kpp_vs_kppa

Unknown Author

December 2, 2014

In [7]:
```
kpp_file_1 = '0-dim_box_model_Exa2Green/box_model.dat_CUDA'
kpp_file_2 = '0-dim_box_model_Exa2Green/box_model.dat_OpenMP'
```

In [8]:
```
%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)
```

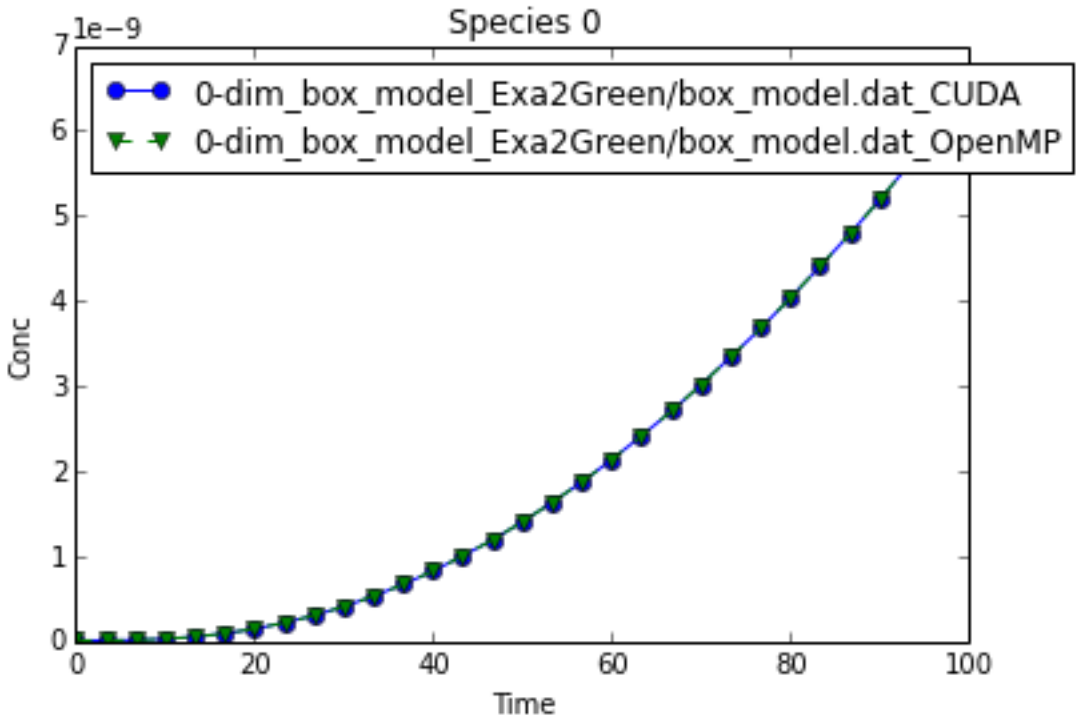
```
            err.append(e)
        if errPow > 0:
            snr = 20 * log10(sigPow / errPow)
        else:
            snr = float('inf')
        print 'SNR: %fdb' % snr
        if errCount:
            print '%d samples with relative error > %g' % (errCount, RTOL)
        return d1[0], err
```

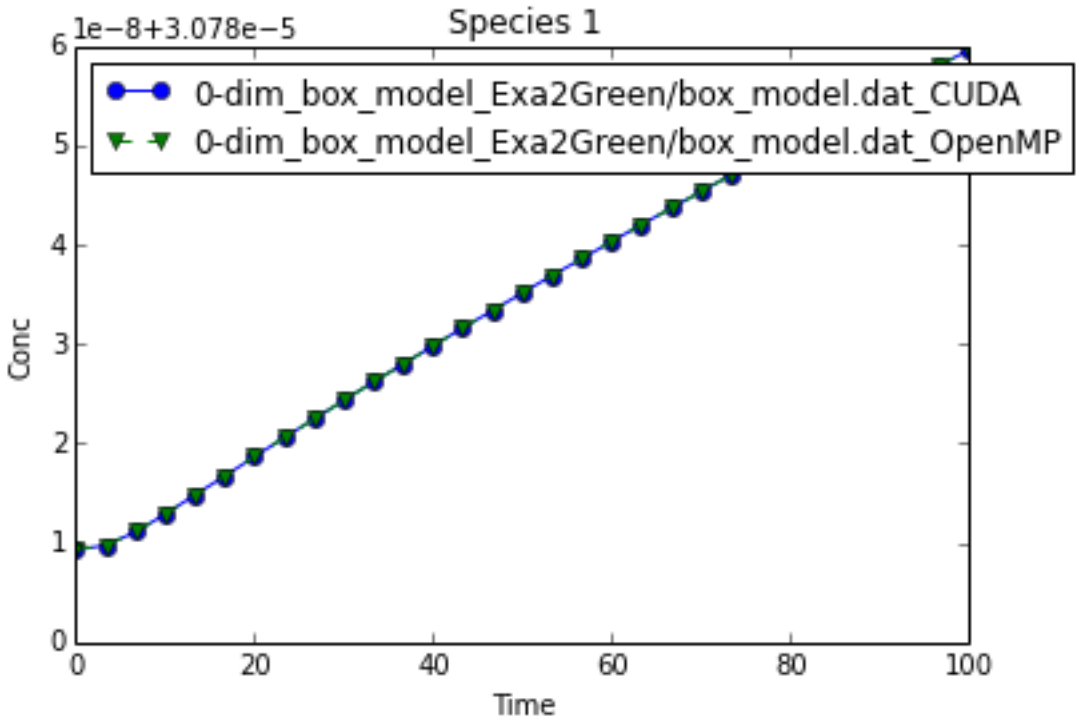
In [9]:
```
kpp_dat_1 = read_datfile(kpp_file_1, 0, 1)
kpp_dat_2 = read_datfile(kpp_file_2, 0, 1)
```

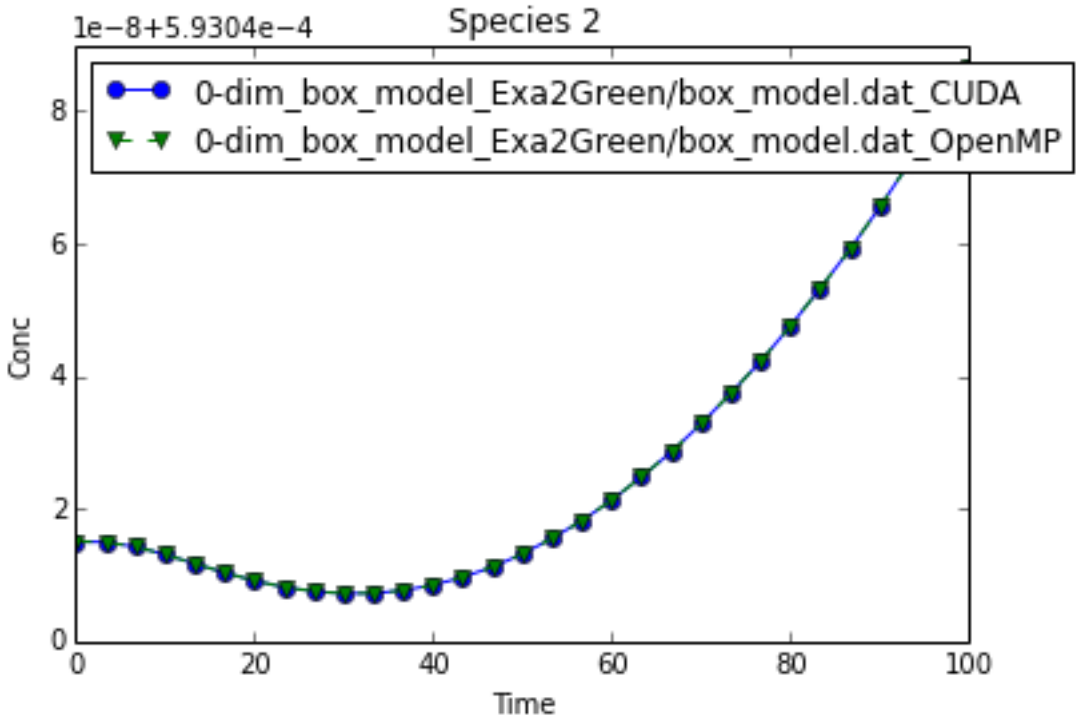
In [10]:
```
err_dat = calc_err(kpp_dat_1, kpp_dat_2)
```

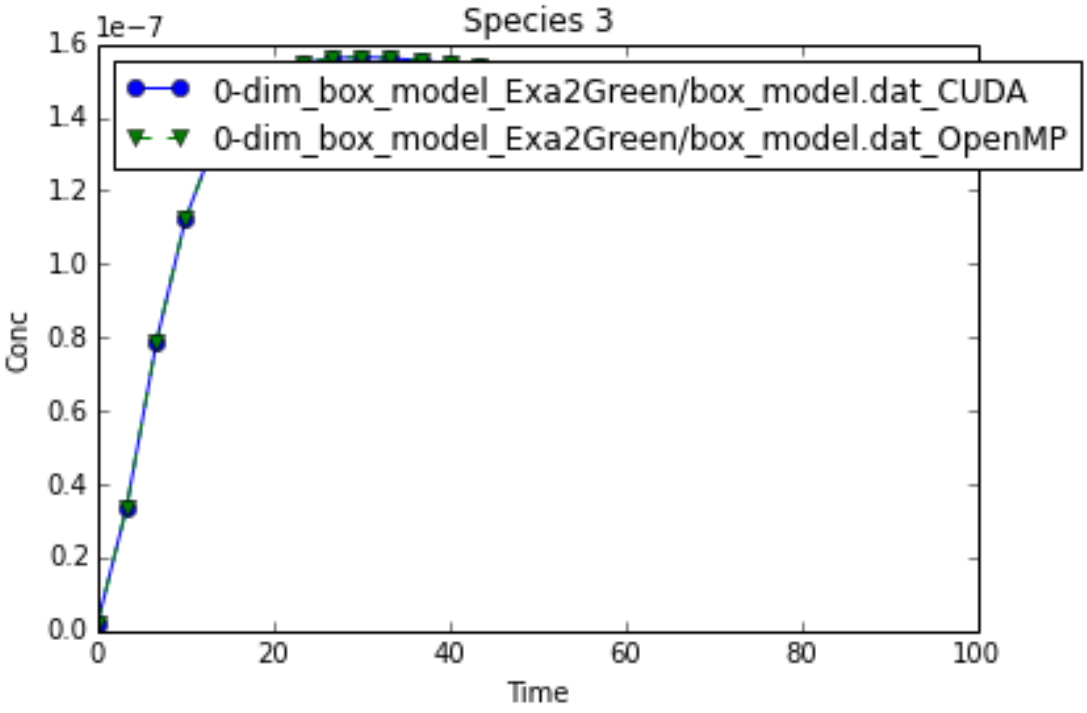
```
SNR: 604.815762db
```

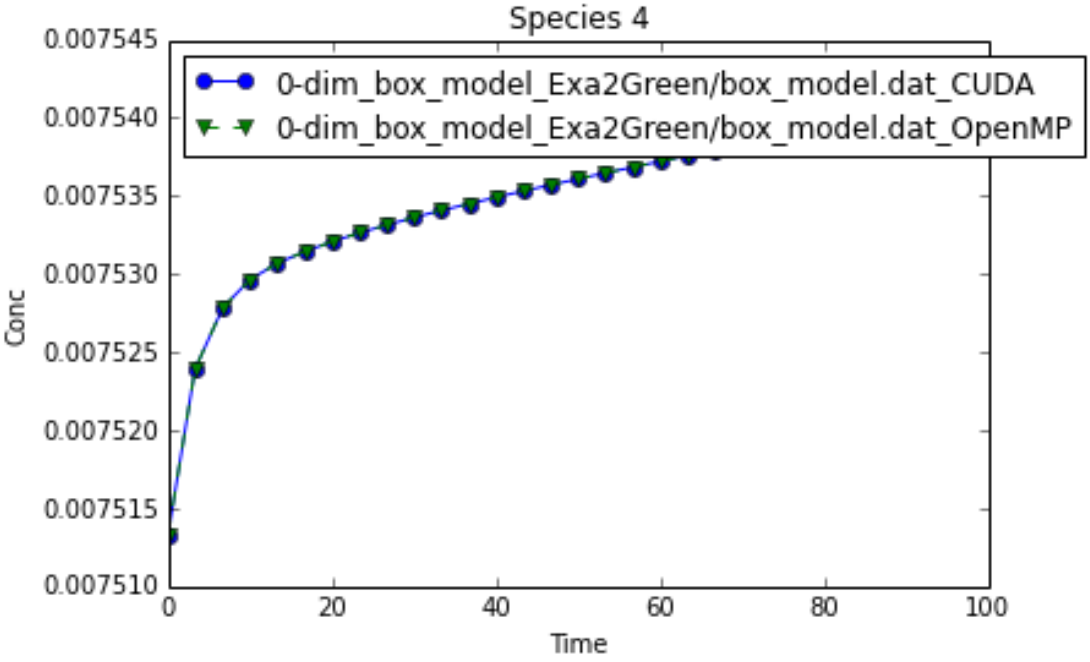
In [11]:
```
plot_dat([kpp_dat_1, kpp_dat_2], names=[kpp_file_1, kpp_file_2], titles=None)
```

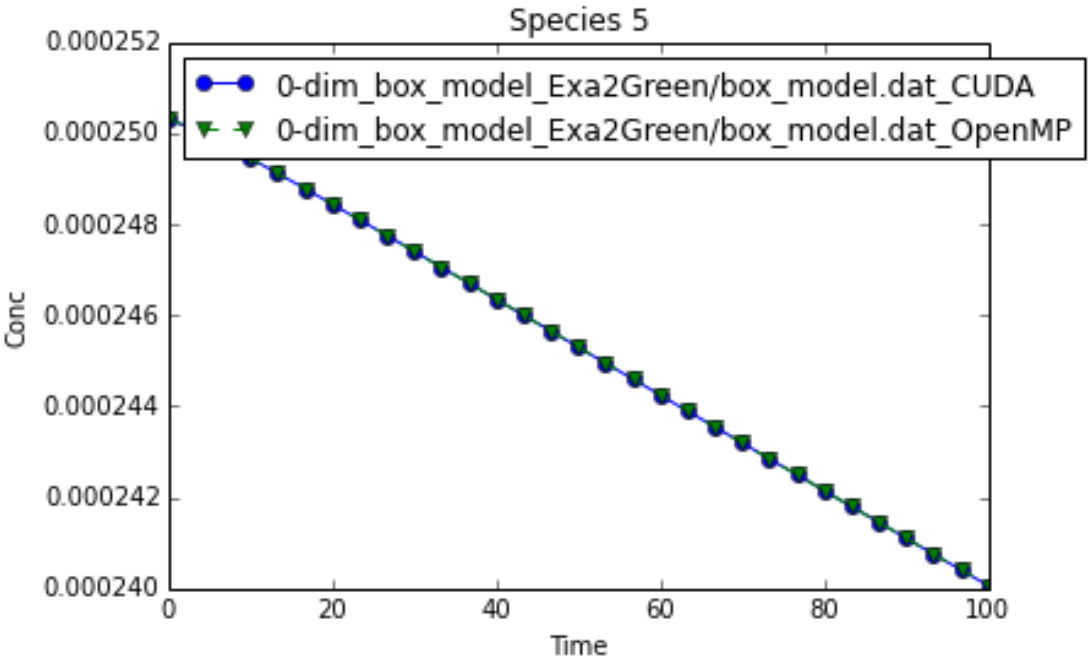


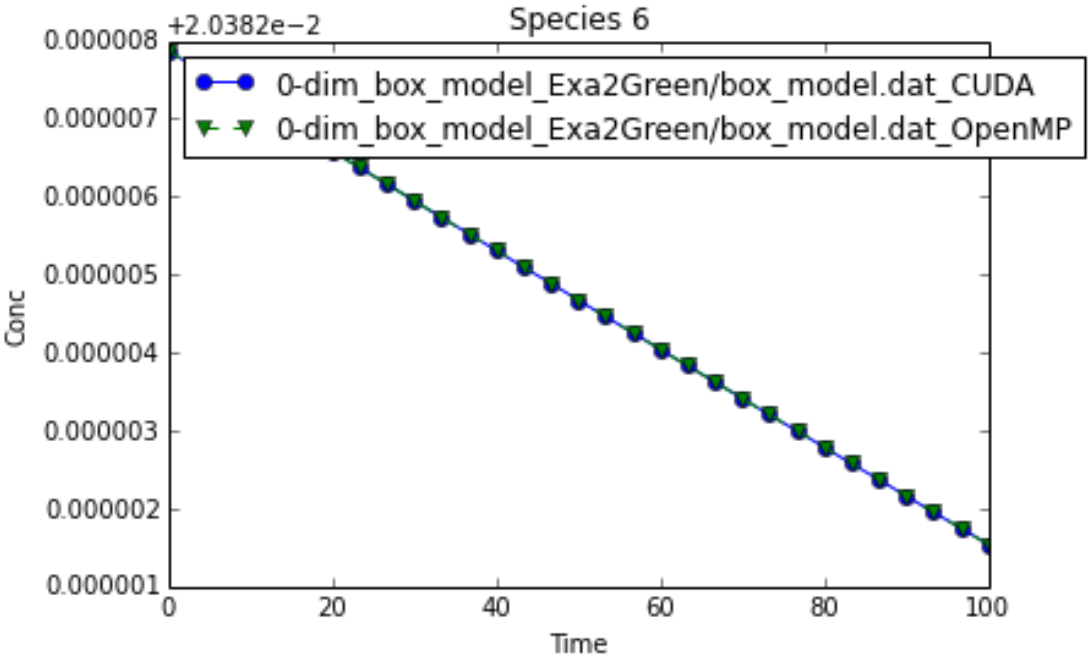

Species 1
1e−8+3.078e−5
0-dim_box_model_Exa2Green/box_model.dat_CUDA
0-dim_box_model_Exa2Green/box_model.dat_OpenMP
Conc
Time
0
1
2
3
4
5
6
0
20
40
60
80
100

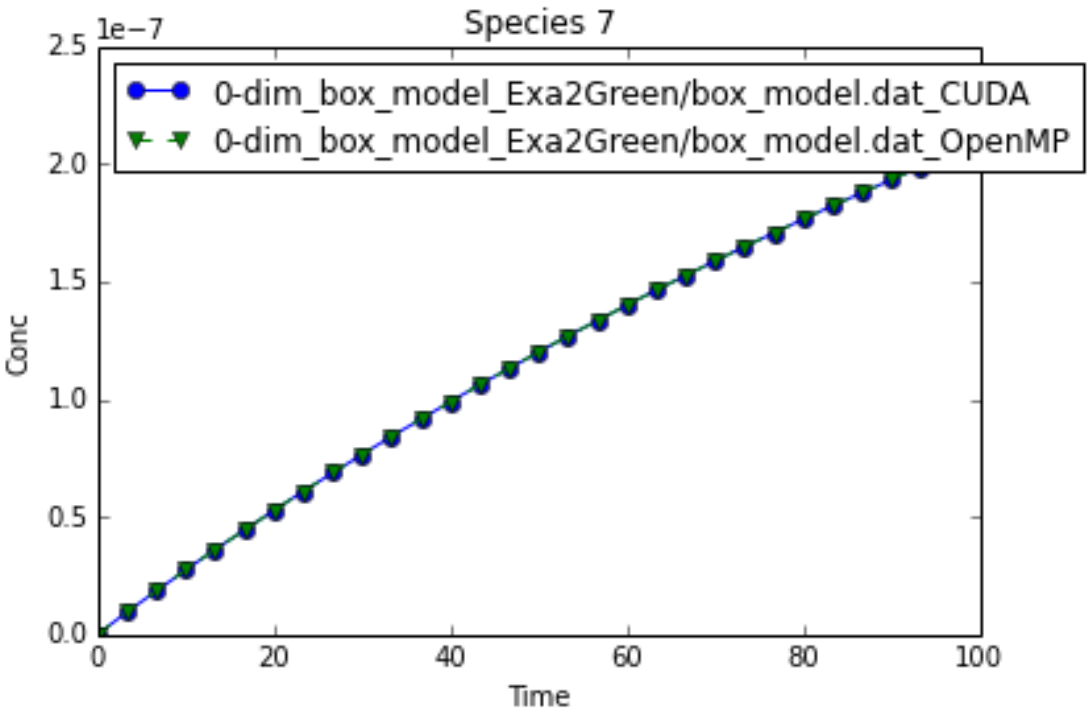

Species 2
1e−8+5.9304e−4
0-dim_box_model_Exa2Green/box_model.dat_CUDA
0-dim_box_model_Exa2Green/box_model.dat_OpenMP
Conc
Time
0
2
4
6
8
0
20
40
60
80
100


Species 3
1e−7
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Conc
0
20
40
60
80
100
Time
0-dim_box_model_Exa2Green/box_model.dat_CUDA
0-dim_box_model_Exa2Green/box_model.dat_OpenMP

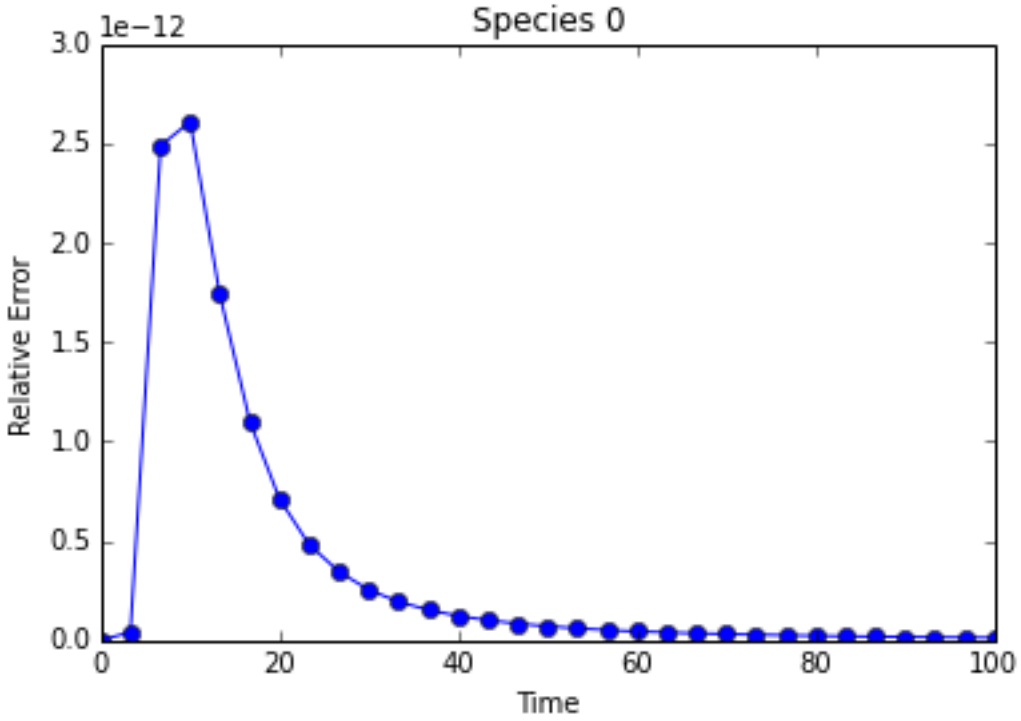

Species 4
0.007545
0.007540
0.007535
0.007530
0.007525
0.007520
0.007515
0.007510
Conc
0
20
40
60
80
100
Time
0-dim_box_model_Exa2Green/box_model.dat_CUDA
0-dim_box_model_Exa2Green/box_model.dat_OpenMP

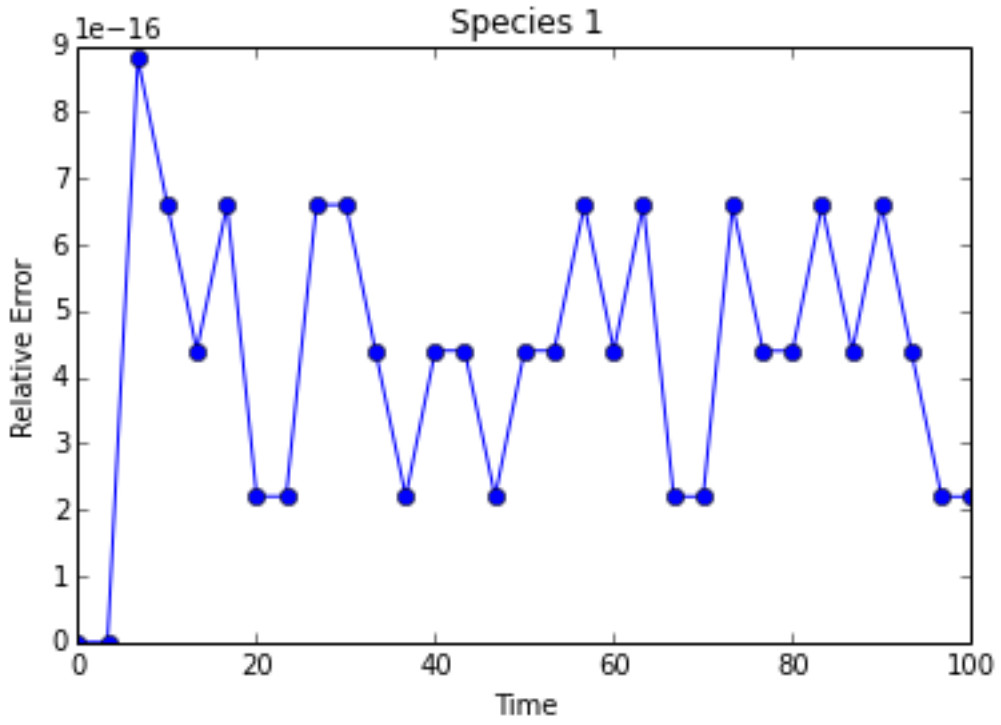

Species 5
0.000252
0.000250
0.000248
0.000246
0.000244
0.000242
0.000240
Conc
0
20
40
60
80
100
Time
0-dim_box_model_Exa2Green/box_model.dat_CUDA
0-dim_box_model_Exa2Green/box_model.dat_OpenMP

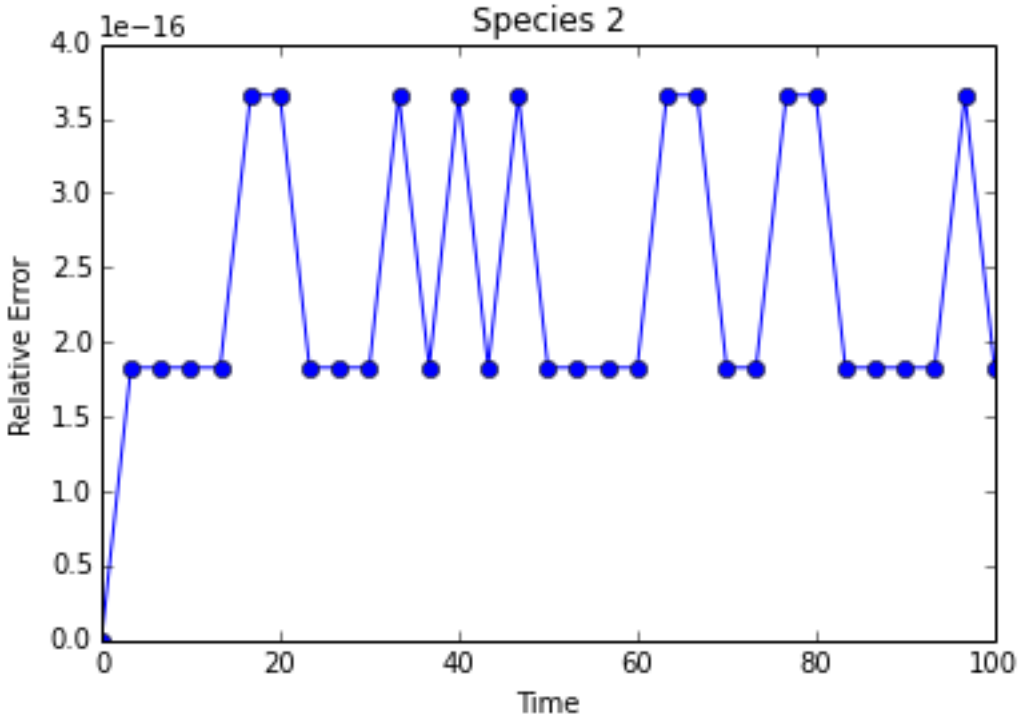

Species 6
+2.0382e−2
0.000008
0.000007
0.000006
0.000005
0.000004
0.000003
0.000002
0.000001
Conc
0
20
40
60
80
100
Time
0-dim_box_model_Exa2Green/box_model.dat_CUDA
0-dim_box_model_Exa2Green/box_model.dat_OpenMP

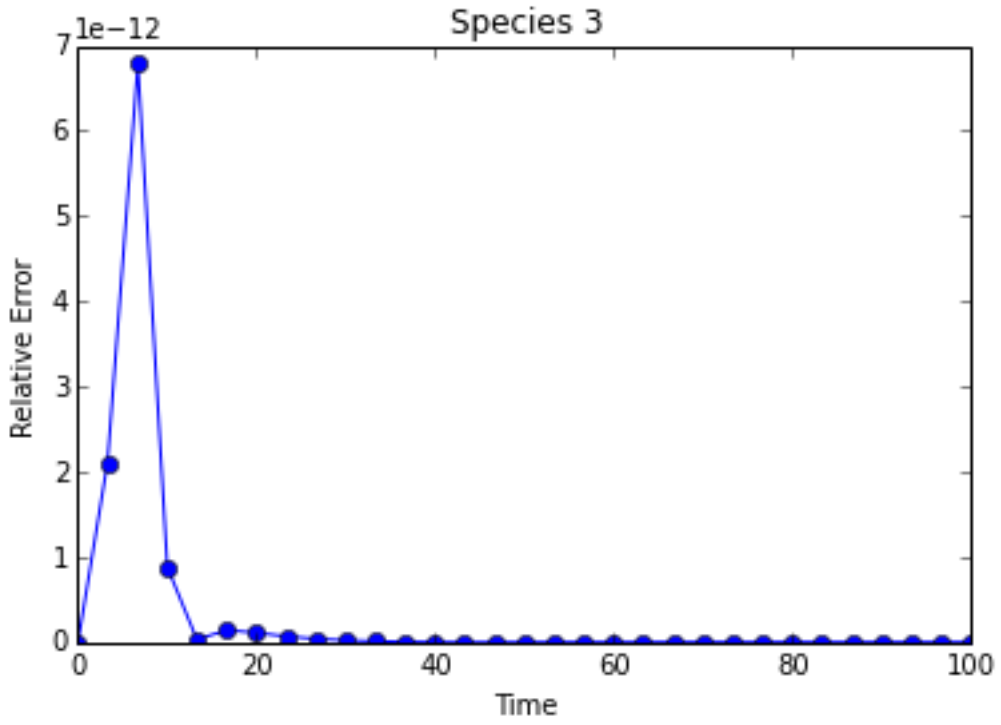


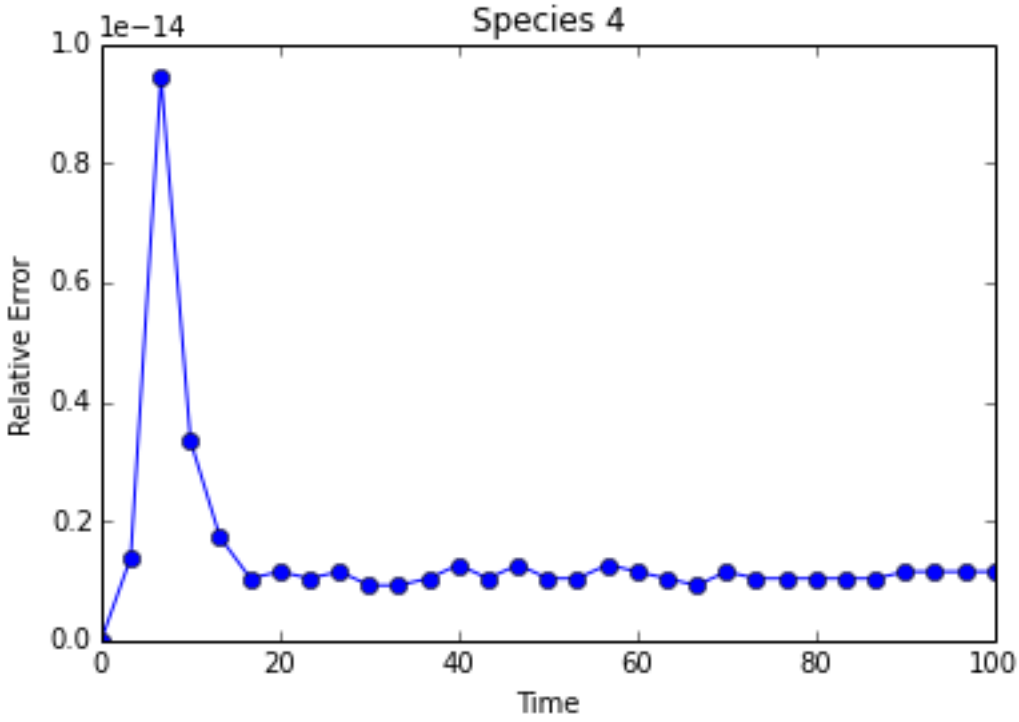
In [12]:
```
plot_dat([err_dat], ylabel='Relative Error')
```

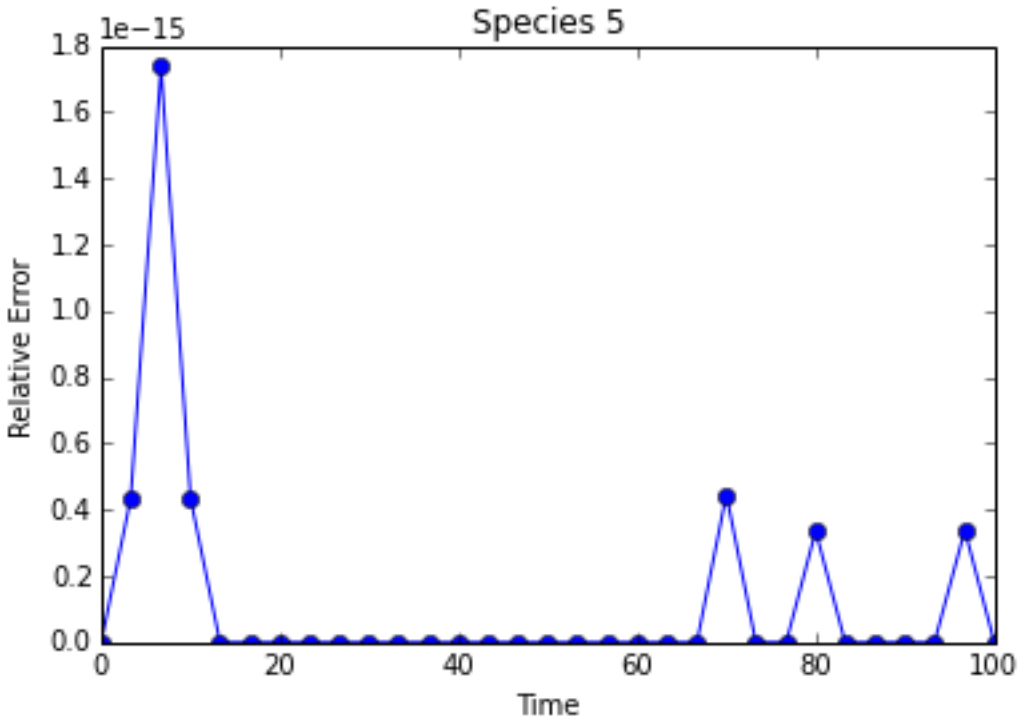


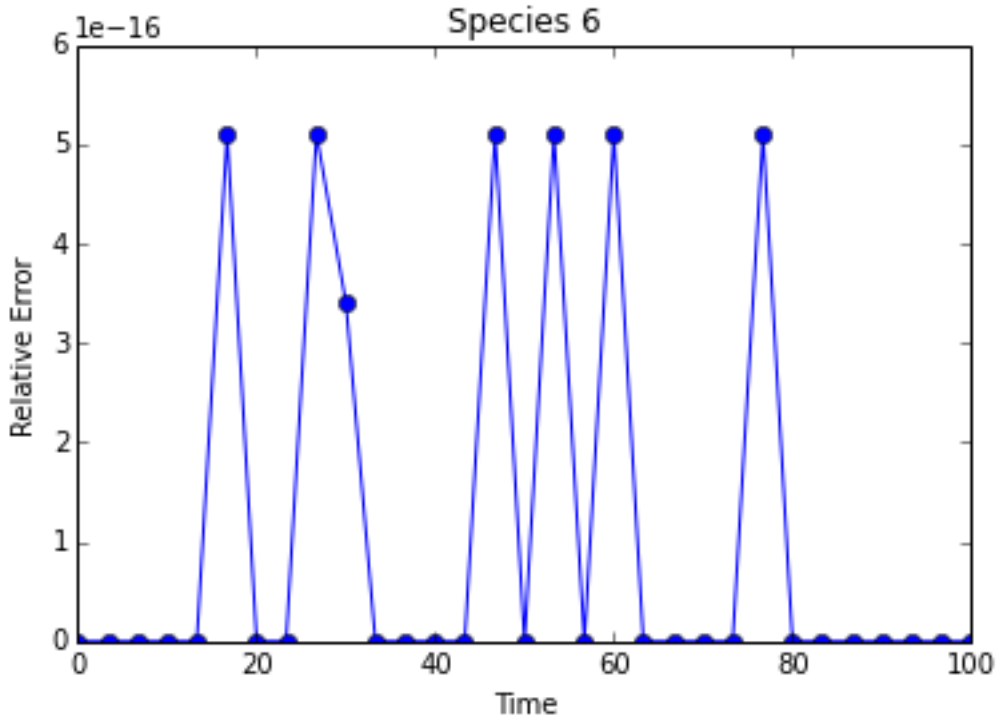

Species 1
1e−16
Relative Error
Time
0
1
2
3
4
5
6
7
8
9
0
20
40
60
80
100

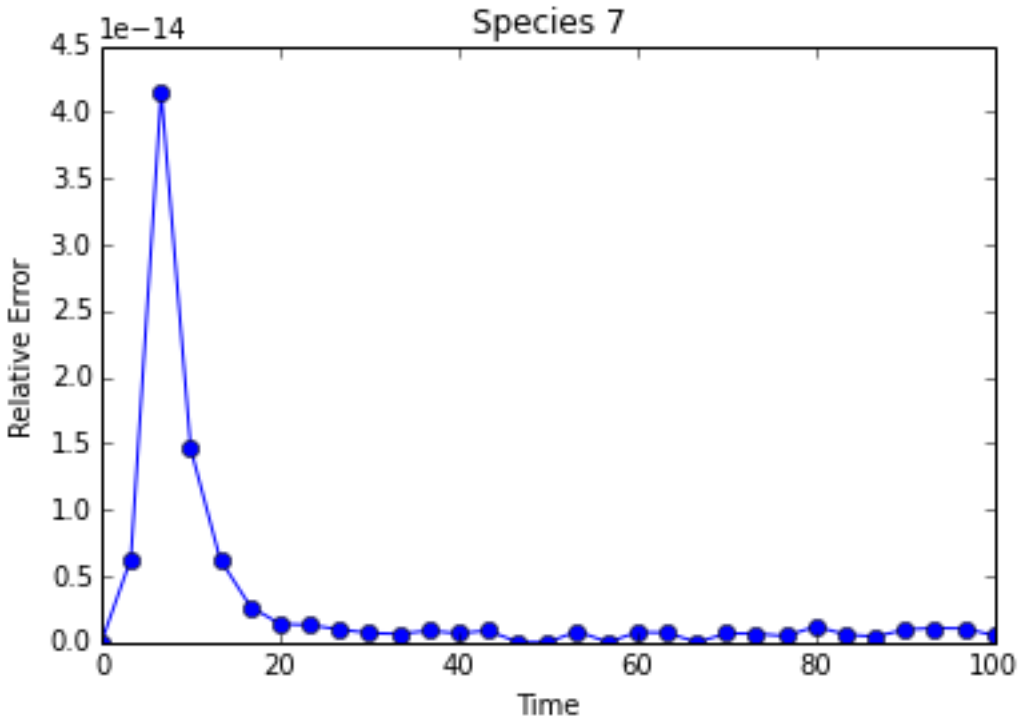

Species 2
1e−16
Relative Error
Time
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
20
40
60
80
100


Species 3
1e−12
7
6
5
4
3
2
1
0
Relative Error
0
20
40
60
80
100
Time


Species 4
1e−14
1.0
0.8
0.6
0.4
0.2
0.0
Relative Error
0
20
40
60
80
100
Time


Species 5
1e−15
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Relative Error
0
20
40
60
80
100
Time


Species 6
1e−16
6
5
4
3
2
1
0
Relative Error
0
20
40
60
80
100
Time


Species 7
1e−14
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Relative Error
0
20
40
60
80
100
Time

In [12]: