kpp_vs_kppa

Unknown Author

May 27, 2015

In [31]:

```
kpp_file_1 = 'extended_box_model_Exa2Green/box_model.dat_CUDA'
kpp_file_2 = 'extended_box_model_Exa2Green/kpp_input.dat'
```

In [32]:

```
%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 [33]:

```
kpp_dat_1 = read_datfile(kpp_file_1, 0, 1)
kpp_dat_2 = read_datfile(kpp_file_2, 0, 1)
```

In [34]:

```
err_dat = calc_err(kpp_dat_1, kpp_dat_2)
```

```
0.710413 > 0.01: 0.000414383, 0.00012
0.254489 > 0.01: 0.0309452, 0.02307
0.156141 > 0.01: 7.24622, 8.587
0.999909 > 0.01: 1.87567e-05, 0.2059
0.16939 > 0.01: 19.8529, 16.49
0.795226 > 0.01: 0.00521062, 0.001067
0.926389 > 0.01: 1.84395e-07, 2.505e-06
0.337159 > 0.01: 0.000585359, 0.000388
0.115121 > 0.01: 0.0234382, 0.02074
0.0251616 > 0.01: 0.640371, 0.6569
0.130004 > 0.01: 8.44157, 9.703
0.999951 > 0.01: 9.05982e-06, 0.1853
0.124496 > 0.01: 16.0593, 14.06
0.648291 > 0.01: 0.00340054, 0.001196
0.863058 > 0.01: 2.46633e-07, 1.801e-06
0.296692 > 0.01: 0.000890791, 0.0006265
0.0250583 > 0.01: 0.0210679, 0.02054
0.074524 > 0.01: 0.744266, 0.6888
0.0608121 > 0.01: 9.5985, 10.22
0.999949 > 0.01: 9.6031e-06, 0.1867
0.066632 > 0.01: 13.7138, 12.8
0.645125 > 0.01: 0.00303487, 0.001077
0.869761 > 0.01: 1.97702e-07, 1.518e-06
0.254427 > 0.01: 0.00104068, 0.0007759
0.0447731 > 0.01: 0.0208327, 0.0199
0.0982486 > 0.01: 0.777598, 0.7012
0.0421641 > 0.01: 10.1435, 10.59
0.999056 > 0.01: 0.00020107, 0.213
0.0597928 > 0.01: 12.4653, 11.72
0.662023 > 0.01: 0.00257858, 0.0008715
0.867467 > 0.01: 1.90715e-07, 1.439e-06
0.0656872 > 0.01: 0.00082014, 0.0008778
0.0477121 > 0.01: 0.020204, 0.01924
0.159376 > 0.01: 0.841875, 0.7077
0.0831815 > 0.01: 10.2684, 11.2
0.570052 > 0.01: 0.692642, 0.2978
0.166981 > 0.01: 12.3887, 10.32
0.694676 > 0.01: 0.000193545, 0.0006339
0.71541 > 0.01: 5.14776e-06, 1.465e-06
0.0344272 > 0.01: 0.000873575, 0.0008435
0.0900204 > 0.01: 0.0195499, 0.01779
0.0933215 > 0.01: 0.736314, 0.6676
```

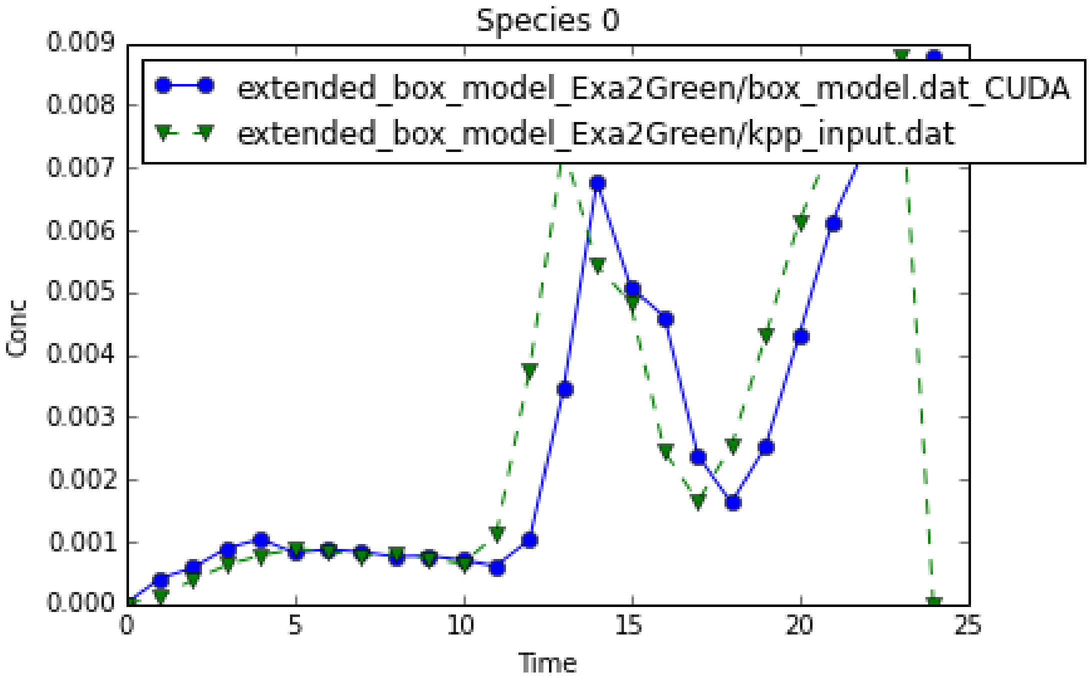
```
0.308299 > 0.01: 8.29349, 11.99
0.692182 > 0.01: 3.30065, 1.016
0.378171 > 0.01: 13.5487, 8.425
0.783437 > 0.01: 2.26958e-05, 0.0001048
0.635048 > 0.01: 6.0282e-06, 2.2e-06
0.0728288 > 0.01: 0.000833934, 0.0007732
0.122286 > 0.01: 0.018195, 0.01597
0.430585 > 0.01: 6.95825, 12.22
0.0308102 > 0.01: 5.95549, 5.772
0.465902 > 0.01: 13.696, 7.315
0.124726 > 0.01: 1.05733e-05, 1.208e-05
0.730927 > 0.01: 1.16846e-05, 3.144e-06
0.0450564 > 0.01: 0.00075813, 0.0007939
0.238463 > 0.01: 0.016434, 0.02158
0.0820388 > 0.01: 0.754472, 0.8219
0.308716 > 0.01: 7.44513, 10.77
0.153261 > 0.01: 10.3527, 8.766
0.155512 > 0.01: 12.5401, 10.59
0.401782 > 0.01: 6.16165e-06, 1.03e-05
0.660371 > 0.01: 1.81816e-05, 6.175e-06
0.0428317 > 0.01: 0.000770815, 0.0007378
0.0812422 > 0.01: 0.0217194, 0.02364
0.0621278 > 0.01: 0.859819, 0.8064
0.147892 > 0.01: 7.7218, 9.062
0.312916 > 0.01: 11.5386, 7.928
0.0415739 > 0.01: 14.2901, 14.91
0.508933 > 0.01: 6.90932e-06, 1.407e-05
0.539615 > 0.01: 2.65865e-05, 1.224e-05
0.0867334 > 0.01: 0.000706803, 0.0006455
0.0695669 > 0.01: 0.0235482, 0.02191
0.100585 > 0.01: 0.842436, 0.7577
0.0542911 > 0.01: 7.09187, 7.499
0.377415 > 0.01: 9.52641, 5.931
0.119994 > 0.01: 17.7321, 20.15
0.544843 > 0.01: 1.00317e-05, 2.204e-05
0.452045 > 0.01: 3.94193e-05, 2.16e-05
0.449194 > 0.01: 0.000608641, 0.001105
0.0900256 > 0.01: 0.021712, 0.02386
0.0702279 > 0.01: 0.803745, 0.7473
0.10442 > 0.01: 6.15353, 6.871
0.235282 > 0.01: 6.80251, 5.202
0.0162397 > 0.01: 22.5379, 22.91
0.413892 > 0.01: 1.59304e-05, 2.718e-05
0.460509 > 0.01: 5.63864e-05, 3.042e-05
0.725827 > 0.01: 0.00102705, 0.003746
0.306625 > 0.01: 0.0235331, 0.03394
0.104698 > 0.01: 0.815032, 0.7297
0.0793153 > 0.01: 5.79203, 6.291
0.225313 > 0.01: 5.70941, 4.423
0.38882 > 0.01: 2.0554e-05, 3.363e-05
0.463533 > 0.01: 6.84292e-05, 3.671e-05
0.531846 > 0.01: 0.00344233, 0.007353
0.309254 > 0.01: 0.0333077, 0.04822
```

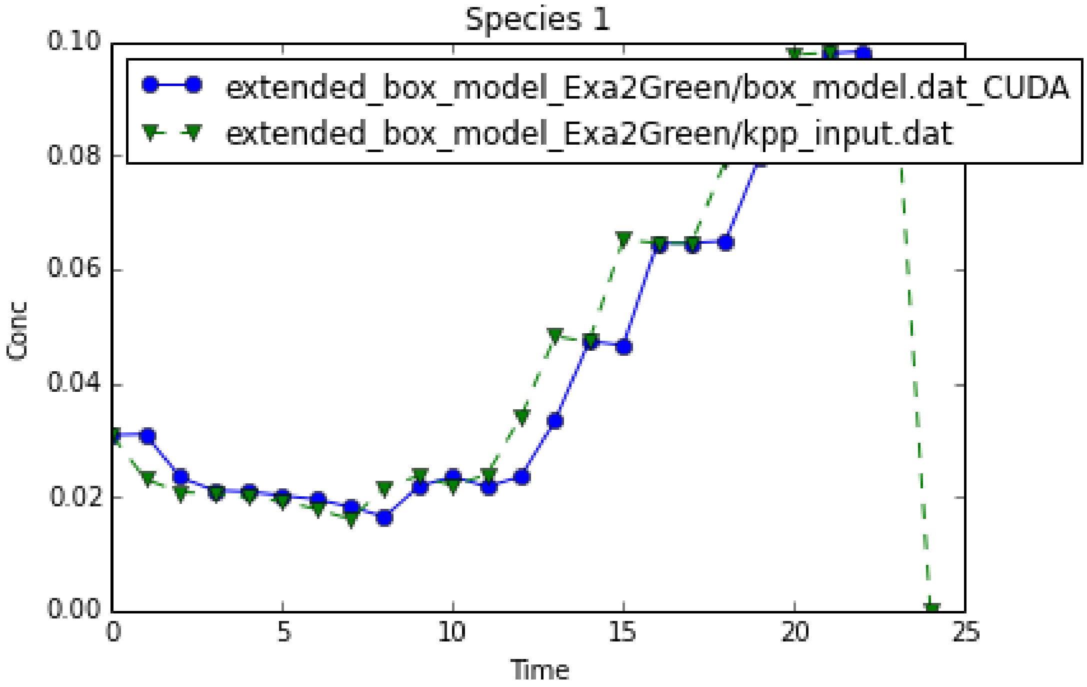
```
0.138307 > 0.01: 0.805508, 0.6941
0.0846589 > 0.01: 5.39777, 5.897
0.244862 > 0.01: 4.71569, 3.561
0.0451277 > 0.01: 27.3649, 26.13
0.408617 > 0.01: 2.55418e-05, 4.319e-05
0.527732 > 0.01: 7.31152e-05, 3.453e-05
0.19786 > 0.01: 0.00676939, 0.00543
0.108831 > 0.01: 0.766858, 0.6834
0.250194 > 0.01: 5.0387, 6.72
0.131562 > 0.01: 3.87823, 3.368
0.186328 > 0.01: 28.2423, 22.98
0.354205 > 0.01: 3.00424e-05, 4.652e-05
0.663414 > 0.01: 6.59861e-05, 2.221e-05
0.0512559 > 0.01: 0.00508989, 0.004829
0.283976 > 0.01: 0.0466275, 0.06512
0.112113 > 0.01: 0.765413, 0.6796
0.089724 > 0.01: 5.51263, 6.056
0.0597657 > 0.01: 4.11706, 3.871
0.0568498 > 0.01: 25.2876, 23.85
0.260425 > 0.01: 2.7808e-05, 3.76e-05
0.403081 > 0.01: 4.6673e-05, 2.786e-05
0.46939 > 0.01: 0.00460602, 0.002444
0.0683101 > 0.01: 0.752289, 0.7009
0.247046 > 0.01: 6.04095, 8.023
0.218651 > 0.01: 3.53404, 4.523
0.169691 > 0.01: 24.6776, 20.49
0.0632035 > 0.01: 3.48863e-05, 3.724e-05
0.477595 > 0.01: 2.99959e-05, 1.567e-05
0.306701 > 0.01: 0.00237848, 0.001649
0.0618724 > 0.01: 0.786034, 0.7374
0.300851 > 0.01: 8.45271, 12.09
0.222583 > 0.01: 3.80779, 4.898
0.279078 > 0.01: 20.6402, 14.88
0.0266856 > 0.01: 3.81788e-05, 3.716e-05
0.515376 > 0.01: 1.53108e-05, 7.42e-06
0.357343 > 0.01: 0.00162656, 0.002531
0.179932 > 0.01: 0.0649166, 0.07916
0.0299977 > 0.01: 0.829689, 0.8048
0.248906 > 0.01: 13.0991, 17.44
0.0878925 > 0.01: 3.65308, 3.332
0.30545 > 0.01: 14.2193, 9.876
0.198615 > 0.01: 4.24253e-05, 5.294e-05
0.486748 > 0.01: 6.84264e-06, 3.512e-06
0.41265 > 0.01: 0.00251856, 0.004288
0.112576 > 0.01: 0.0797261, 0.08984
0.0453839 > 0.01: 0.870801, 0.9122
0.160701 > 0.01: 19.3374, 23.04
0.332338 > 0.01: 1.28057, 1.918
0.284619 > 0.01: 8.07402, 5.776
0.156729 > 0.01: 9.3078e-05, 7.849e-05
0.0272741 > 0.01: 2.15374e-06, 2.095e-06
0.300816 > 0.01: 0.0042874, 0.006132
0.0760825 > 0.01: 0.0902021, 0.09763
```

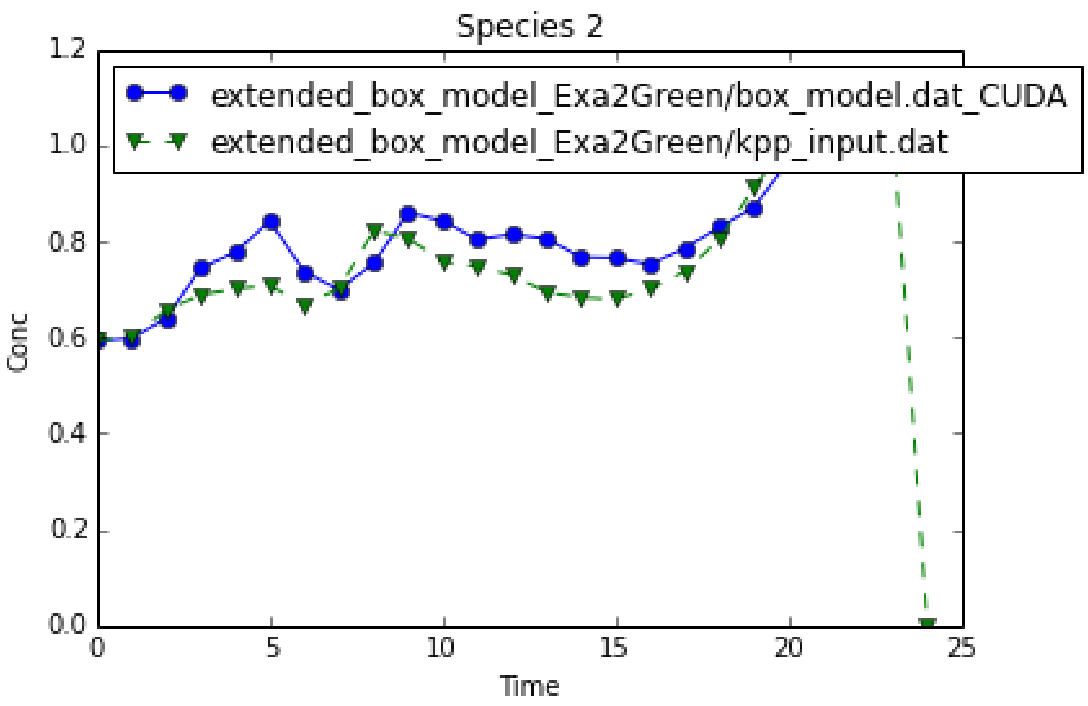
```
0.0396031 > 0.01: 0.96808, 1.008
0.0172419 > 0.01: 24.795, 25.23
0.986877 > 0.01: 0.019172, 1.461
0.0686788 > 0.01: 3.86033, 4.145
0.73352 > 0.01: 0.000345392, 9.204e-05
0.825142 > 0.01: 3.03379e-07, 1.735e-06
0.167853 > 0.01: 0.00613542, 0.007373
0.0538848 > 0.01: 26.5718, 25.14
0.994985 > 0.01: 0.00626344, 1.249
0.270156 > 0.01: 2.66247, 3.648
0.676029 > 0.01: 0.000299502, 9.703e-05
0.878605 > 0.01: 1.793e-07, 1.477e-06
0.0642065 > 0.01: 0.00737405, 0.00788
0.0443969 > 0.01: 0.0982102, 0.09385
0.0303905 > 0.01: 1.12726, 1.093
0.0887361 > 0.01: 26.2932, 23.96
0.989821 > 0.01: 0.0114104, 1.121
0.357213 > 0.01: 2.39824, 3.731
0.598881 > 0.01: 0.000273235, 0.0001096
0.879986 > 0.01: 1.673e-07, 1.394e-06
0.101302 > 0.01: 0.00788248, 0.008771
0.025376 > 0.01: 0.0940465, 0.09166
0.079152 > 0.01: 1.1446, 1.054
0.105649 > 0.01: 24.979, 22.34
0.993405 > 0.01: 0.0060078, 0.911
0.385715 > 0.01: 2.59413, 4.223
0.584931 > 0.01: 0.000319224, 0.0001325
0.887212 > 0.01: 1.55422e-07, 1.378e-06
1 > 0.01: 0.00878439, 0
1 > 0.01: 0.0918722, 0
1 > 0.01: 1.10444, 0
1 > 0.01: 23.1185, 0
1 > 0.01: 0.000270325, 0
1 > 0.01: 3.2589, 0
1 > 0.01: 0.000465294, 0
1 > 0.01: 1.5601e-07, 0
SNR: 22.621857db
184 samples with relative error > 0.01
```

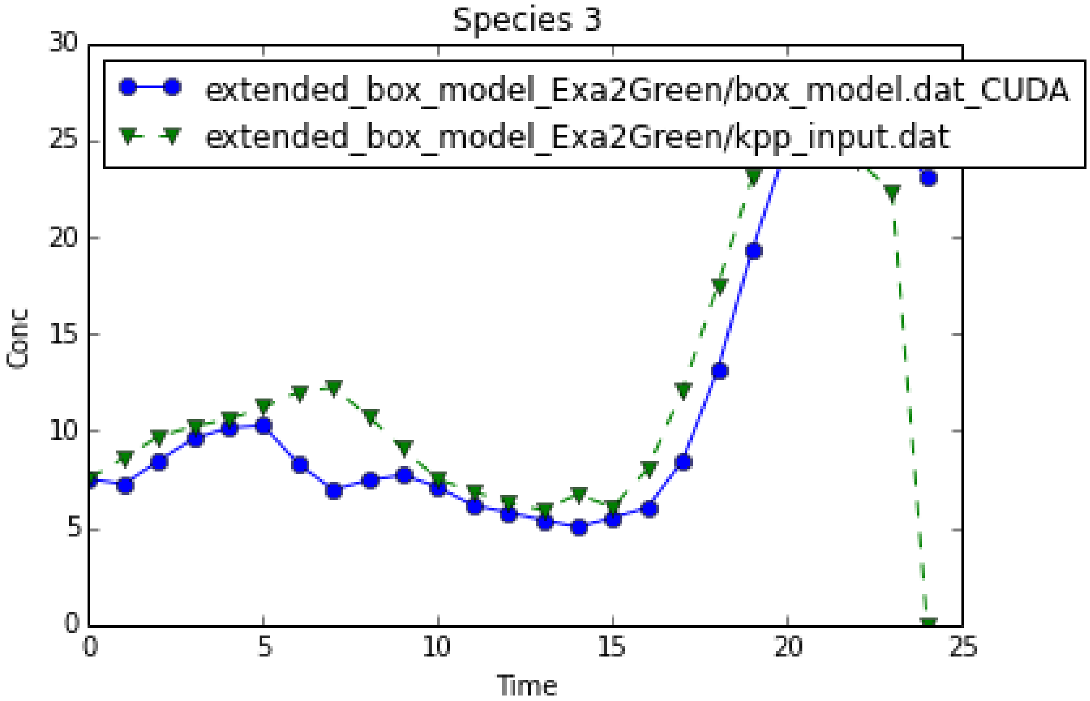
In [35]:

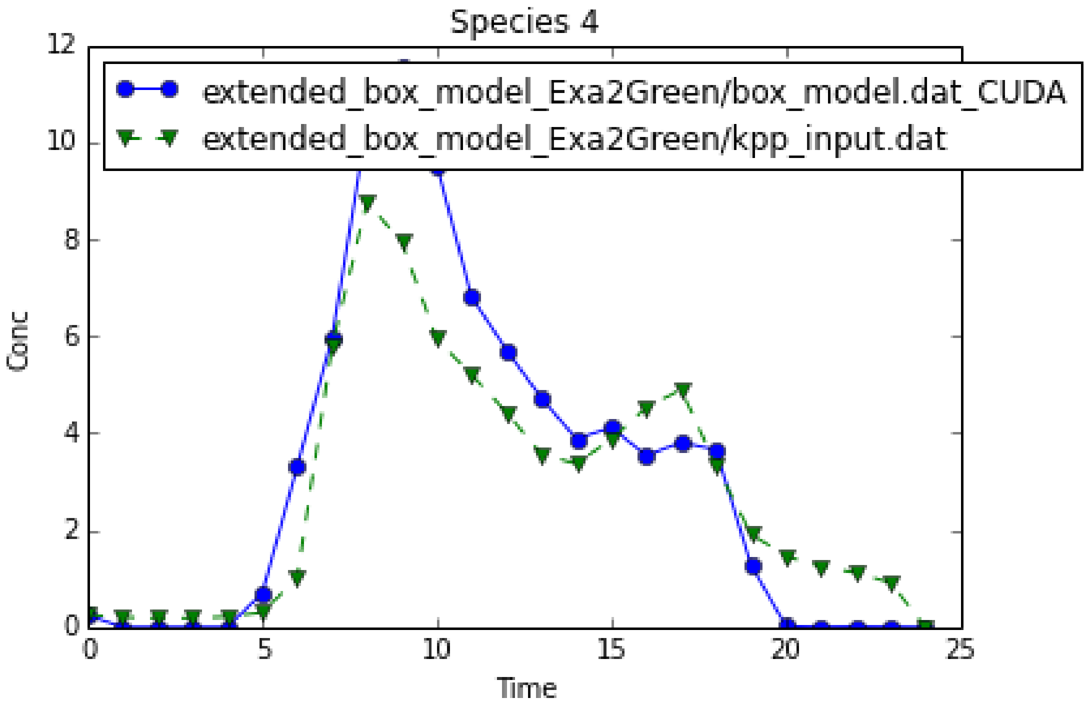
```
plot_dat([kpp_dat_1, kpp_dat_2], names=[kpp_file_1, kpp_file_2], titles=None)
```


Species 0
Conc
Time
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
5
10
15
20
25
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp_input.dat

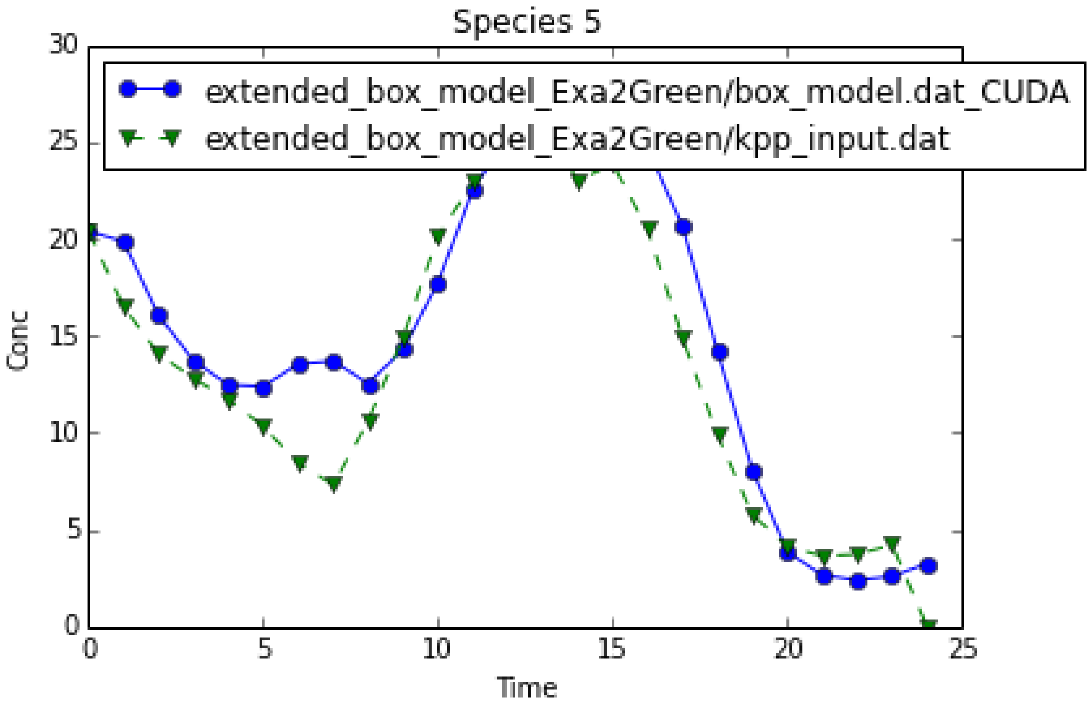

Species 1
Conc
Time
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20
25
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp_input.dat


Species 2

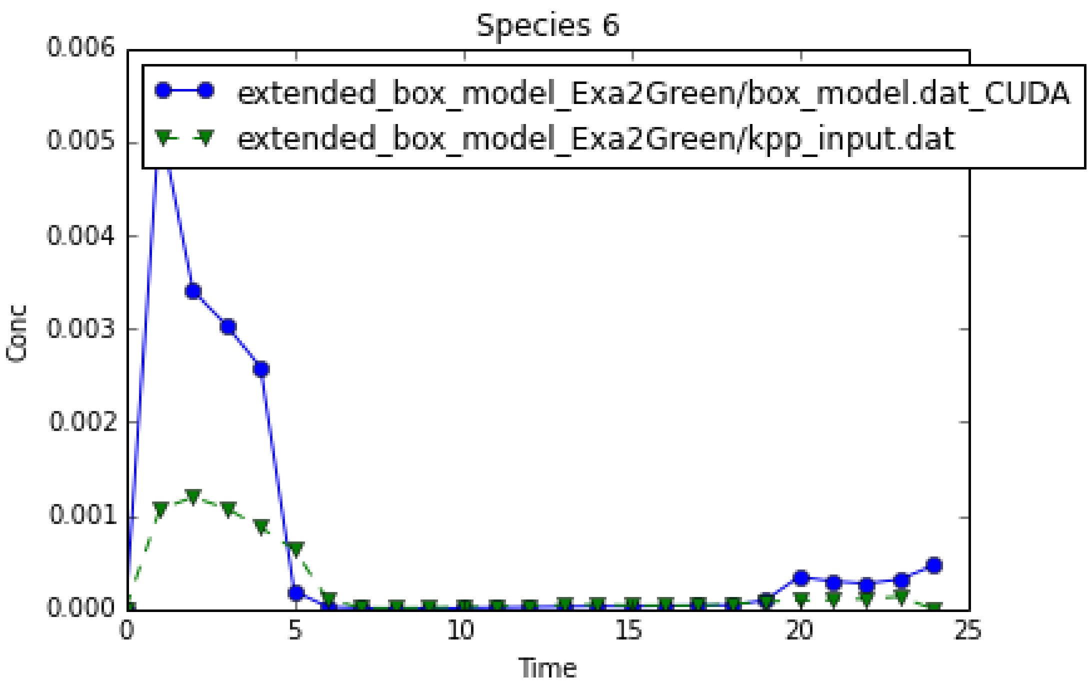

Species 3

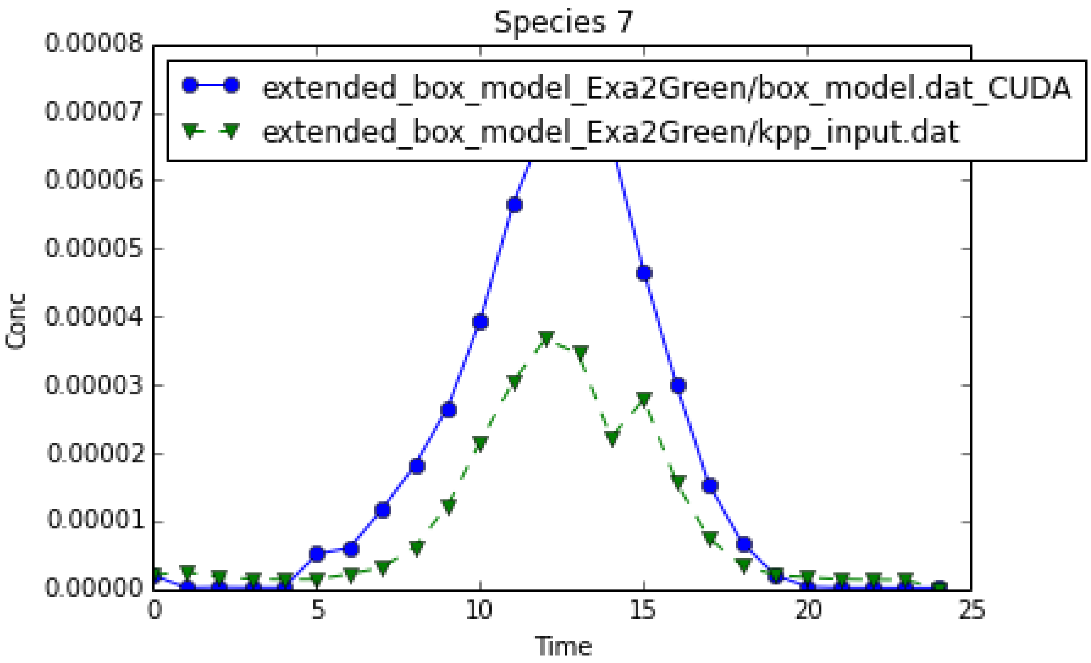


Species 4



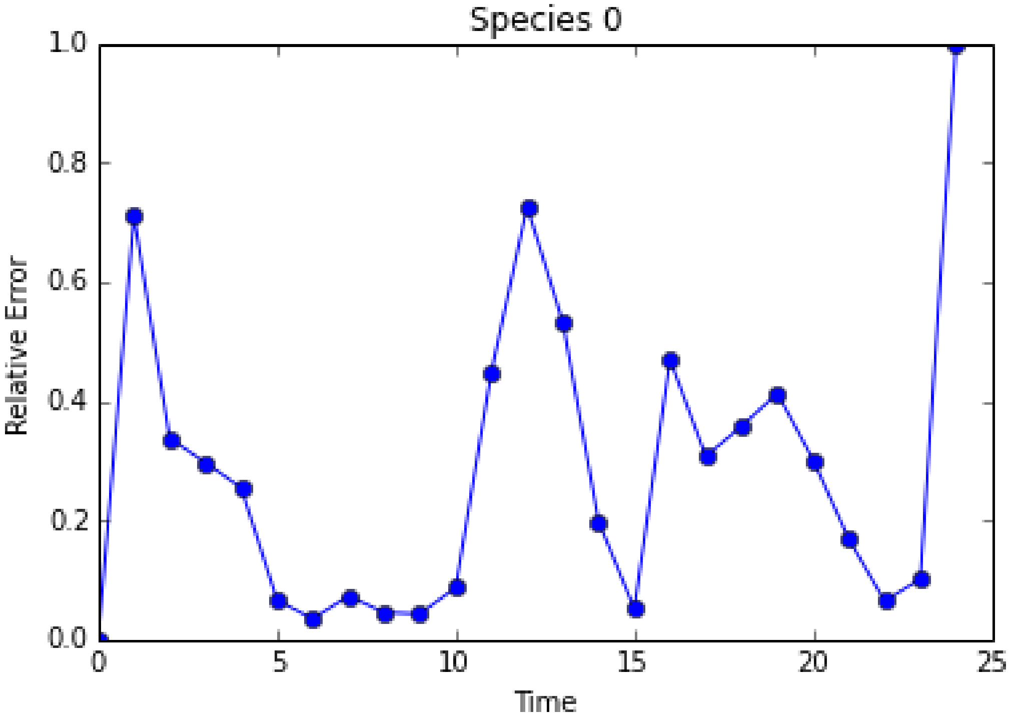
Species 5



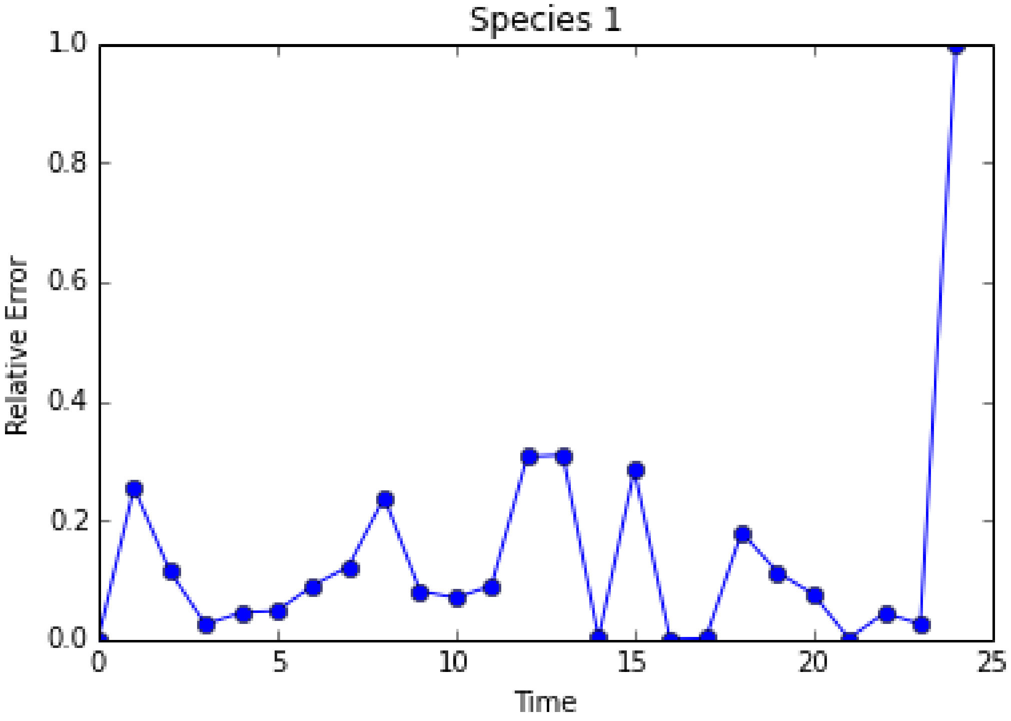


In [36]:

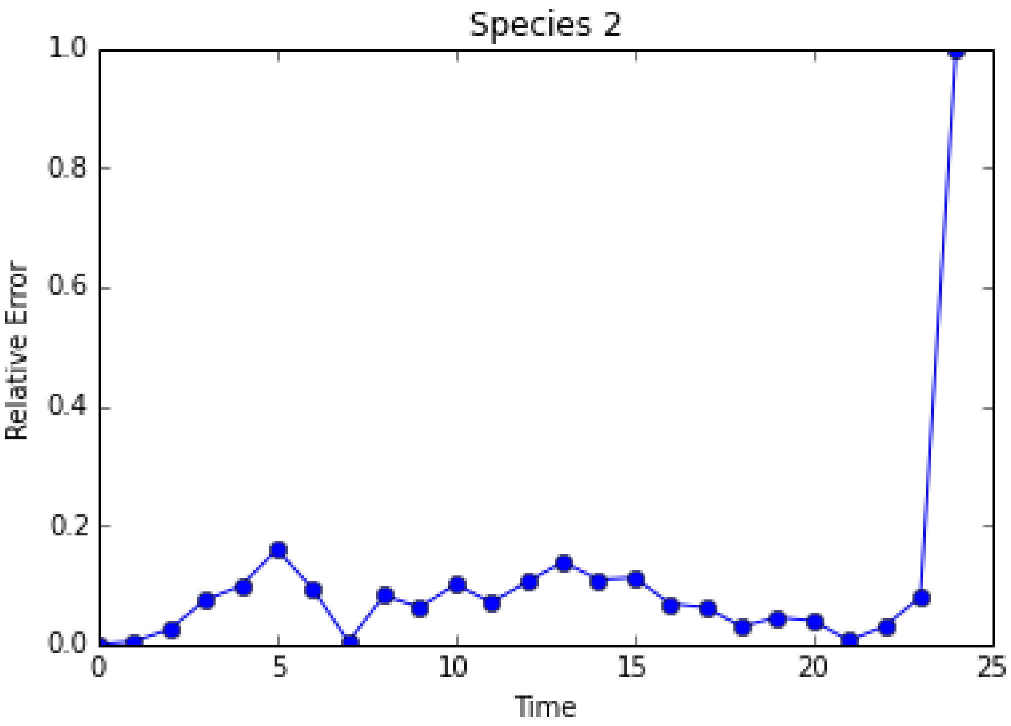
```
plot_dat([err_dat], ylabel='Relative Error')
```



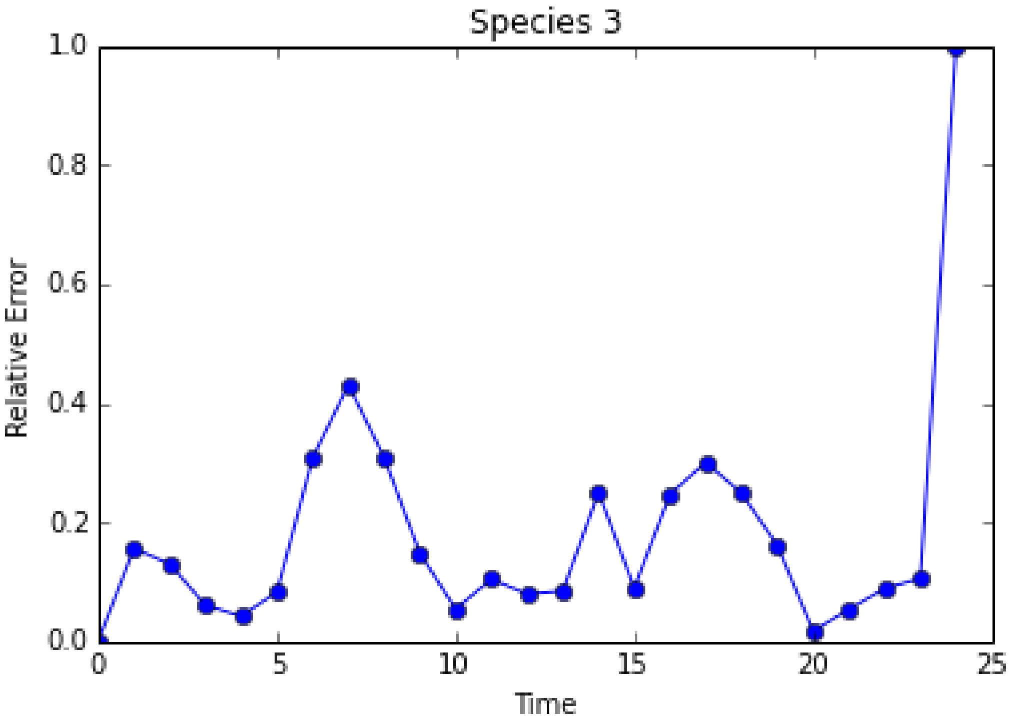
Species 0



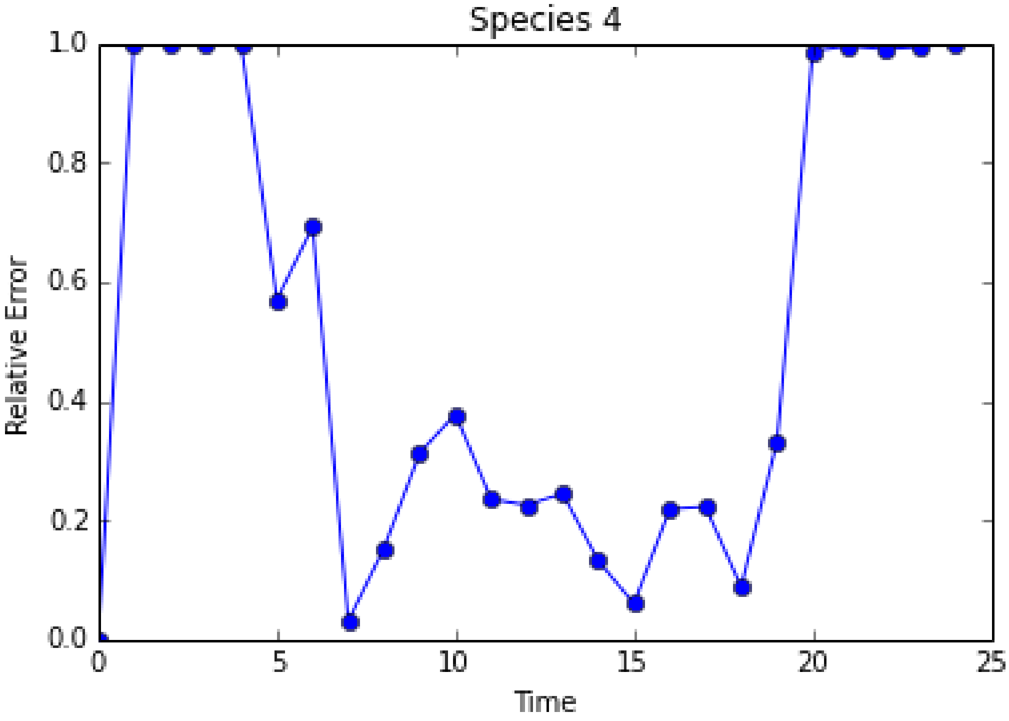
Species 1



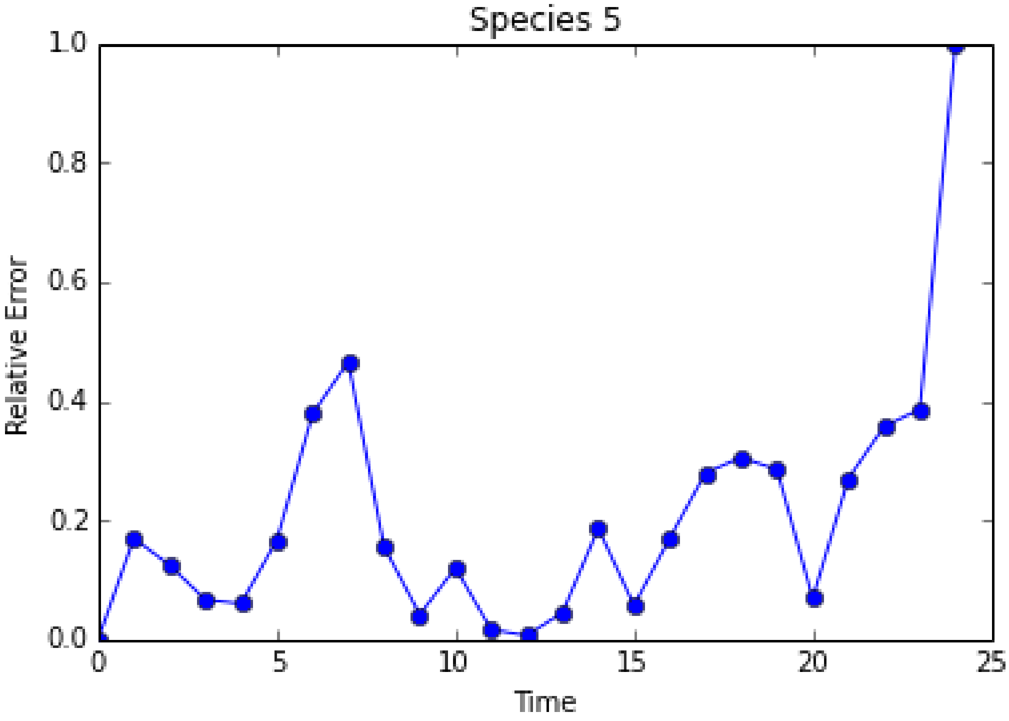
Species 2



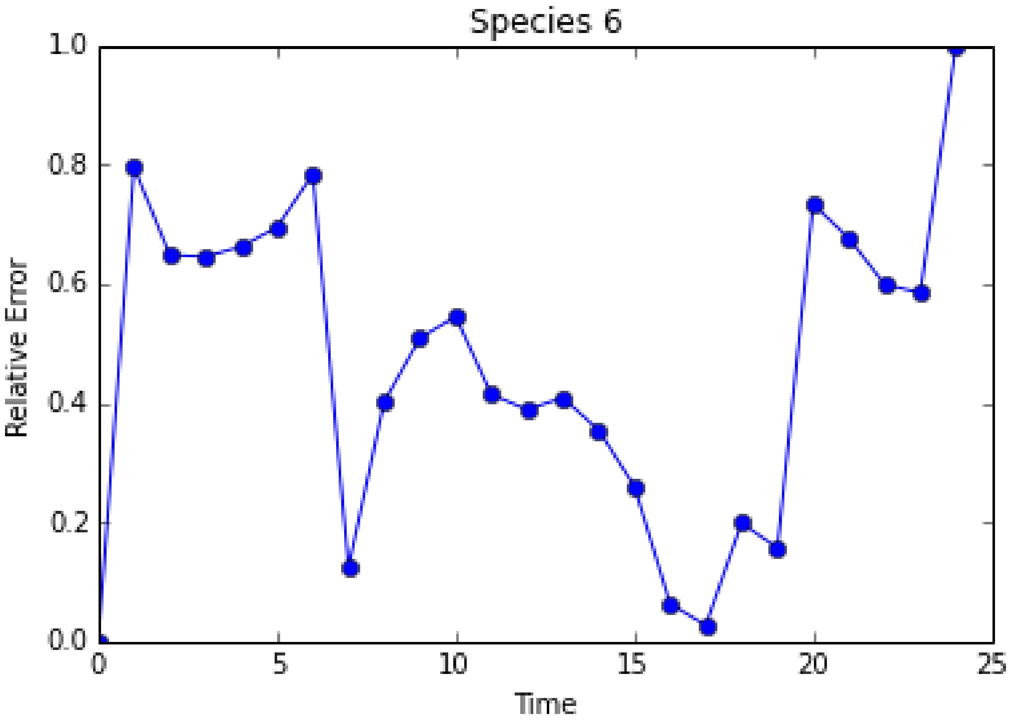
Species 3



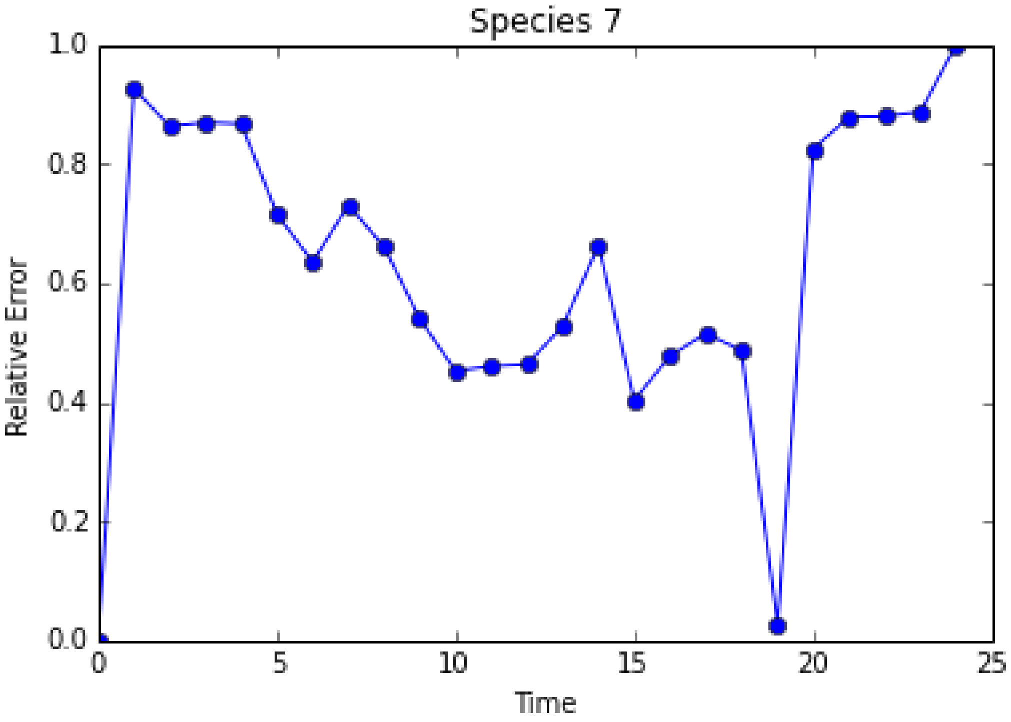
Species 4



Species 5



Species 6



Species 7

In [36]: