kpp_vs_kppa

Unknown Author

November 25, 2014

In [13]:
```
kppa_file = 'extended_box_model_Exa2Green/box_model.dat_CUDA'
kpp_file = 'extended_box_model_Exa2Green/kpp-2.2.1.dat'
```

In [14]:
```
%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)
```

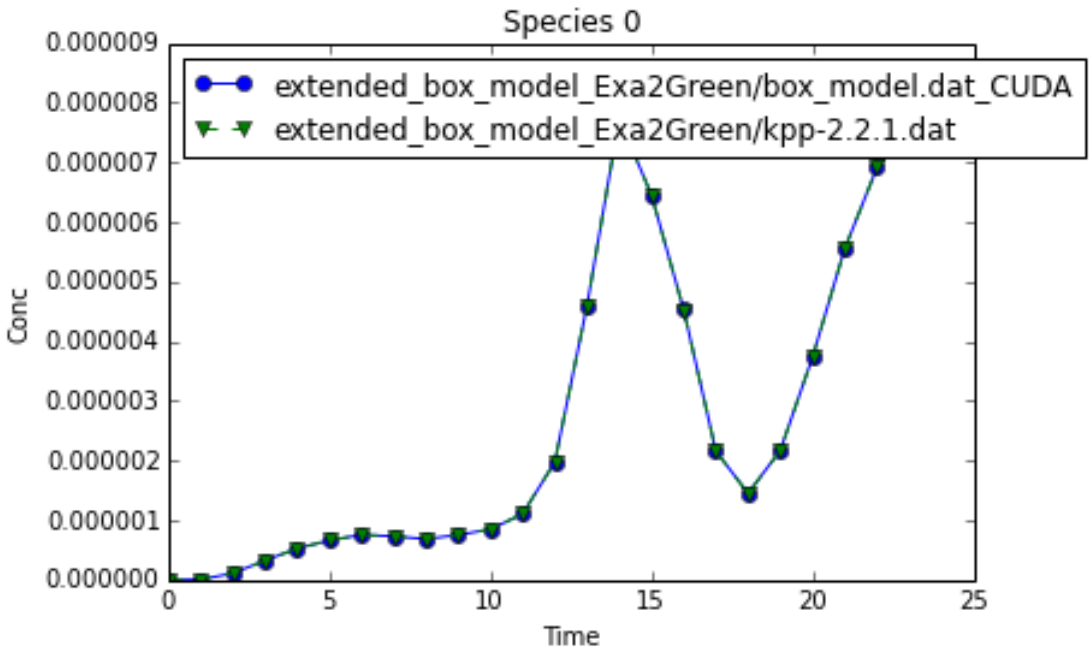
```
            err.append(e)
        if errPow > 0:
            snr = 20 * log10(sigPow / errPow)
        else:
            snr = float('inf')
        print 'SNR: %fdb' % snr
        if errCount:
            print '%d samples with relative error > %g' % (errCount, RTOL)
        return d1[0], err
```

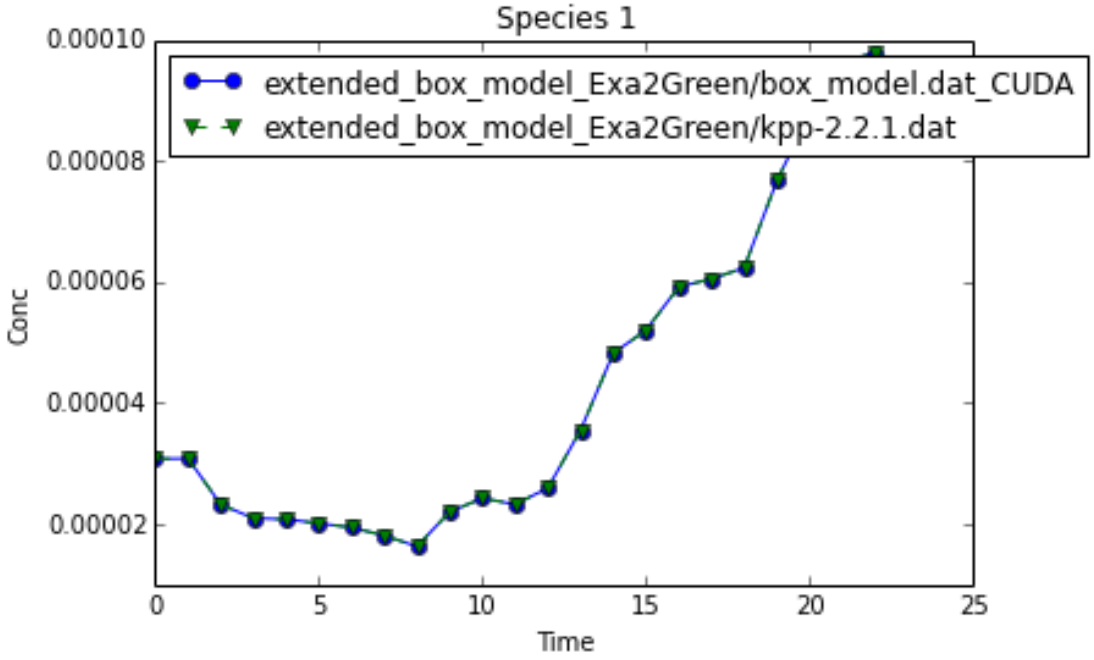
In [15]:
```
kppa_dat = read_datfile(kppa_file, 0, 1)
kpp_dat = read_datfile(kpp_file, 0, 1)
```

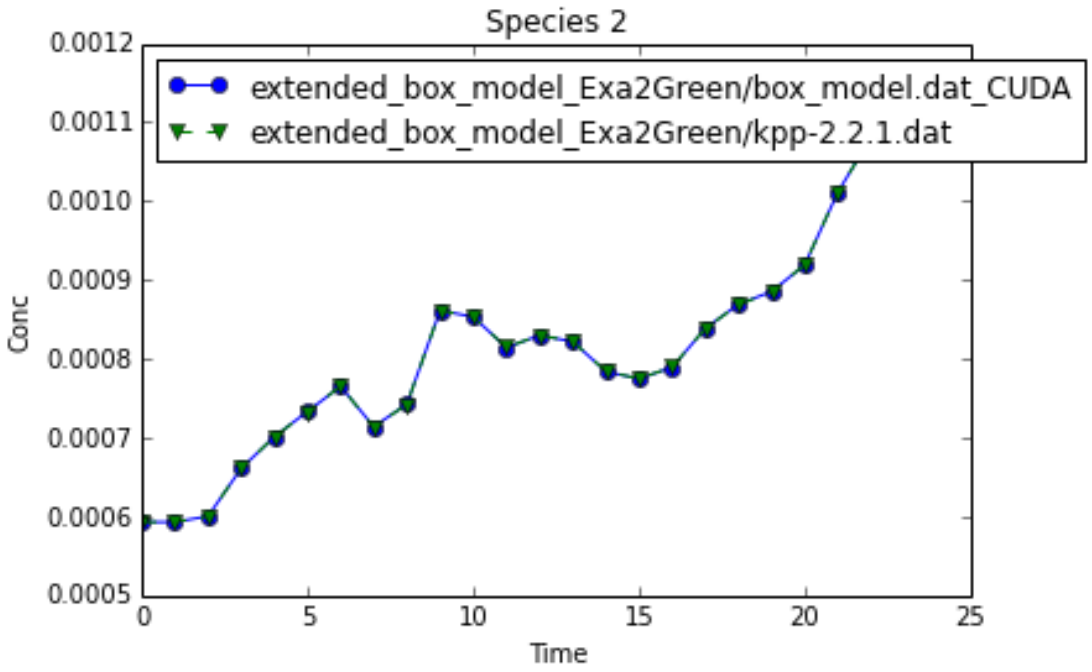
In [16]:
```
err_dat = calc_err(kpp_dat, kppa_dat)
```

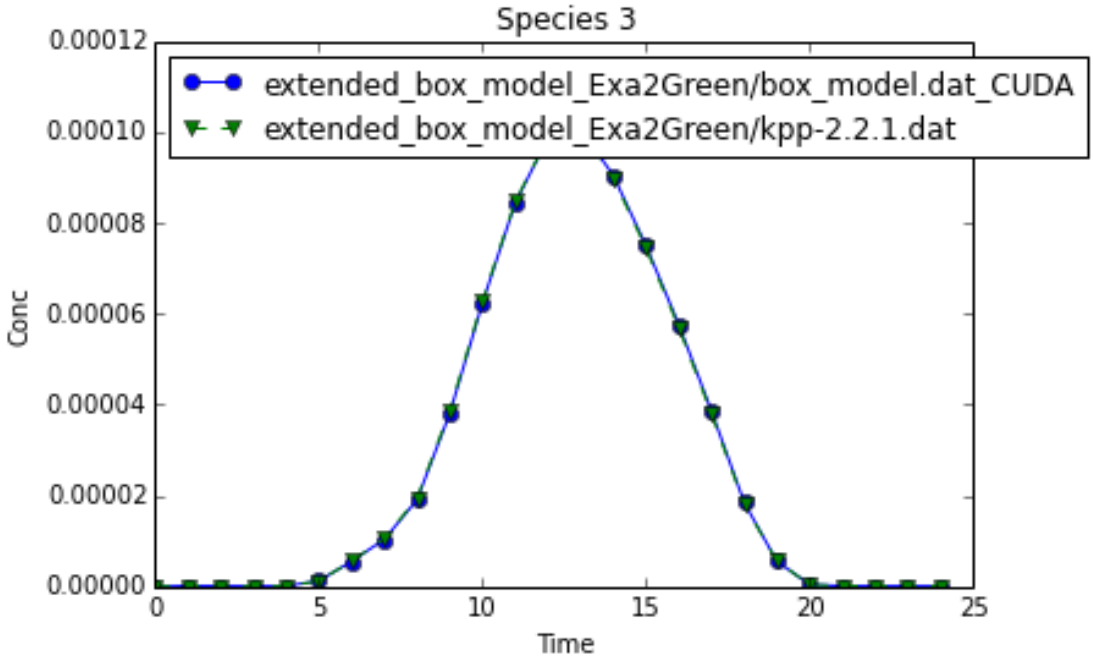
```
0.0328179 > 0.01: 1.21248e-06, 1.25362e-06
0.0131839 > 0.01: 0.00838765, 0.00827707
0.0357179 > 0.01: 0.00301928, 0.00313112
0.284233 > 0.01: 1.47252e-06, 1.05398e-06
0.0265835 > 0.01: 0.00044494, 0.000457091
0.0136652 > 0.01: 1.28911e-10, 1.30697e-10
0.0157434 > 0.01: 1.05342e-05, 1.03683e-05
0.0155812 > 0.01: 5.0805e-05, 5.16092e-05
0.0167379 > 0.01: 1.93813e-05, 1.90569e-05
0.0190176 > 0.01: 7.51184e-11, 7.36898e-11
0.0164722 > 0.01: 3.86554e-05, 3.80186e-05
0.0112297 > 0.01: 5.69911e-05, 5.76383e-05
0.0141985 > 0.01: 6.45114e-10, 6.54406e-10
0.0156143 > 0.01: 3.80322e-05, 3.86354e-05
0.0183154 > 0.01: 0.000137925, 0.000135399
0.0182973 > 0.01: 1.81486e-05, 1.84868e-05
0.0235857 > 0.01: 0.000253063, 0.000247094
0.0240564 > 0.01: 2.06929e-10, 2.01951e-10
0.0147845 > 0.01: 2.84859e-09, 2.89134e-09
0.032188 > 0.01: 4.41267e-07, 4.27063e-07
0.0312234 > 0.01: 0.00426031, 0.00412729
0.0166782 > 0.01: 0.00806905, 0.00793447
0.0271128 > 0.01: 3.38665e-08, 3.48103e-08
SNR: 106.323018db
23 samples with relative error > 0.01
```

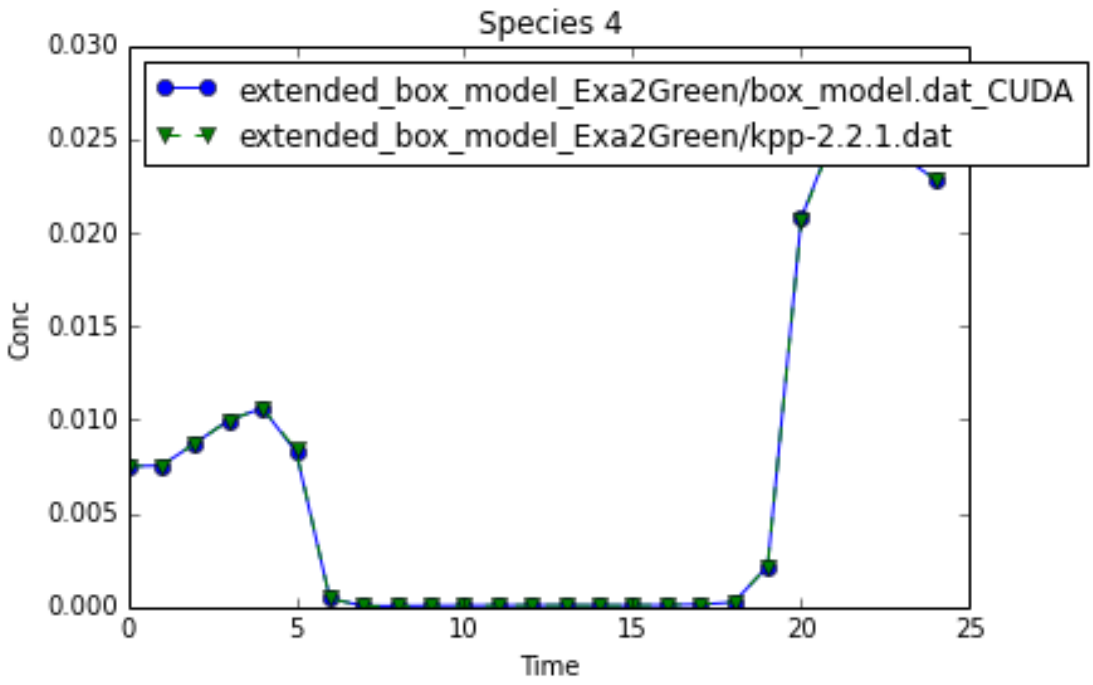
In [17]:
```
plot_dat([kppa_dat, kpp_dat], names=[kppa_file, kpp_file], titles=None)
```

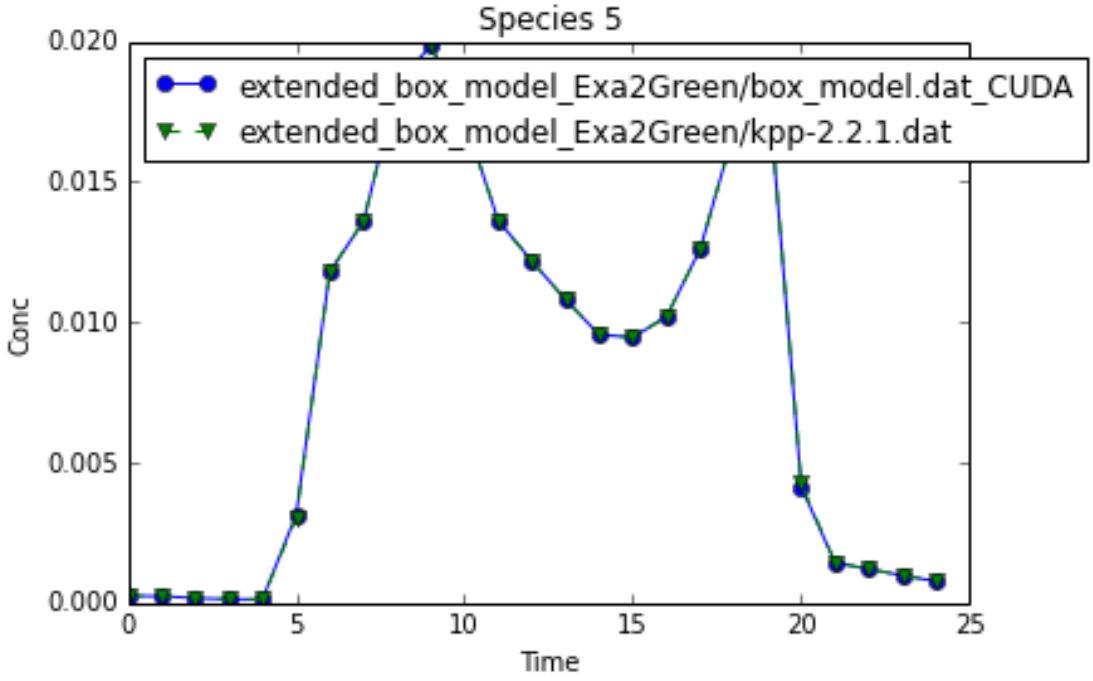

Species 0
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp-2.2.1.dat
Conc
Time
0.000009
0.000008
0.000007
0.000006
0.000005
0.000004
0.000003
0.000002
0.000001
0.000000
0
5
10
15
20
25

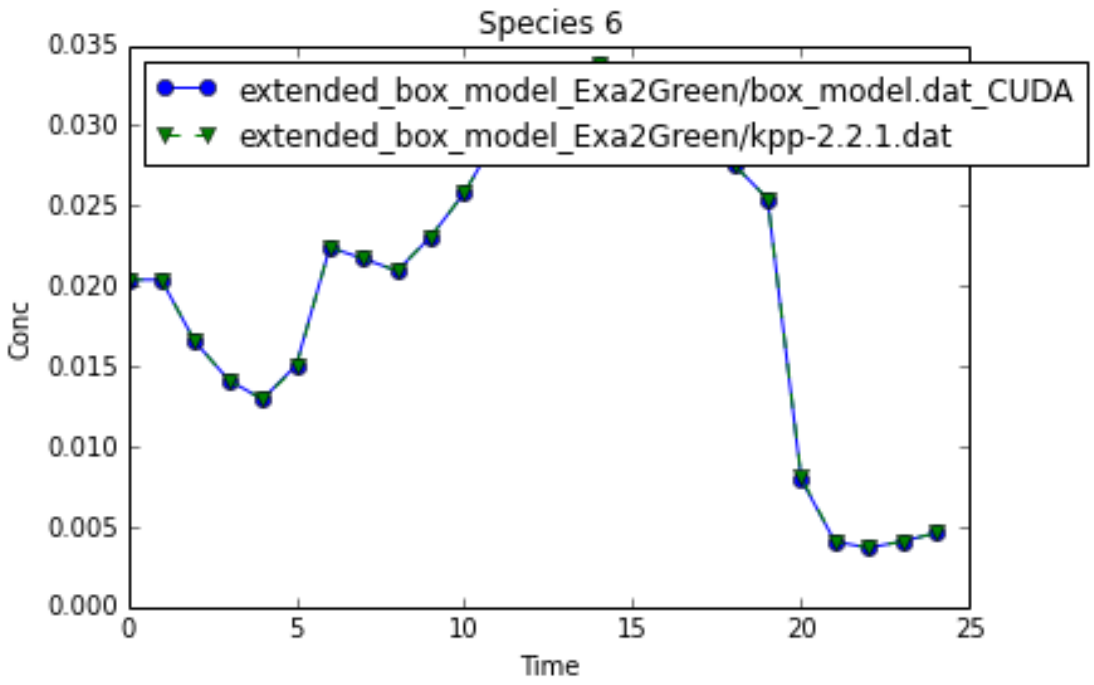

Species 1
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp-2.2.1.dat
Conc
Time
0.00010
0.00008
0.00006
0.00004
0.00002
0
5
10
15
20
25

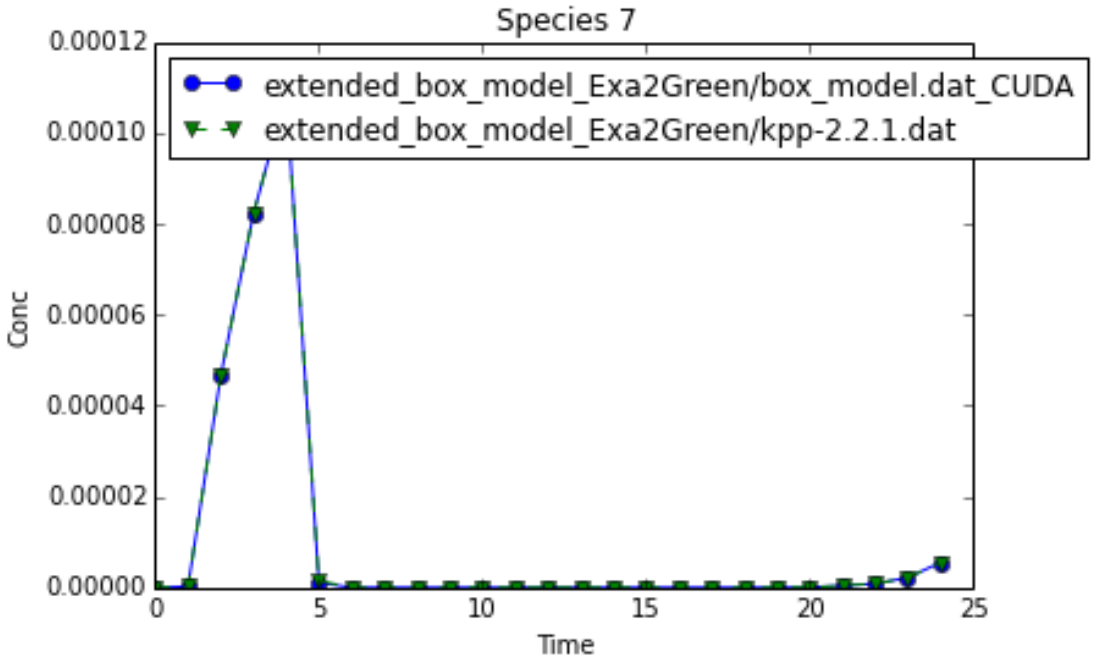

Species 2
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp-2.2.1.dat
Conc
Time
0.0012
0.0011
0.0010
0.0009
0.0008
0.0007
0.0006
0.0005
0
5
10
15
20
25


Species 3
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp-2.2.1.dat
Conc
Time
0.00012
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
0
5
10
15
20
25

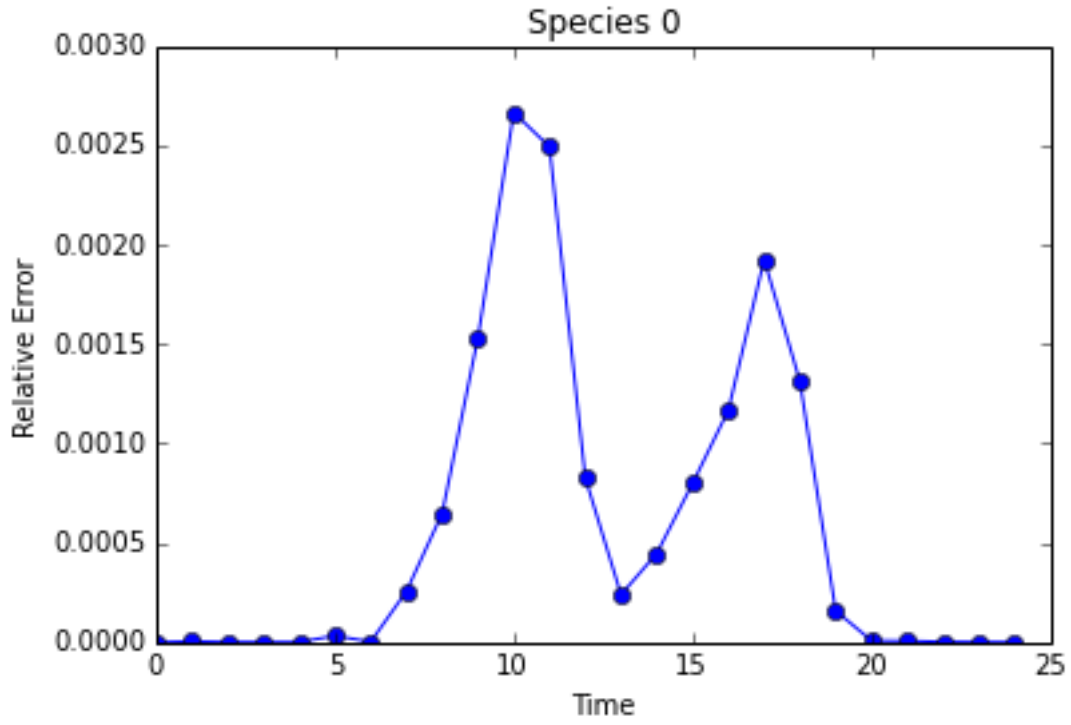

Species 4
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp-2.2.1.dat
Conc
Time
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
5
10
15
20
25

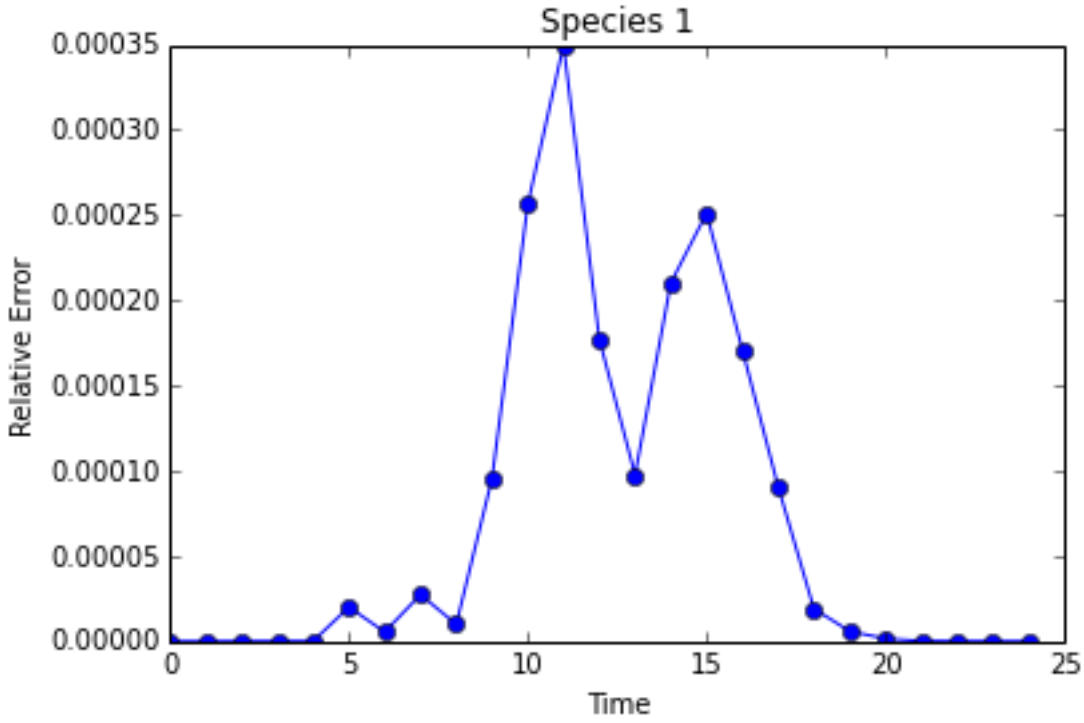

Species 5
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp-2.2.1.dat
Conc
Time
0.020
0.015
0.010
0.005
0.000
0
5
10
15
20
25

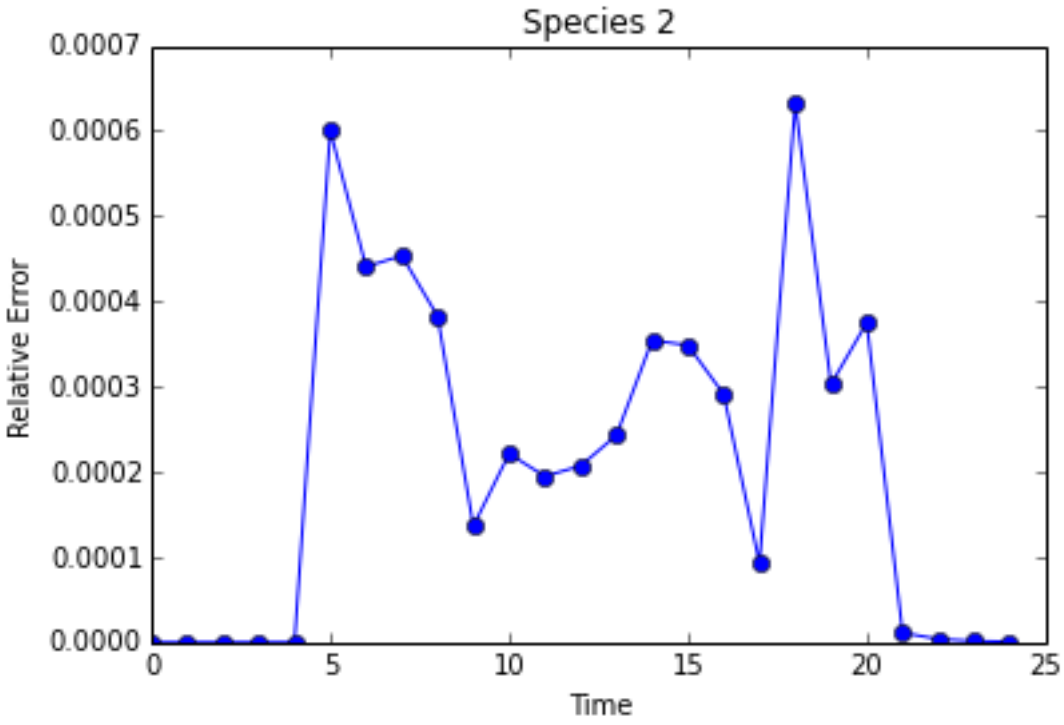


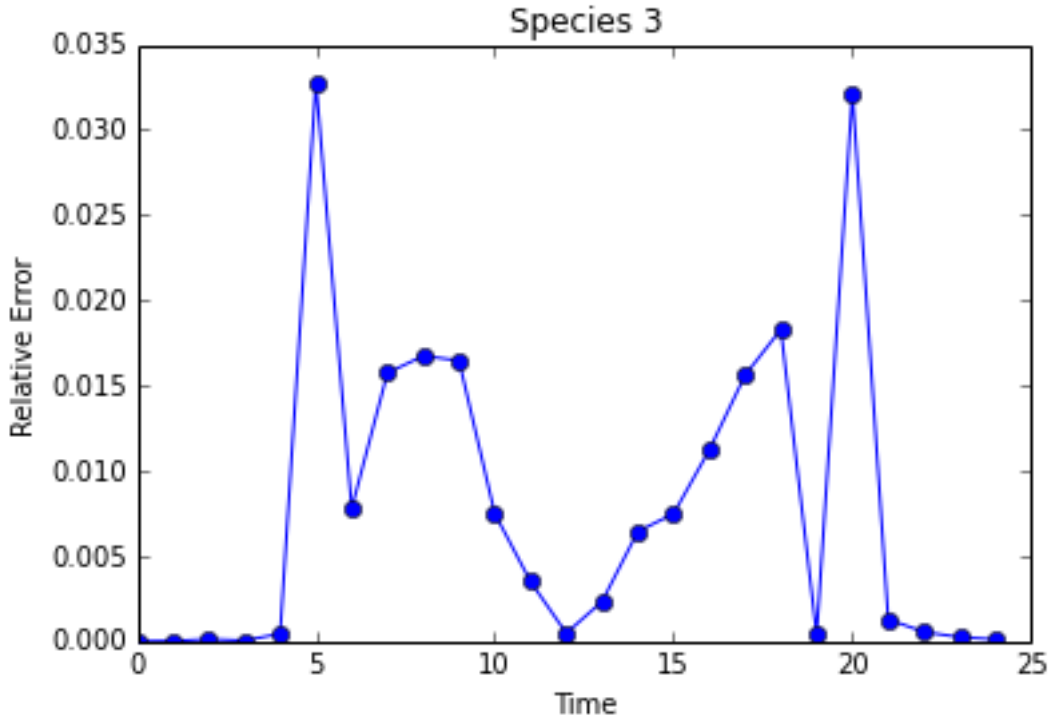


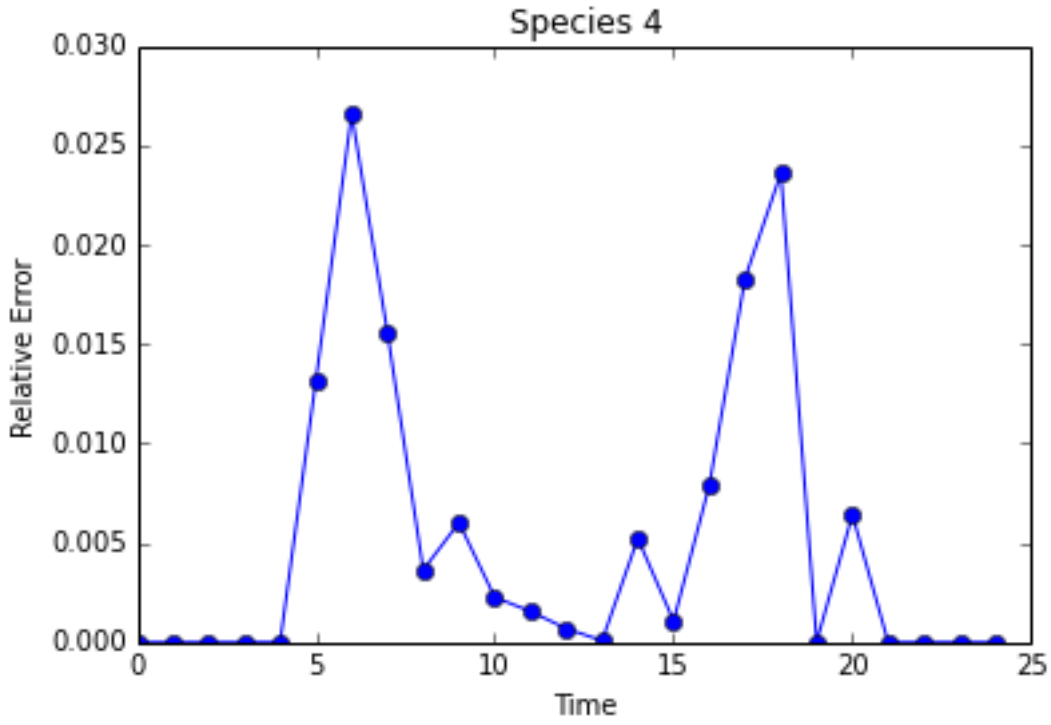
In [18]:
```
plot_dat([err_dat], ylabel='Relative Error')
```

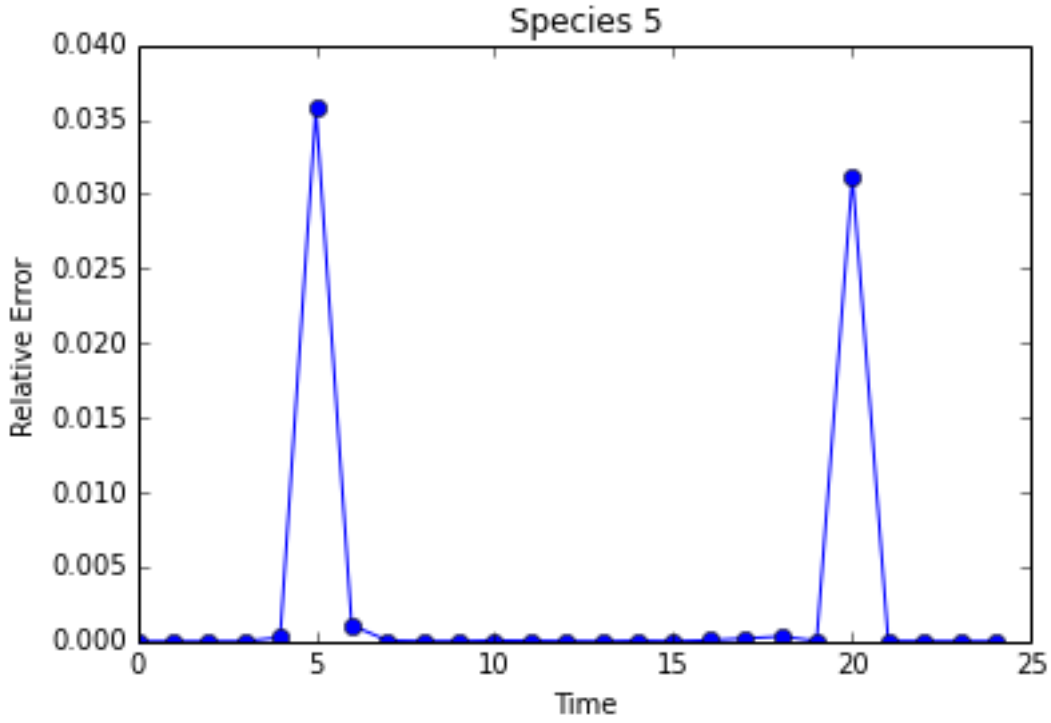

Species 0
Relative Error
Time
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
5
10
15
20
25

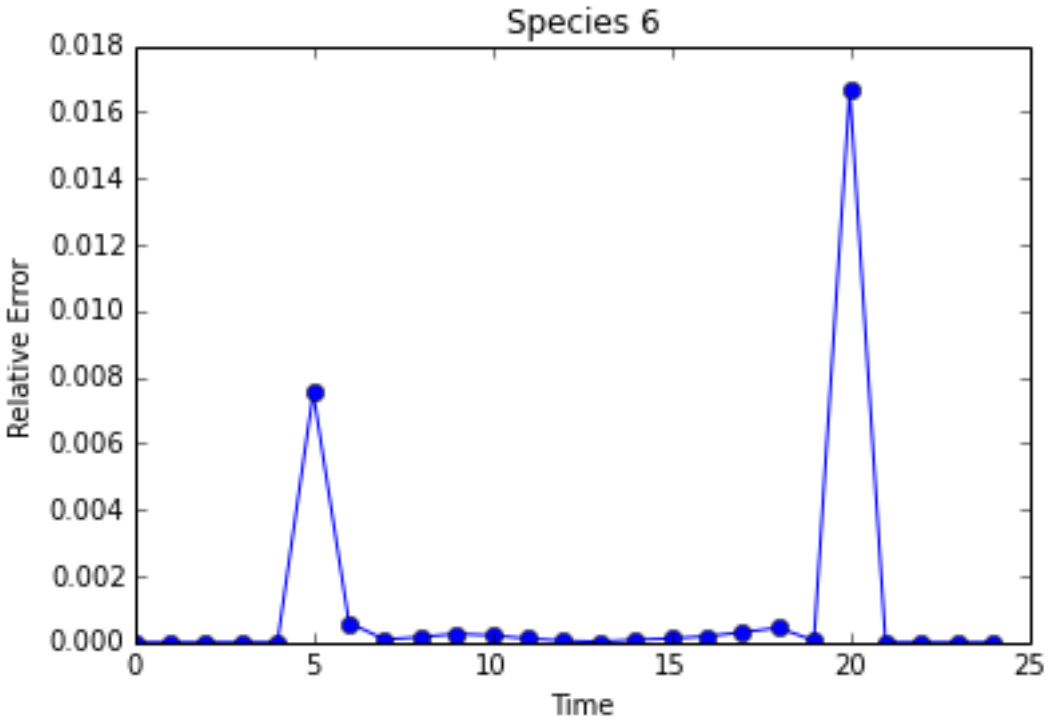

Species 1
Relative Error
Time
0.00035
0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
0
5
10
15
20
25

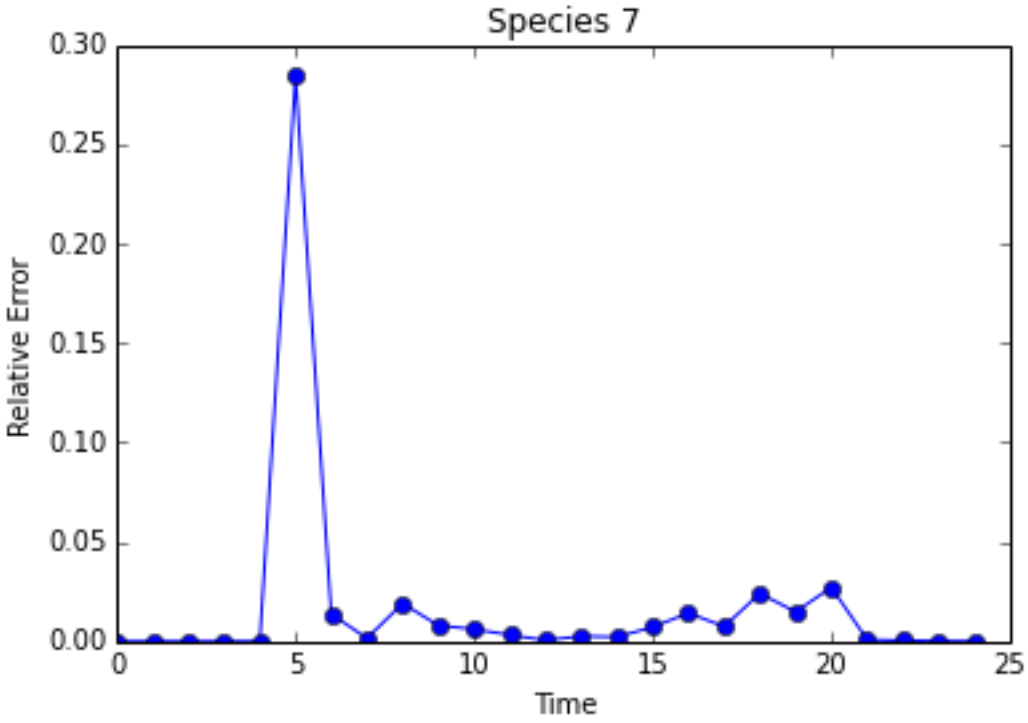

Species 2
Relative Error
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
0
5
10
15
20
25
Time


Species 3
Relative Error
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
5
10
15
20
25
Time


Species 4
Relative Error
Time
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
5
10
15
20
25


Species 5
Relative Error
Time
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
5
10
15
20
25


Species 6
Relative Error
Time
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
5
10
15
20
25


Species 7
Relative Error
Time
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25

In [18]: