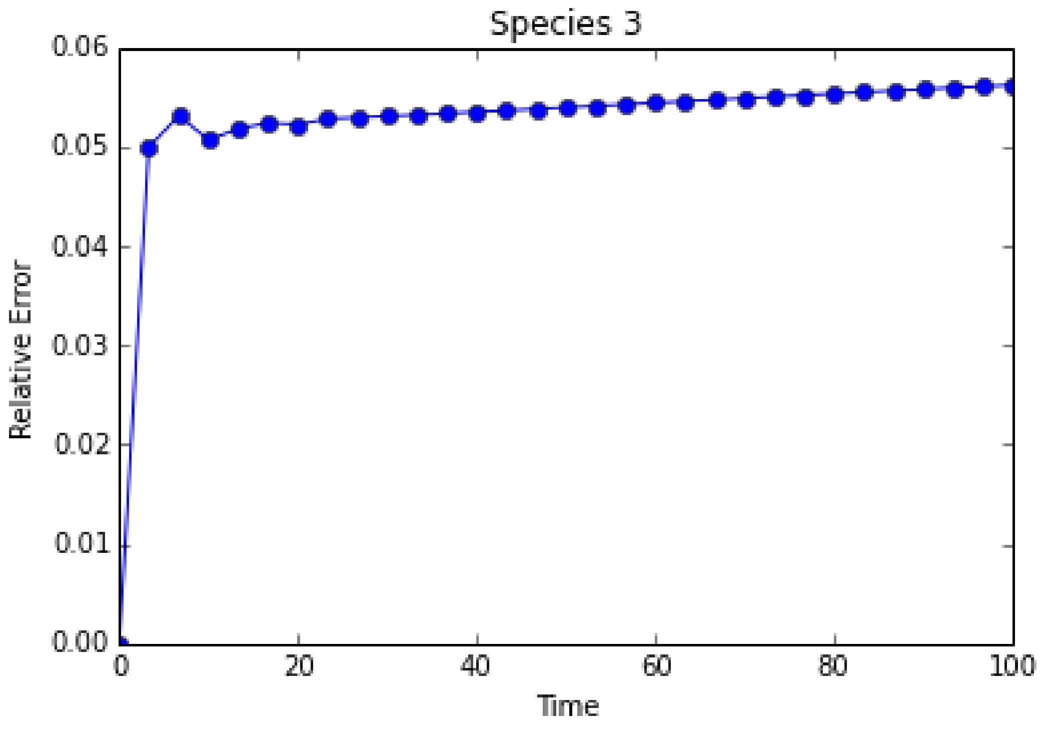
In [19]:

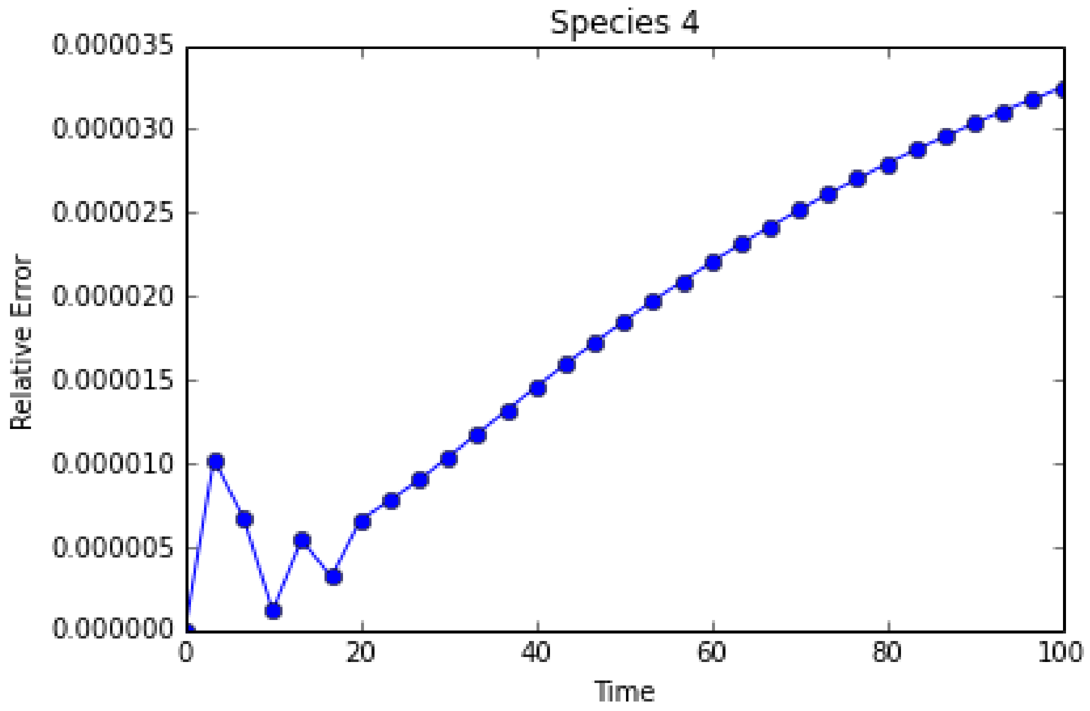
```
plot_dat([err_dat], ylabel='Relative Error')
```

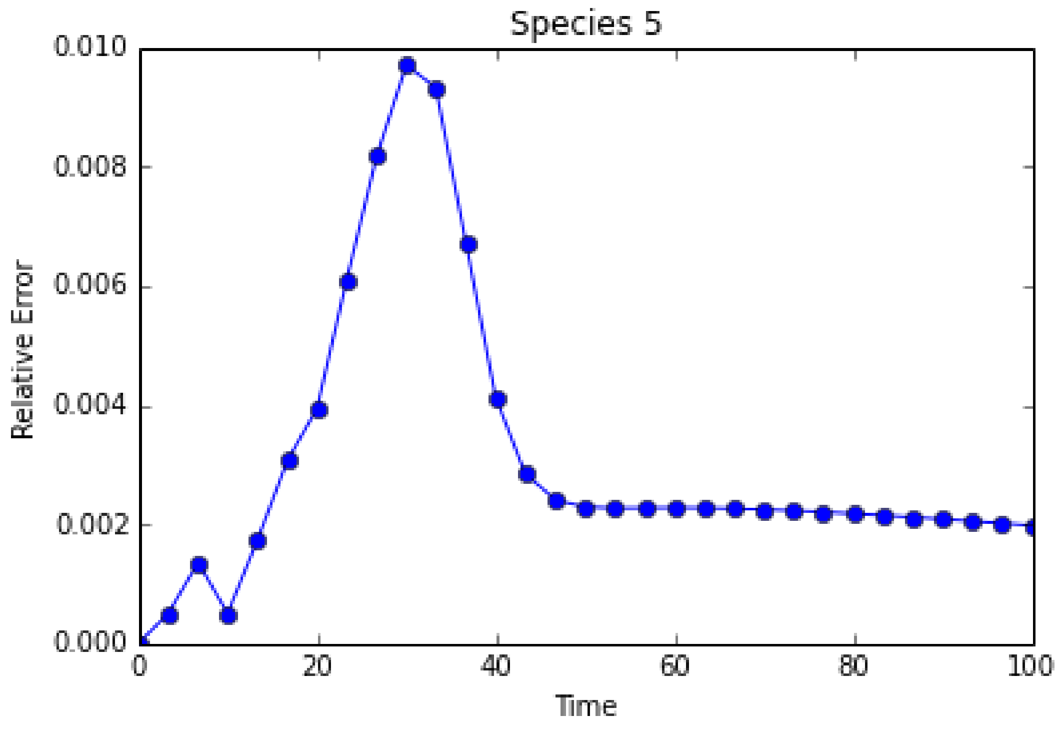


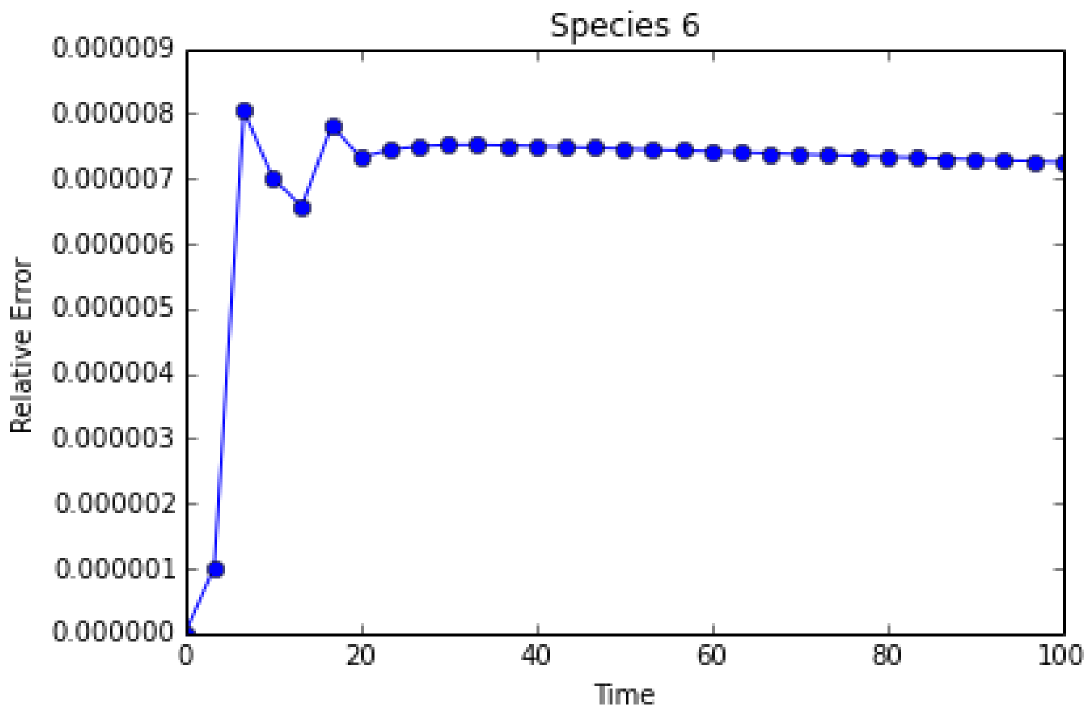

Species 1
0.000025
0.000020
0.000015
0.000010
0.000005
0.000000
Relative Error
0
20
40
60
80
100
Time

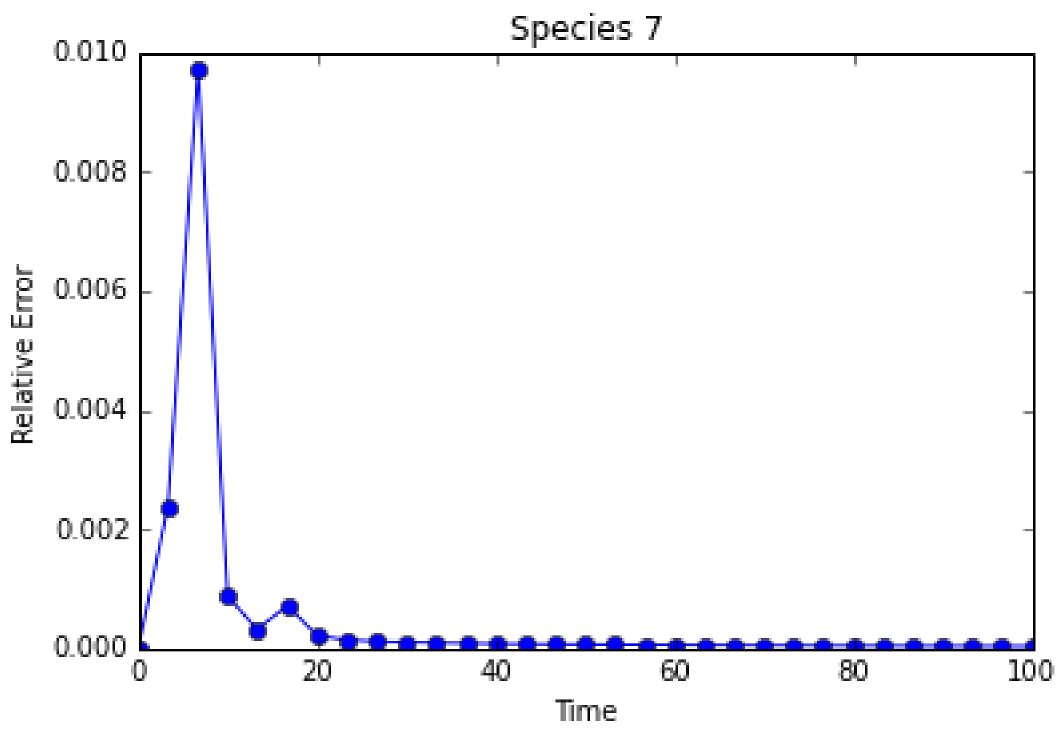

Species 2
0.00040
0.00035
0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
Relative Error
0
20
40
60
80
100
Time


Species 3
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Relative Error
0
20
40
60
80
100
Time


Species 4
0.000035
0.000030
0.000025
0.000020
0.000015
0.000010
0.000005
0.000000
Relative Error
0
20
40
60
80
100
Time


Species 5
Relative Error
Time
0.000
0.002
0.004
0.006
0.008
0.010
0
20
40
60
80
100


Species 6
Relative Error
Time
0.000000
0.000001
0.000002
0.000003
0.000004
0.000005
0.000006
0.000007
0.000008
0.000009
0
20
40
60
80
100


Species 7
Relative Error
Time
0.010
0.008
0.006
0.004
0.002
0.000
0
20
40
60
80
100

```
In [19]:
```