kpp_vs_kppa

Unknown Author

December 16, 2014

In [13]:
```
kpp_file_1 = 'extended_box_model_Exa2Green/box_model.dat_CUDA'
kpp_file_2 = 'extended_box_model_Exa2Green/kpp_input.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]:
```
kpp_dat_1 = read_datfile(kpp_file_1, 0, 1)
kpp_dat_2 = read_datfile(kpp_file_2, 0, 1)
```

In [16]:
```
err_dat = calc_err(kpp_dat_1, kpp_dat_2)
```

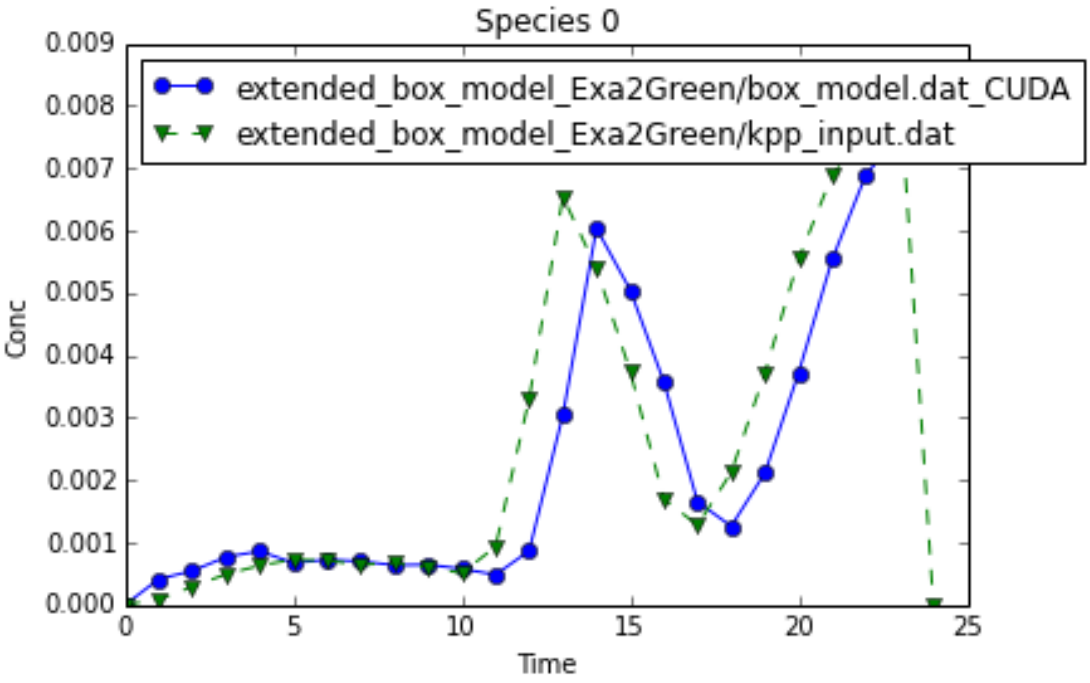
```
0.78462 > 0.01: 0.000414384, 8.925e-05
0.251904 > 0.01: 0.0309452, 0.02315
0.924272 > 0.01: 1.84398e-07, 2.435e-06
0.156043 > 0.01: 7.24621, 8.586
0.999908 > 0.01: 1.87544e-05, 0.2041
0.167376 > 0.01: 19.8529, 16.53
0.775839 > 0.01: 0.00521054, 0.001168
0.46455 > 0.01: 0.000545336, 0.000292
0.113167 > 0.01: 0.0234993, 0.02084
0.0306084 > 0.01: 0.639605, 0.6598
0.859691 > 0.01: 2.33335e-07, 1.663e-06
0.13344 > 0.01: 8.44116, 9.741
0.999948 > 0.01: 9.38301e-06, 0.1805
0.121624 > 0.01: 16.0979, 14.14
0.560522 > 0.01: 0.00352464, 0.001549
0.358343 > 0.01: 0.000768323, 0.000493
0.0213152 > 0.01: 0.0211304, 0.02068
0.0639184 > 0.01: 0.746623, 0.6989
0.874105 > 0.01: 1.6316e-07, 1.296e-06
0.0682622 > 0.01: 9.63417, 10.34
0.999939 > 0.01: 1.08944e-05, 0.1778
0.0621066 > 0.01: 13.7969, 12.94
0.508015 > 0.01: 0.00346555, 0.001705
0.268421 > 0.01: 0.000861698, 0.0006304
0.0405954 > 0.01: 0.0208984, 0.02005
0.0872357 > 0.01: 0.787279, 0.7186
0.88402 > 0.01: 1.32101e-07, 1.139e-06
0.0516525 > 0.01: 10.2516, 10.81
0.998978 > 0.01: 0.000204046, 0.1997
0.0542857 > 0.01: 12.6148, 11.93
0.508139 > 0.01: 0.00333428, 0.00164
0.0765432 > 0.01: 0.000672369, 0.0007281
0.0410053 > 0.01: 0.0202504, 0.01942
0.152762 > 0.01: 0.865401, 0.7332
0.733508 > 0.01: 4.13896e-06, 1.103e-06
0.0798084 > 0.01: 10.6466, 11.57
0.602491 > 0.01: 0.698853, 0.2778
0.166128 > 0.01: 12.6878, 10.58
0.826484 > 0.01: 0.000240147, 0.001384
0.0251677 > 0.01: 0.00072515, 0.0007069
0.0772728 > 0.01: 0.0196483, 0.01813
0.0807606 > 0.01: 0.760411, 0.699
```

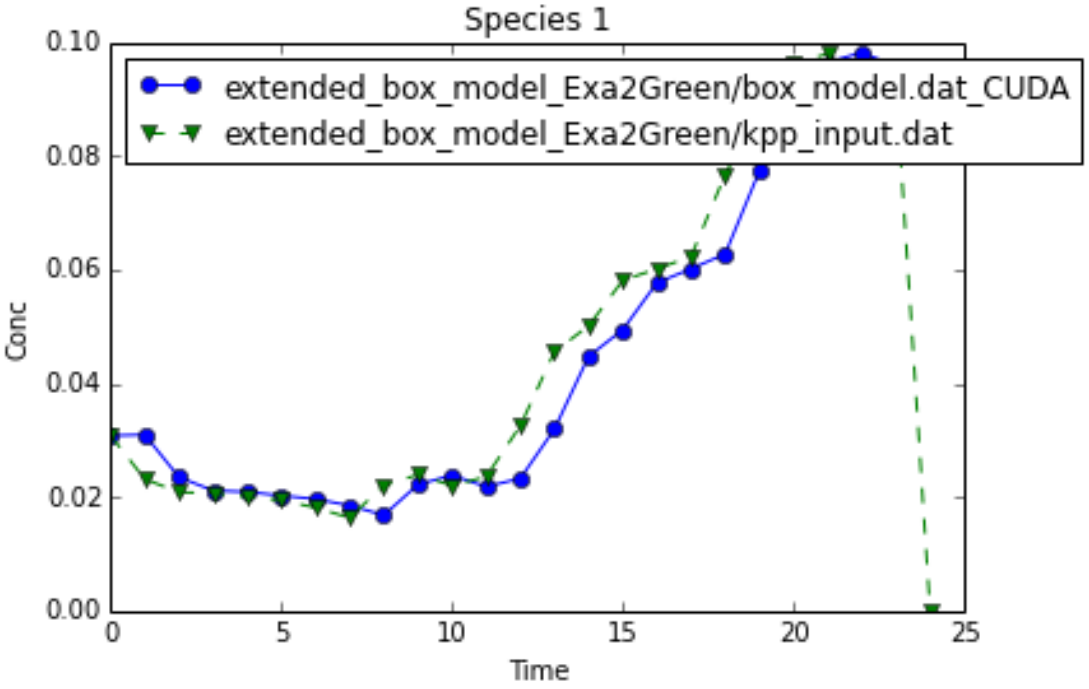
0.597033 > 0.01: 4.89123e-06, 1.971e-06
0.302848 > 0.01: 8.79109, 12.61
0.704345 > 0.01: 3.38232, 1
0.36806 > 0.01: 13.9966, 8.845
0.830099 > 0.01: 2.43978e-05, 0.0001436
0.0723807 > 0.01: 0.000699533, 0.0006489
0.116275 > 0.01: 0.0185012, 0.01635
0.713069 > 0.01: 1.04764e-05, 3.006e-06
0.419068 > 0.01: 7.43593, 12.8
0.0658542 > 0.01: 6.10611, 5.704
0.458897 > 0.01: 14.2598, 7.716
0.162638 > 0.01: 1.15137e-05, 1.375e-05
0.041158 > 0.01: 0.000636767, 0.0006641
0.237003 > 0.01: 0.0168088, 0.02203
0.0826661 > 0.01: 0.763314, 0.8321
0.658274 > 0.01: 1.7356e-05, 5.931e-06
0.304476 > 0.01: 7.83856, 11.27
0.160986 > 0.01: 10.4682, 8.783
0.157722 > 0.01: 13.0836, 11.02
0.403991 > 0.01: 6.70511e-06, 1.125e-05
0.0651836 > 0.01: 0.000645367, 0.0006033
0.0735048 > 0.01: 0.022171, 0.02393
0.0605586 > 0.01: 0.862214, 0.81
0.535682 > 0.01: 2.54567e-05, 1.182e-05
0.143508 > 0.01: 8.07586, 9.429
0.312434 > 0.01: 11.6978, 8.043
0.0290666 > 0.01: 14.8164, 15.26
0.500012 > 0.01: 7.39983e-06, 1.48e-05
0.115175 > 0.01: 0.000578645, 0.000512
0.077044 > 0.01: 0.0238365, 0.022
0.0987563 > 0.01: 0.840061, 0.7571
0.450516 > 0.01: 3.80539e-05, 2.091e-05
0.0480823 > 0.01: 7.38974, 7.763
0.375171 > 0.01: 9.70986, 6.067
0.109545 > 0.01: 18.1029, 20.33
0.535229 > 0.01: 1.04713e-05, 2.253e-05
0.47979 > 0.01: 0.000483743, 0.0009299
0.074907 > 0.01: 0.0218044, 0.02357
0.0698137 > 0.01: 0.798872, 0.7431
0.454598 > 0.01: 5.42902e-05, 2.961e-05
0.0983932 > 0.01: 6.38247, 7.079
0.238134 > 0.01: 6.97367, 5.313
0.0163741 > 0.01: 22.7119, 23.09
0.412109 > 0.01: 1.62493e-05, 2.764e-05
0.737856 > 0.01: 0.000865863, 0.003303
0.2848 > 0.01: 0.0232655, 0.03253
0.104727 > 0.01: 0.807463, 0.7229
0.463989 > 0.01: 6.66963e-05, 3.575e-05
0.0779528 > 0.01: 5.97026, 6.475
0.226688 > 0.01: 5.85016, 4.524
0.387751 > 0.01: 2.08226e-05, 3.401e-05
0.535149 > 0.01: 0.00304152, 0.006543
0.298706 > 0.01: 0.0319509, 0.04556

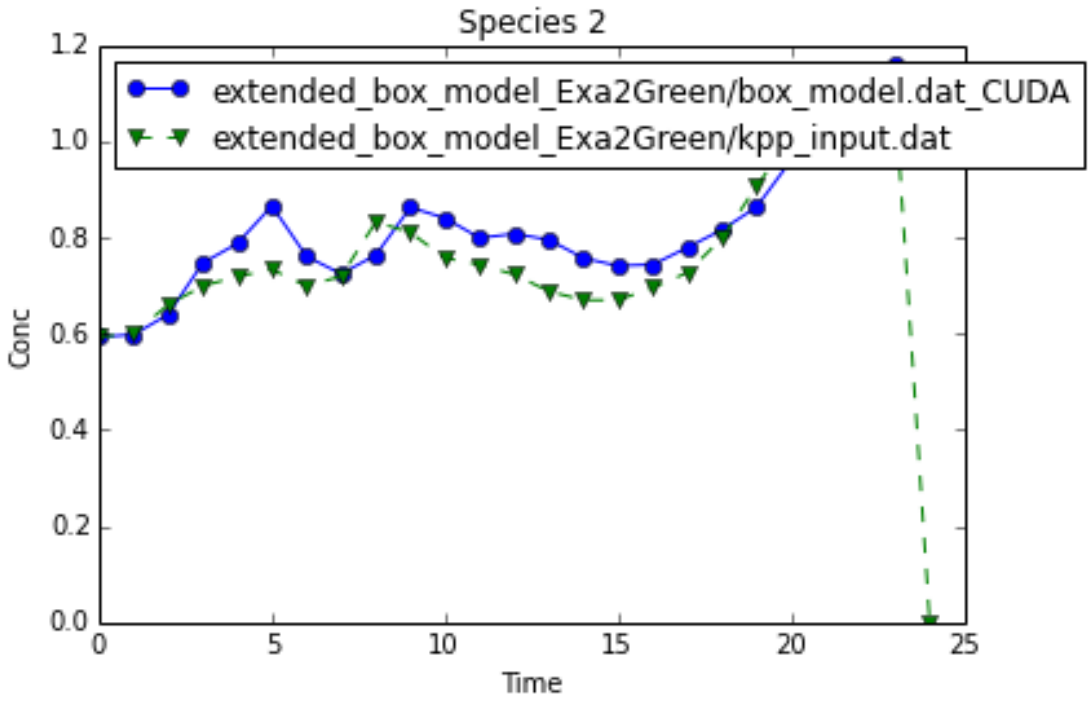
0.135064 > 0.01: 0.795088, 0.6877
0.519629 > 0.01: 7.04872e-05, 3.386e-05
0.0809308 > 0.01: 5.56772, 6.058
0.244305 > 0.01: 4.83263, 3.652
0.0431248 > 0.01: 27.4644, 26.28
0.405087 > 0.01: 2.58192e-05, 4.34e-05
0.106475 > 0.01: 0.00603565, 0.005393
0.10712 > 0.01: 0.0447244, 0.05009
0.115139 > 0.01: 0.756842, 0.6697
0.500139 > 0.01: 6.43778e-05, 3.218e-05
0.123152 > 0.01: 5.18831, 5.917
0.0731824 > 0.01: 3.98352, 3.692
0.119599 > 0.01: 28.3507, 24.96
0.245785 > 0.01: 3.02893e-05, 4.016e-05
0.258383 > 0.01: 0.00504843, 0.003744
0.149537 > 0.01: 0.0494119, 0.0581
0.0962153 > 0.01: 0.740884, 0.6696
0.524658 > 0.01: 4.75237e-05, 2.259e-05
0.189613 > 0.01: 5.37287, 6.63
0.0143381 > 0.01: 3.78832, 3.734
0.130795 > 0.01: 26.53, 23.06
0.253714 > 0.01: 3.11201e-05, 4.17e-05
0.525651 > 0.01: 0.00357542, 0.001696
0.0407418 > 0.01: 0.057661, 0.06011
0.0643259 > 0.01: 0.743314, 0.6955
0.465456 > 0.01: 2.95766e-05, 1.581e-05
0.224023 > 0.01: 6.25671, 8.063
0.21232 > 0.01: 3.75015, 4.761
0.162079 > 0.01: 24.2266, 20.3
0.0395386 > 0.01: 3.35009e-05, 3.488e-05
0.226345 > 0.01: 0.0016519, 0.001278
0.0346738 > 0.01: 0.0601205, 0.06228
0.0670588 > 0.01: 0.778934, 0.7267
0.645254 > 0.01: 1.48191e-05, 5.257e-06
0.355625 > 0.01: 8.60885, 13.36
0.0394233 > 0.01: 3.93097, 3.776
0.33361 > 0.01: 20.3184, 13.54
0.268169 > 0.01: 3.71477e-05, 5.076e-05
0.406044 > 0.01: 0.00126156, 0.002124
0.182415 > 0.01: 0.0627414, 0.07674
0.0215912 > 0.01: 0.815917, 0.7983
0.491318 > 0.01: 6.66821e-06, 3.392e-06
0.249709 > 0.01: 13.1826, 17.57
0.117916 > 0.01: 3.72981, 3.29
0.318238 > 0.01: 14.0665, 9.59
0.226669 > 0.01: 4.11412e-05, 5.32e-05
0.431641 > 0.01: 0.00211373, 0.003719
0.118767 > 0.01: 0.0773017, 0.08772
0.0476907 > 0.01: 0.862983, 0.9062
0.0155273 > 0.01: 2.10773e-06, 2.075e-06
0.15569 > 0.01: 19.3853, 22.96
0.31241 > 0.01: 1.32223, 1.923
0.279398 > 0.01: 7.86842, 5.67

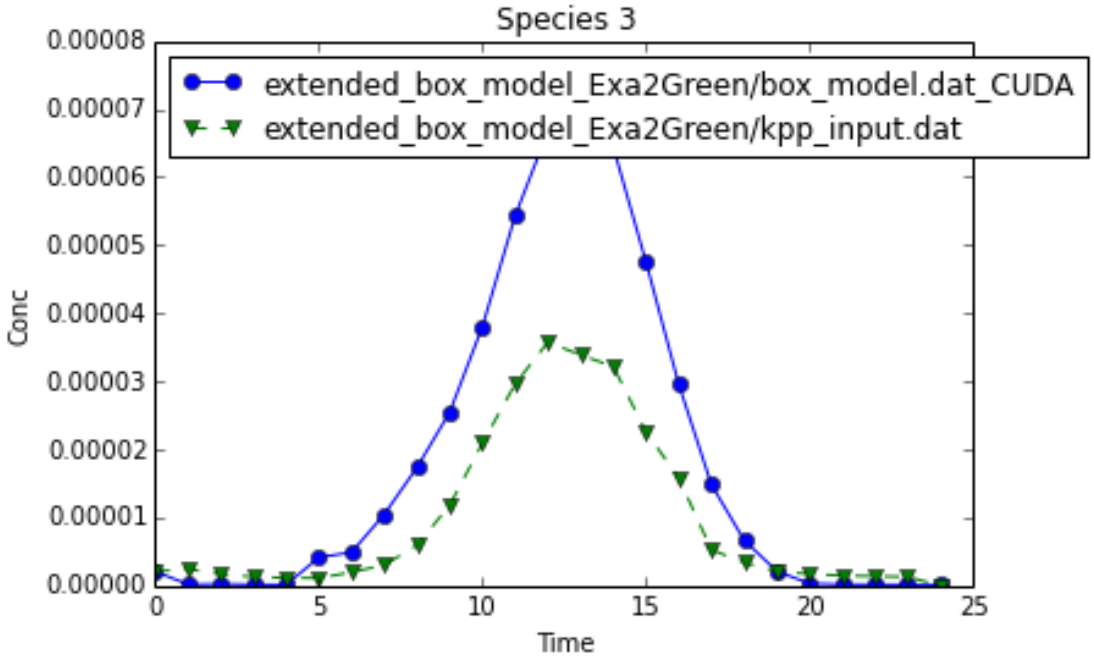
```
0.102168 > 0.01: 8.85355e-05, 7.949e-05
0.328589 > 0.01: 0.00371828, 0.005538
0.0843906 > 0.01: 0.0880816, 0.0962
0.0456978 > 0.01: 0.960982, 1.007
0.820292 > 0.01: 3.07301e-07, 1.71e-06
0.0222322 > 0.01: 24.7277, 25.29
0.985069 > 0.01: 0.0220535, 1.477
0.0773878 > 0.01: 3.75688, 4.072
0.706636 > 0.01: 0.000332522, 9.755e-05
0.196334 > 0.01: 0.00554048, 0.006894
0.0153221 > 0.01: 0.0964394, 0.09794
0.0132279 > 0.01: 1.06177, 1.076
0.877093 > 0.01: 1.80797e-07, 1.471e-06
0.0510152 > 0.01: 26.6495, 25.29
0.993357 > 0.01: 0.00822403, 1.238
0.301784 > 0.01: 2.58061, 3.696
0.593915 > 0.01: 0.000287378, 0.0001167
0.0817845 > 0.01: 0.0068958, 0.00751
0.0259813 > 0.01: 0.0981501, 0.0956
0.0221415 > 0.01: 1.13207, 1.107
0.882982 > 0.01: 1.61602e-07, 1.381e-06
0.0933031 > 0.01: 26.4366, 23.97
0.990966 > 0.01: 0.00925128, 1.024
0.399562 > 0.01: 2.45399, 4.087
0.390048 > 0.01: 0.000293793, 0.0001792
0.0991535 > 0.01: 0.00752027, 0.008348
0.034892 > 0.01: 0.0958027, 0.09246
0.0584681 > 0.01: 1.15981, 1.092
0.888492 > 0.01: 1.44068e-07, 1.292e-06
0.0881504 > 0.01: 24.8835, 22.69
0.99912 > 0.01: 0.000717783, 0.8158
0.344037 > 0.01: 3.02465, 4.611
0.394157 > 0.01: 0.000451932, 0.0002738
1 > 0.01: 0.00837719, 0
1 > 0.01: 0.092658, 0
1 > 0.01: 1.15049, 0
1 > 0.01: 1.39532e-07, 0
1 > 0.01: 23.3724, 0
1 > 0.01: 2.17898e-05, 0
1 > 0.01: 3.73502, 0
1 > 0.01: 0.000672572, 0
SNR: 22.491445db
189 samples with relative error > 0.01
```

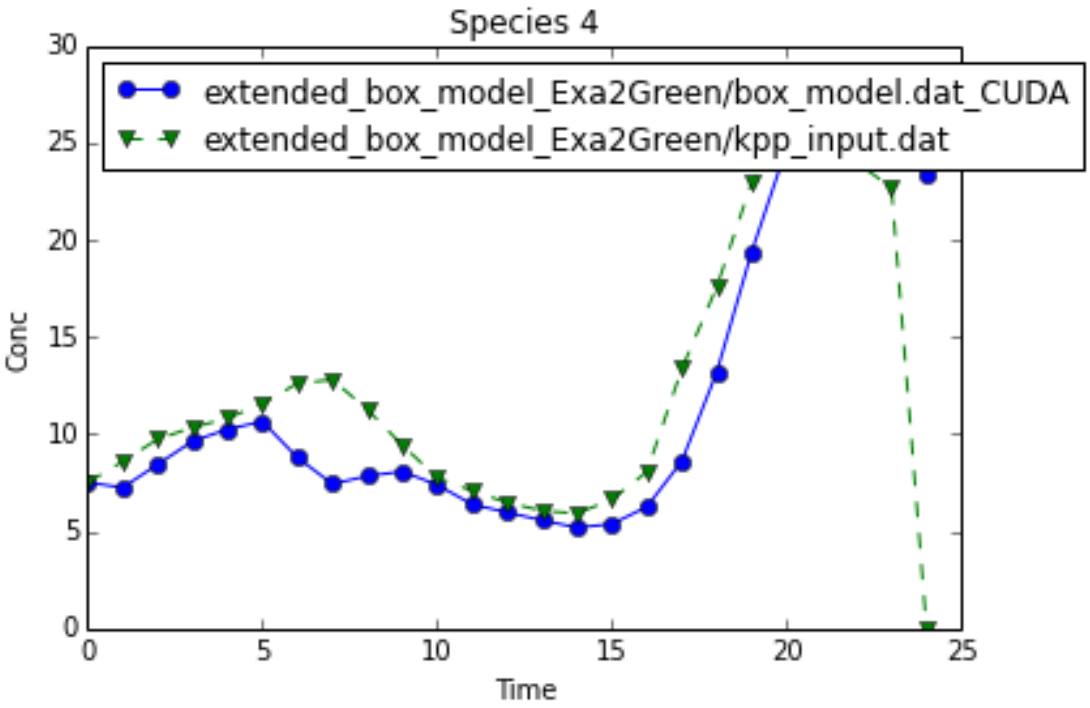
In [17]:
```
plot_dat([kpp_dat_1, kpp_dat_2], names=[kpp_file_1, kpp_file_2], titles=None)
```

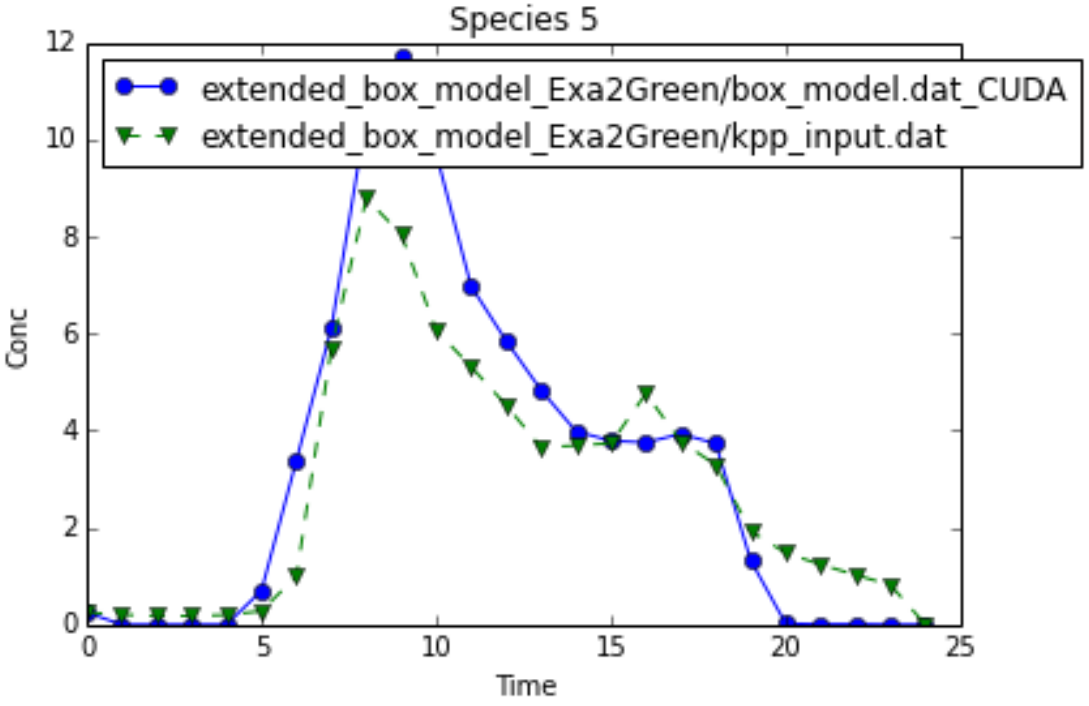

Species 0
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp_input.dat
Conc
Time
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
5
10
15
20
25

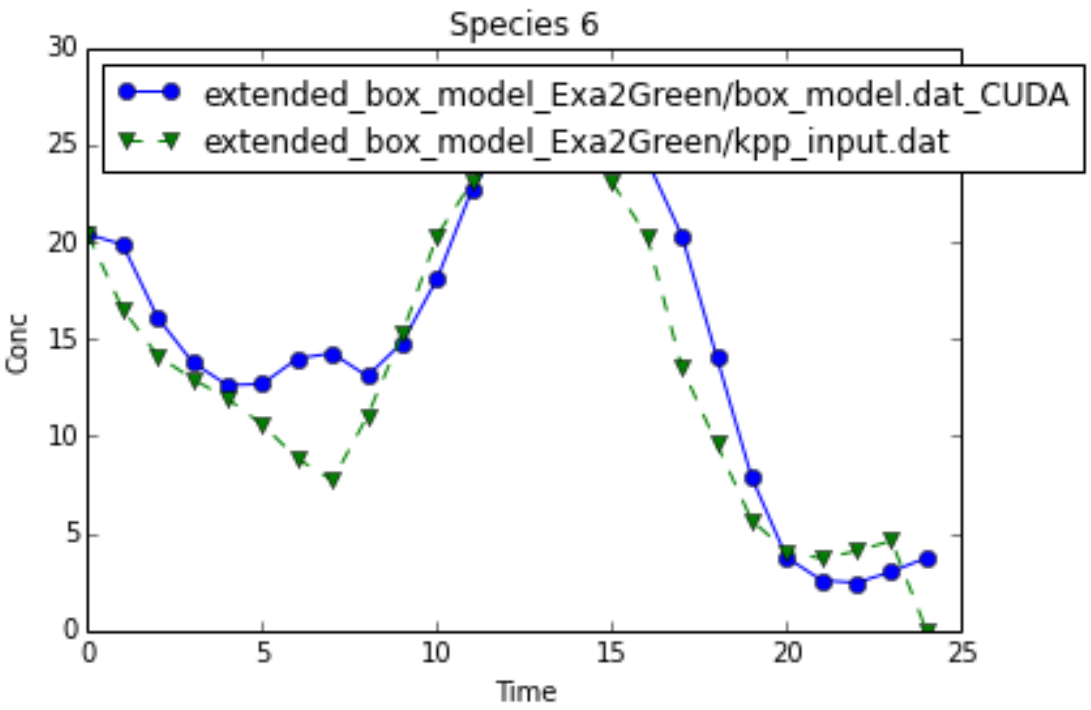

Species 1
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp_input.dat
Conc
Time
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20
25

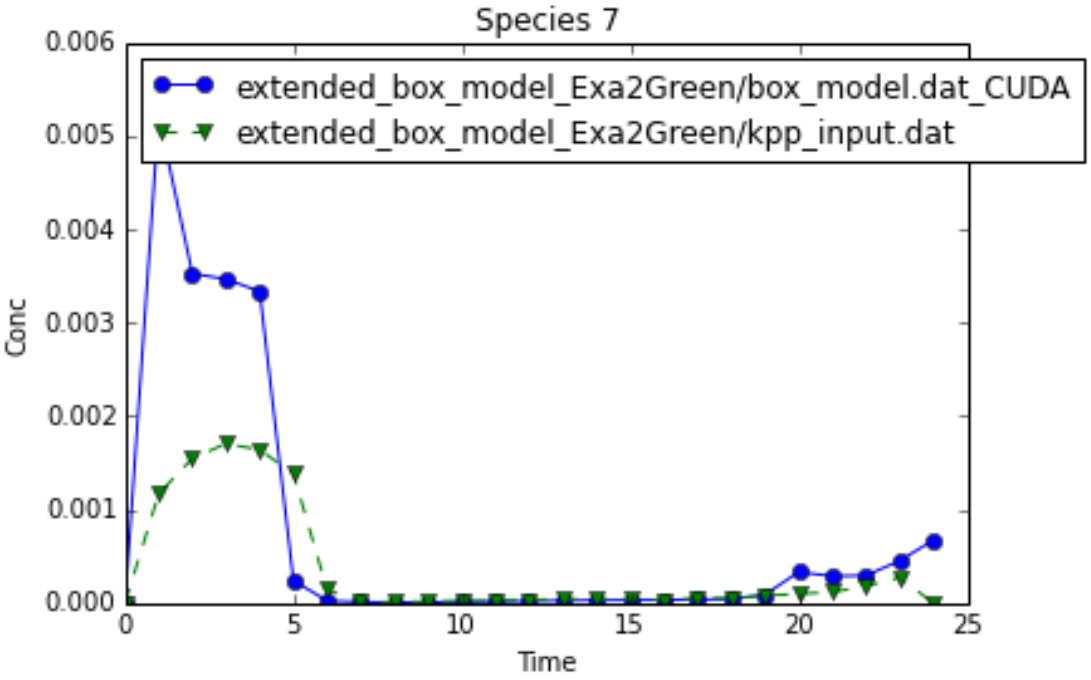

Species 2
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp_input.dat
Conc
Time
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20
25


Species 3
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp_input.dat
Conc
Time
0.00000
0.00001
0.00002
0.00003
0.00004
0.00005
0.00006
0.00007
0.00008
0
5
10
15
20
25

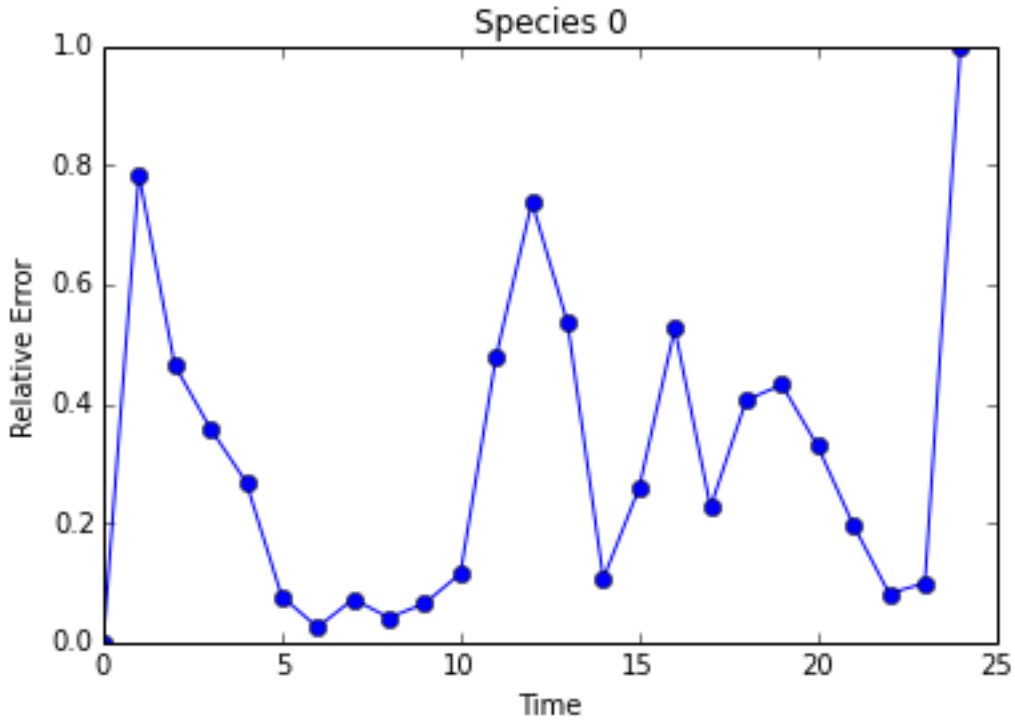

Species 4
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp_input.dat
Conc
Time
0
5
10
15
20
25
30

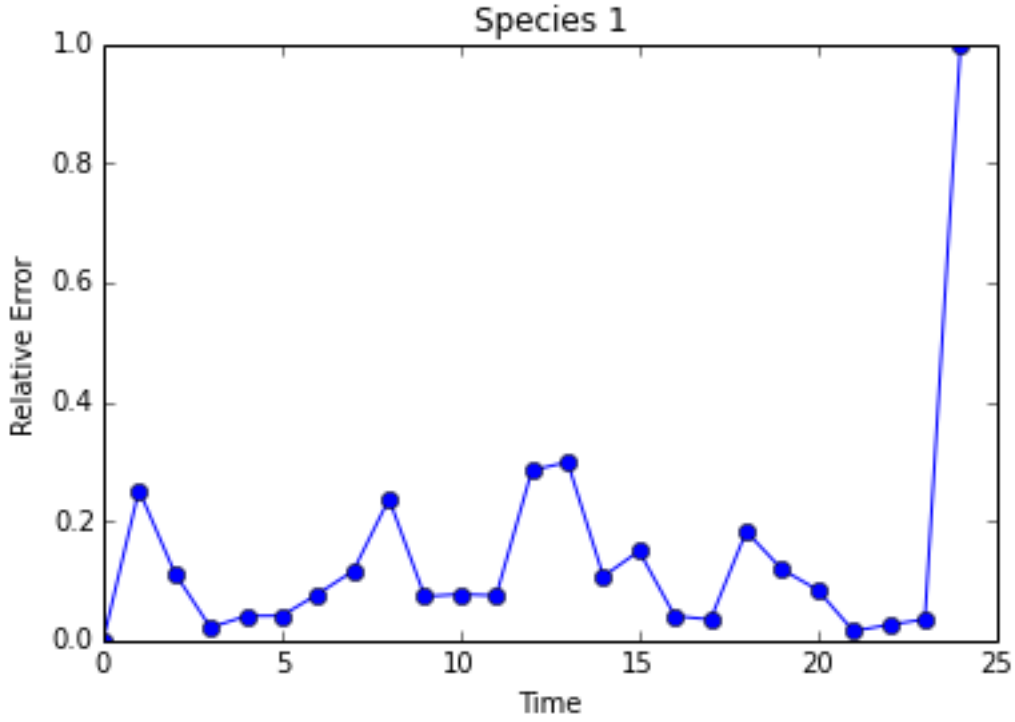

Species 5
extended_box_model_Exa2Green/box_model.dat_CUDA
extended_box_model_Exa2Green/kpp_input.dat
Conc
Time
0
2
4
6
8
10
12
0
5
10
15
20
25

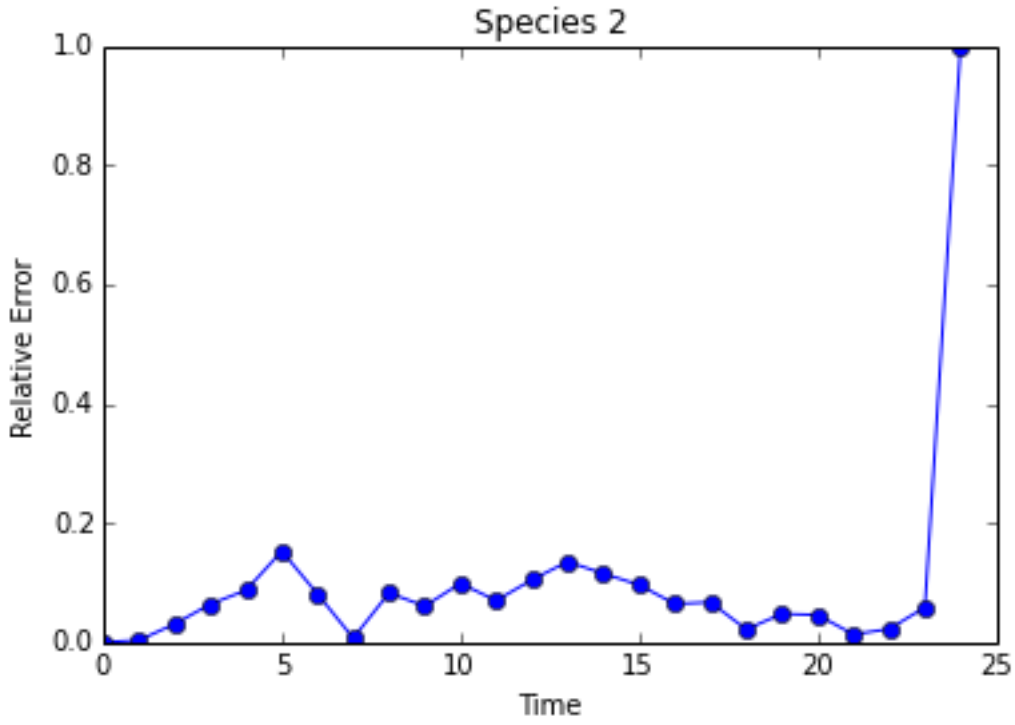


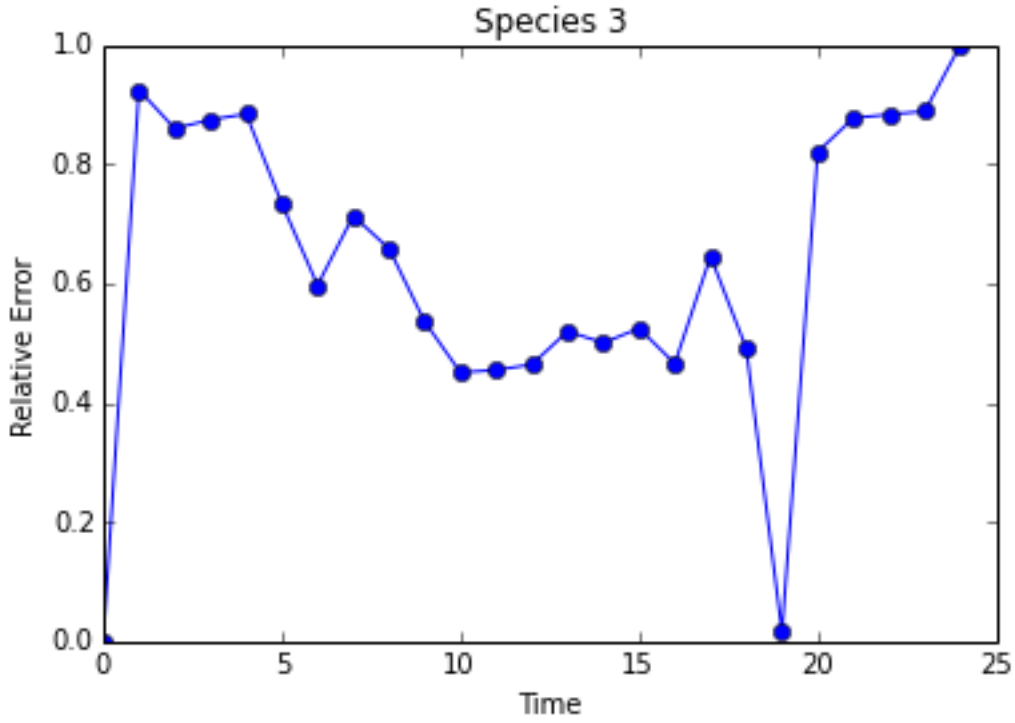


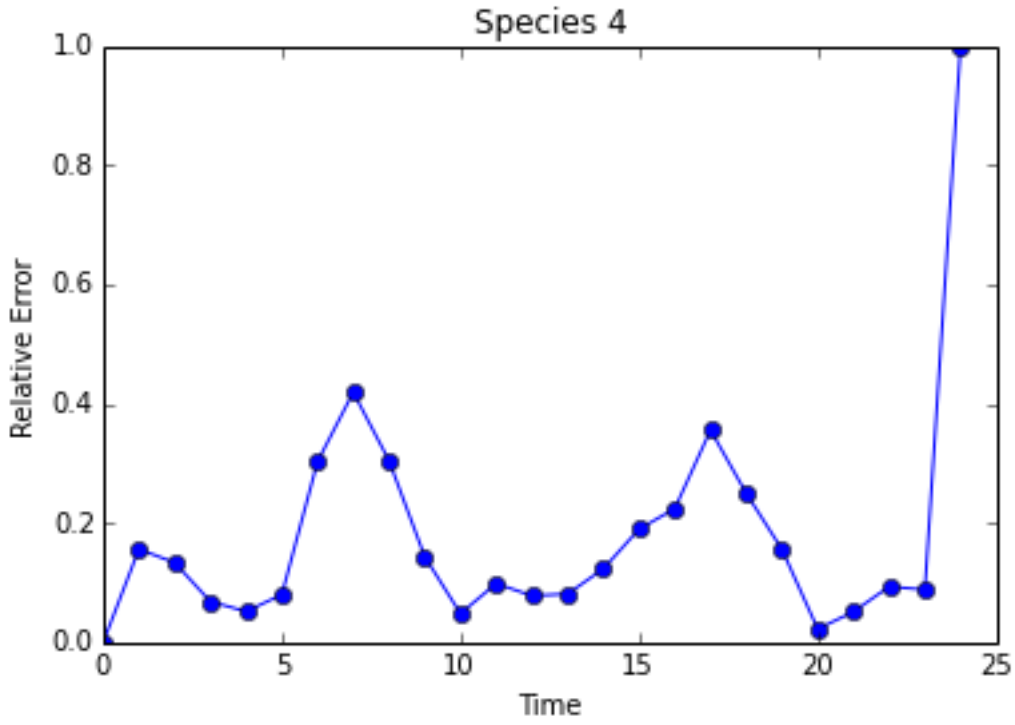
In [18]:
```
plot_dat([err_dat], ylabel='Relative Error')
```

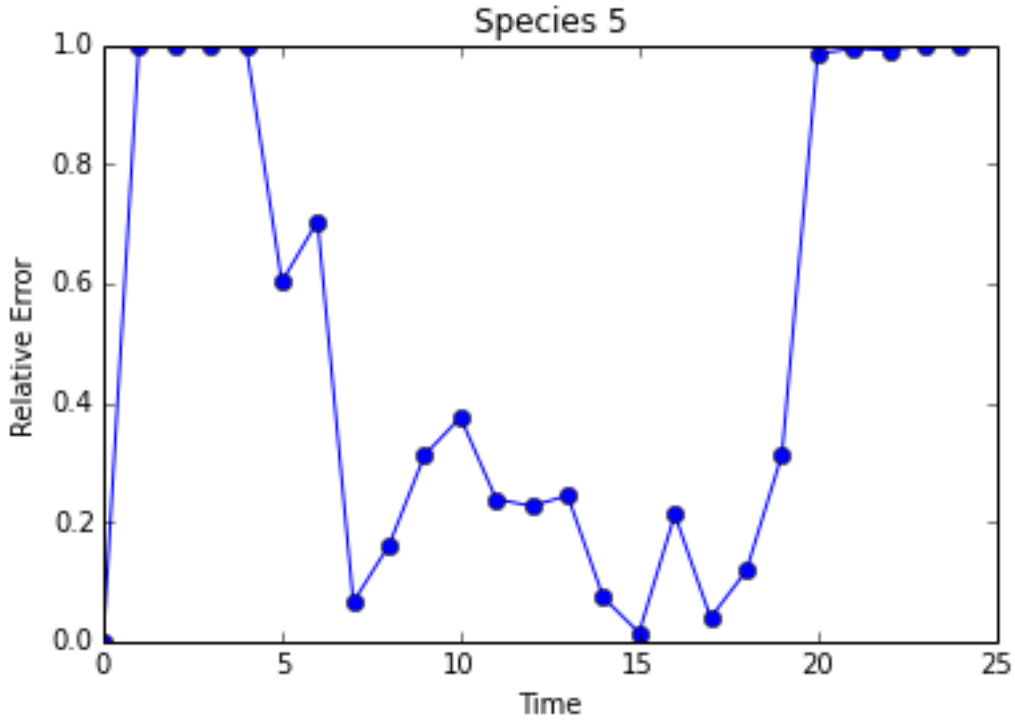

Species 0
Relative Error
Time
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25

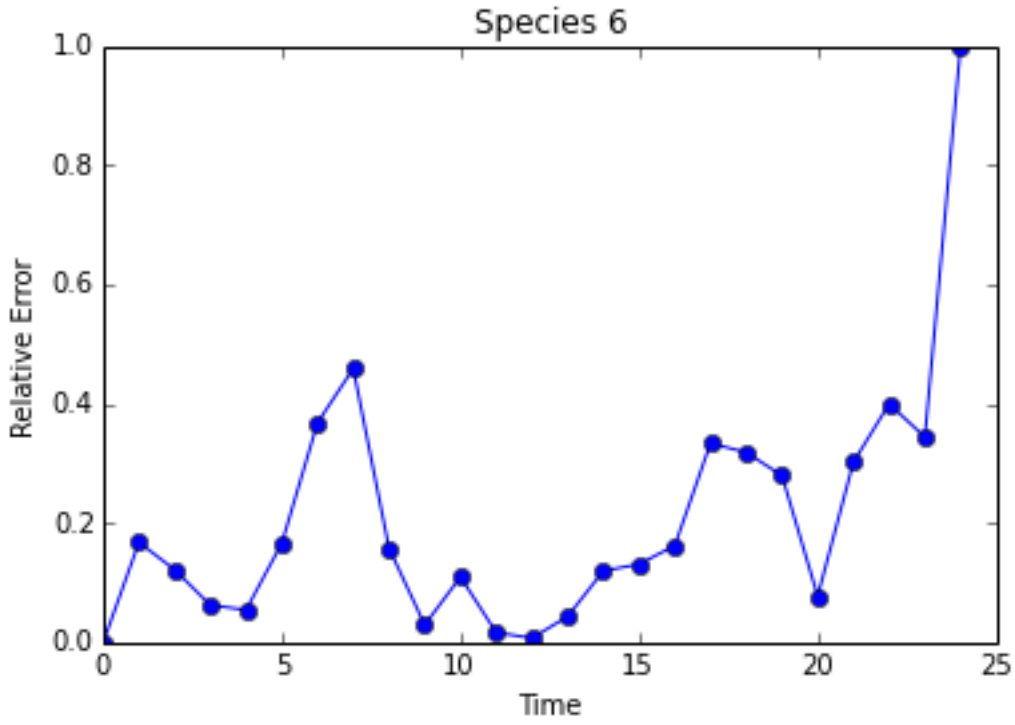

Species 1
Relative Error
Time
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25

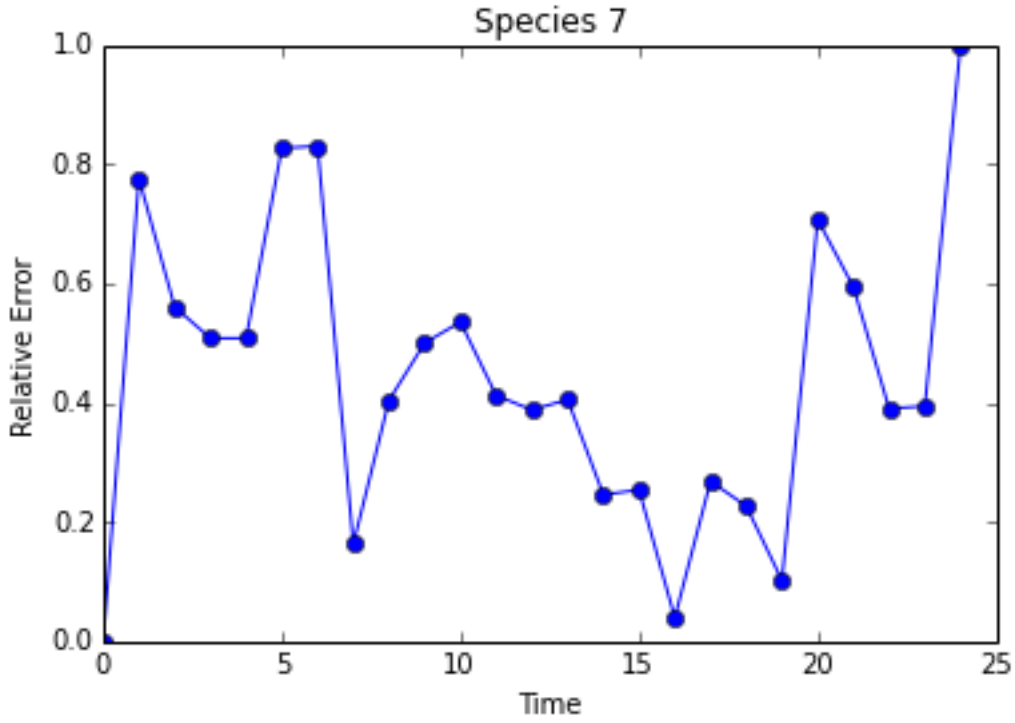

Species 2
Relative Error
Time
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25


Species 3
Relative Error
Time
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25


Species 4
Relative Error
Time
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25


Species 5
Relative Error
Time
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25


Species 6
Relative Error
Time
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25


Species 7
Relative Error
Time
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25

In [18]: