kpp_vs_kppa

Unknown Author

May 27, 2015

In [7]:

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

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()
```

```python
            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)
```

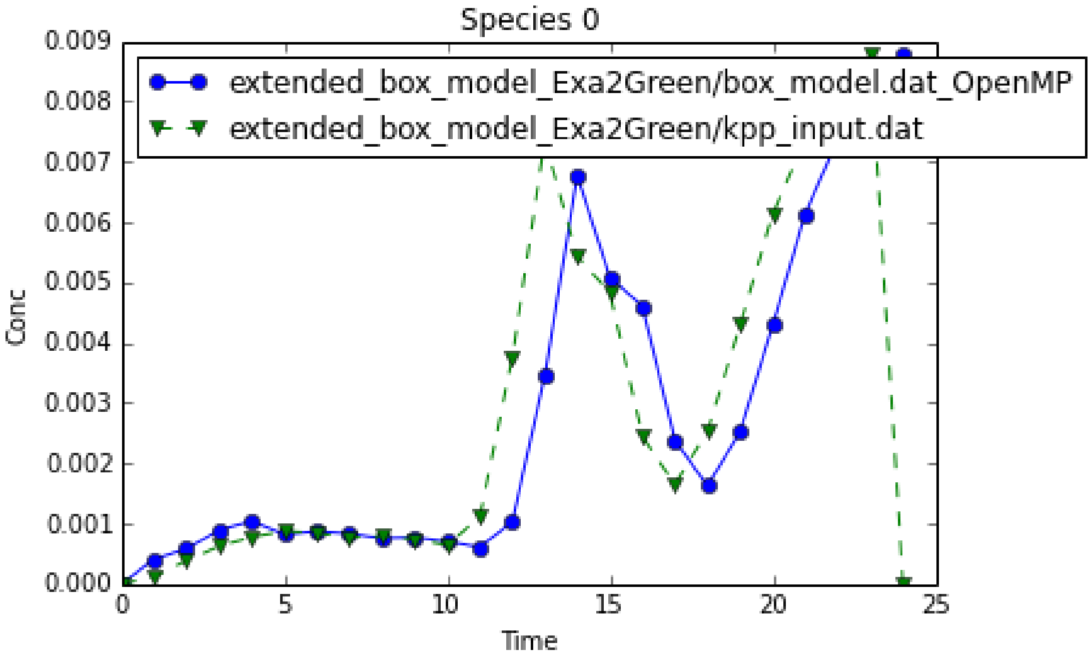
```
0.710416 > 0.01: 0.000414387, 0.00012
0.25449 > 0.01: 0.0309453, 0.02307
0.156141 > 0.01: 7.24622, 8.587
0.999909 > 0.01: 1.87488e-05, 0.2059
0.16939 > 0.01: 19.8529, 16.49
0.795186 > 0.01: 0.00520961, 0.001067
0.926456 > 0.01: 1.84229e-07, 2.505e-06
0.337172 > 0.01: 0.00058537, 0.000388
0.115126 > 0.01: 0.0234384, 0.02074
0.0251421 > 0.01: 0.640384, 0.6569
0.129998 > 0.01: 8.44163, 9.703
0.999951 > 0.01: 9.05375e-06, 0.1853
0.124496 > 0.01: 16.0593, 14.06
0.64825 > 0.01: 0.00340014, 0.001196
0.86348 > 0.01: 2.45873e-07, 1.801e-06
0.296737 > 0.01: 0.000890847, 0.0006265
0.0250619 > 0.01: 0.021068, 0.02054
0.0746165 > 0.01: 0.74434, 0.6888
0.0608055 > 0.01: 9.59857, 10.22
0.999949 > 0.01: 9.60519e-06, 0.1867
0.066636 > 0.01: 13.7138, 12.8
0.644931 > 0.01: 0.00303321, 0.001077
0.869878 > 0.01: 1.97525e-07, 1.518e-06
0.25442 > 0.01: 0.00104067, 0.0007759
0.0447757 > 0.01: 0.0208328, 0.0199
0.0982584 > 0.01: 0.777606, 0.7012
0.0421648 > 0.01: 10.1435, 10.59
0.999048 > 0.01: 0.000202853, 0.213
0.0597921 > 0.01: 12.4653, 11.72
0.662019 > 0.01: 0.00257855, 0.0008715
0.867551 > 0.01: 1.90594e-07, 1.439e-06
0.0654276 > 0.01: 0.000820368, 0.0008778
0.0477377 > 0.01: 0.0202045, 0.01924
0.159637 > 0.01: 0.842136, 0.7077
0.0822183 > 0.01: 10.2792, 11.2
0.562118 > 0.01: 0.680091, 0.2978
0.166014 > 0.01: 12.3743, 10.32
0.68549 > 0.01: 0.000199368, 0.0006339
0.716534 > 0.01: 5.16817e-06, 1.465e-06
0.0344548 > 0.01: 0.0008736, 0.0008435
0.0900143 > 0.01: 0.0195498, 0.01779
0.0933375 > 0.01: 0.736327, 0.6676
```

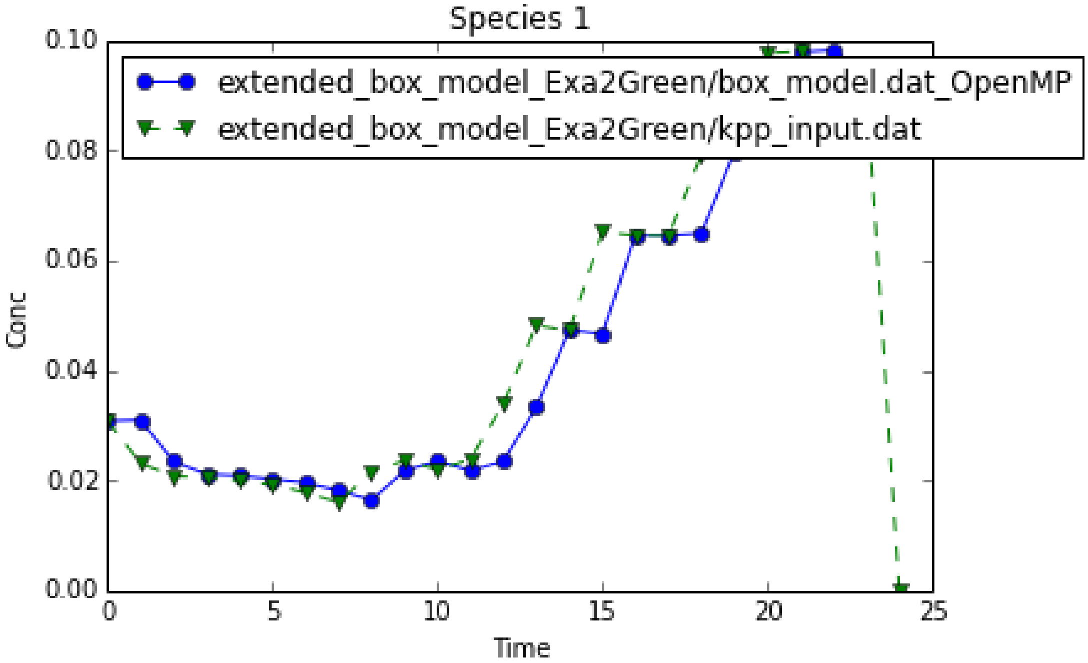
```
0.307325 > 0.01: 8.30518, 11.99
0.691104 > 0.01: 3.28913, 1.016
0.37761 > 0.01: 13.5365, 8.425
0.782567 > 0.01: 2.2787e-05, 0.0001048
0.63461 > 0.01: 6.02096e-06, 2.2e-06
0.072896 > 0.01: 0.000833995, 0.0007732
0.122297 > 0.01: 0.0181952, 0.01597
0.430317 > 0.01: 6.96153, 12.22
0.0303448 > 0.01: 5.95263, 5.772
0.465723 > 0.01: 13.6914, 7.315
0.123828 > 0.01: 1.05842e-05, 1.208e-05
0.729487 > 0.01: 1.16224e-05, 3.144e-06
0.0449447 > 0.01: 0.000758218, 0.0007939
0.238439 > 0.01: 0.0164345, 0.02158
0.082101 > 0.01: 0.754421, 0.8219
0.308612 > 0.01: 7.44625, 10.77
0.153237 > 0.01: 10.3524, 8.766
0.155267 > 0.01: 12.5365, 10.59
0.40182 > 0.01: 6.16125e-06, 1.03e-05
0.662022 > 0.01: 1.82704e-05, 6.175e-06
0.0429606 > 0.01: 0.000770919, 0.0007378
0.081185 > 0.01: 0.0217208, 0.02364
0.0620825 > 0.01: 0.859777, 0.8064
0.149023 > 0.01: 7.71156, 9.062
0.313588 > 0.01: 11.5499, 7.928
0.0410826 > 0.01: 14.2975, 14.91
0.509679 > 0.01: 6.89881e-06, 1.407e-05
0.534442 > 0.01: 2.6291e-05, 1.224e-05
0.0869016 > 0.01: 0.000706934, 0.0006455
0.0696458 > 0.01: 0.0235502, 0.02191
0.100498 > 0.01: 0.842355, 0.7577
0.0529245 > 0.01: 7.10212, 7.499
0.376841 > 0.01: 9.51764, 5.931
0.120679 > 0.01: 17.7183, 20.15
0.544143 > 0.01: 1.00471e-05, 2.204e-05
0.452877 > 0.01: 3.94793e-05, 2.16e-05
0.4491 > 0.01: 0.000608745, 0.001105
0.0899584 > 0.01: 0.0217136, 0.02386
0.0701012 > 0.01: 0.803636, 0.7473
0.10363 > 0.01: 6.15896, 6.871
0.234816 > 0.01: 6.79837, 5.202
0.0166034 > 0.01: 22.5296, 22.91
0.413285 > 0.01: 1.59469e-05, 2.718e-05
0.457781 > 0.01: 5.61028e-05, 3.042e-05
0.725799 > 0.01: 0.00102716, 0.003746
0.306598 > 0.01: 0.0235341, 0.03394
0.104613 > 0.01: 0.814955, 0.7297
0.0792768 > 0.01: 5.79227, 6.291
0.225377 > 0.01: 5.70987, 4.423
0.388921 > 0.01: 2.05506e-05, 3.363e-05
0.463201 > 0.01: 6.83869e-05, 3.671e-05
0.531961 > 0.01: 0.00344149, 0.007353
0.309309 > 0.01: 0.0333051, 0.04822
```

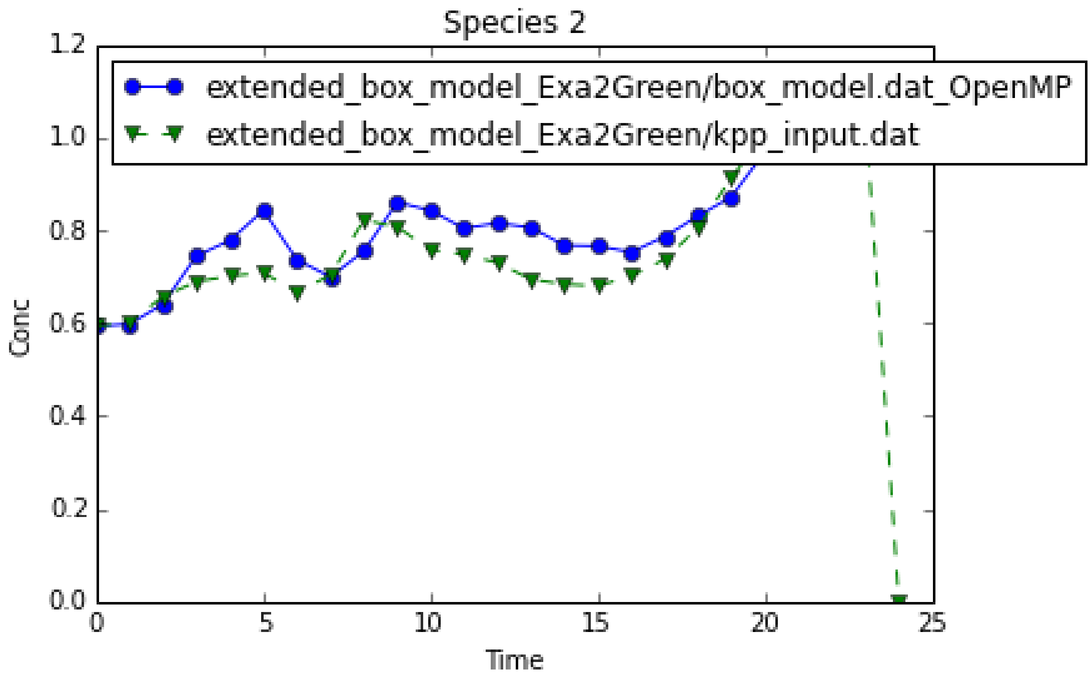
```
0.138463 > 0.01: 0.805653, 0.6941
0.0832804 > 0.01: 5.4059, 5.897
0.243297 > 0.01: 4.70594, 3.561
0.0449701 > 0.01: 27.3604, 26.13
0.406788 > 0.01: 2.56208e-05, 4.319e-05
0.525019 > 0.01: 7.26977e-05, 3.453e-05
0.19757 > 0.01: 0.00676694, 0.00543
0.109011 > 0.01: 0.767013, 0.6834
0.250737 > 0.01: 5.03505, 6.72
0.131914 > 0.01: 3.8798, 3.368
0.186572 > 0.01: 28.2508, 22.98
0.354662 > 0.01: 3.00211e-05, 4.652e-05
0.665815 > 0.01: 6.64601e-05, 2.221e-05
0.0508866 > 0.01: 0.00508791, 0.004829
0.284077 > 0.01: 0.0466209, 0.06512
0.112326 > 0.01: 0.765596, 0.6796
0.088658 > 0.01: 5.51909, 6.056
0.0577668 > 0.01: 4.10832, 3.871
0.0568325 > 0.01: 25.2871, 23.85
0.258624 > 0.01: 2.78757e-05, 3.76e-05
0.404958 > 0.01: 4.68202e-05, 2.786e-05
0.469191 > 0.01: 0.0046043, 0.002444
0.0684329 > 0.01: 0.752388, 0.7009
0.248252 > 0.01: 6.03127, 8.023
0.216984 > 0.01: 3.54158, 4.523
0.170215 > 0.01: 24.6931, 20.49
0.066102 > 0.01: 3.47784e-05, 3.724e-05
0.482602 > 0.01: 3.02862e-05, 1.567e-05
0.306537 > 0.01: 0.00237792, 0.001649
0.0620631 > 0.01: 0.786194, 0.7374
0.301392 > 0.01: 8.44617, 12.09
0.221569 > 0.01: 3.81275, 4.898
0.279483 > 0.01: 20.6518, 14.88
0.0248322 > 0.01: 3.81063e-05, 3.716e-05
0.515672 > 0.01: 1.53202e-05, 7.42e-06
0.357408 > 0.01: 0.0016264, 0.002531
0.179957 > 0.01: 0.0649146, 0.07916
0.0303167 > 0.01: 0.829962, 0.8048
0.250179 > 0.01: 13.0769, 17.44
0.0931744 > 0.01: 3.67436, 3.332
0.306686 > 0.01: 14.2446, 9.876
0.203433 > 0.01: 4.21703e-05, 5.294e-05
0.489747 > 0.01: 6.88286e-06, 3.512e-06
0.412684 > 0.01: 0.00251841, 0.004288
0.112568 > 0.01: 0.0797269, 0.08984
0.0450039 > 0.01: 0.871147, 0.9122
0.162103 > 0.01: 19.3051, 23.04
0.315891 > 0.01: 1.31212, 1.918
0.287707 > 0.01: 8.10902, 5.776
0.140928 > 0.01: 9.1366e-05, 7.849e-05
0.041002 > 0.01: 2.18457e-06, 2.095e-06
0.300832 > 0.01: 0.0042873, 0.006132
0.0760752 > 0.01: 0.0902028, 0.09763
```

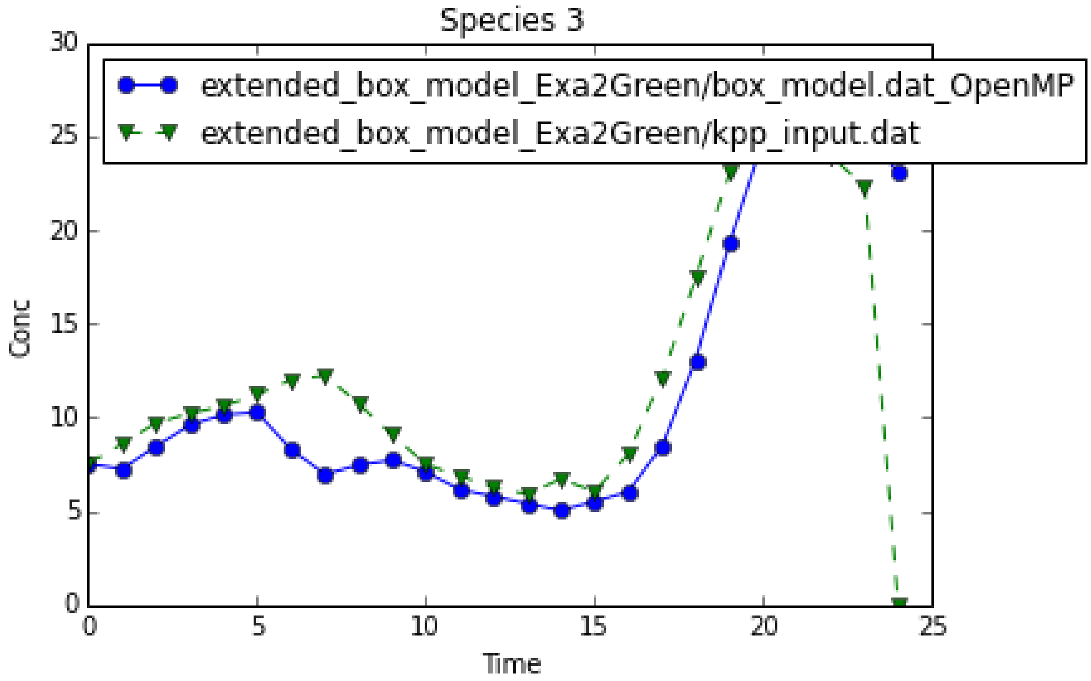
```
0.0397176 > 0.01: 0.967965, 1.008
0.0172417 > 0.01: 24.795, 25.23
0.986597 > 0.01: 0.0195825, 1.461
0.0683707 > 0.01: 3.8616, 4.145
0.732679 > 0.01: 0.000344305, 9.204e-05
0.824391 > 0.01: 3.04682e-07, 1.735e-06
0.167855 > 0.01: 0.00613541, 0.007373
0.0538859 > 0.01: 26.5718, 25.14
0.995022 > 0.01: 0.00621802, 1.249
0.270182 > 0.01: 2.66237, 3.648
0.676004 > 0.01: 0.000299479, 9.703e-05
0.878692 > 0.01: 1.79172e-07, 1.477e-06
0.0642084 > 0.01: 0.00737404, 0.00788
0.0443969 > 0.01: 0.0982102, 0.09385
0.0303889 > 0.01: 1.12726, 1.093
0.0887386 > 0.01: 26.2932, 23.96
0.989872 > 0.01: 0.011354, 1.121
0.357226 > 0.01: 2.39819, 3.731
0.598878 > 0.01: 0.000273234, 0.0001096
0.880205 > 0.01: 1.66995e-07, 1.394e-06
0.101304 > 0.01: 0.00788246, 0.008771
0.025376 > 0.01: 0.0940465, 0.09166
0.0791385 > 0.01: 1.14458, 1.054
0.10565 > 0.01: 24.979, 22.34
0.993445 > 0.01: 0.00597157, 0.911
0.385727 > 0.01: 2.59408, 4.223
0.584938 > 0.01: 0.000319229, 0.0001325
0.887357 > 0.01: 1.55222e-07, 1.378e-06
1 > 0.01: 0.00878438, 0
1 > 0.01: 0.0918722, 0
1 > 0.01: 1.10442, 0
1 > 0.01: 23.1185, 0
1 > 0.01: 0.00024813, 0
1 > 0.01: 3.25886, 0
1 > 0.01: 0.000465228, 0
1 > 0.01: 1.55954e-07, 0
SNR: 22.614475db
184 samples with relative error > 0.01
```

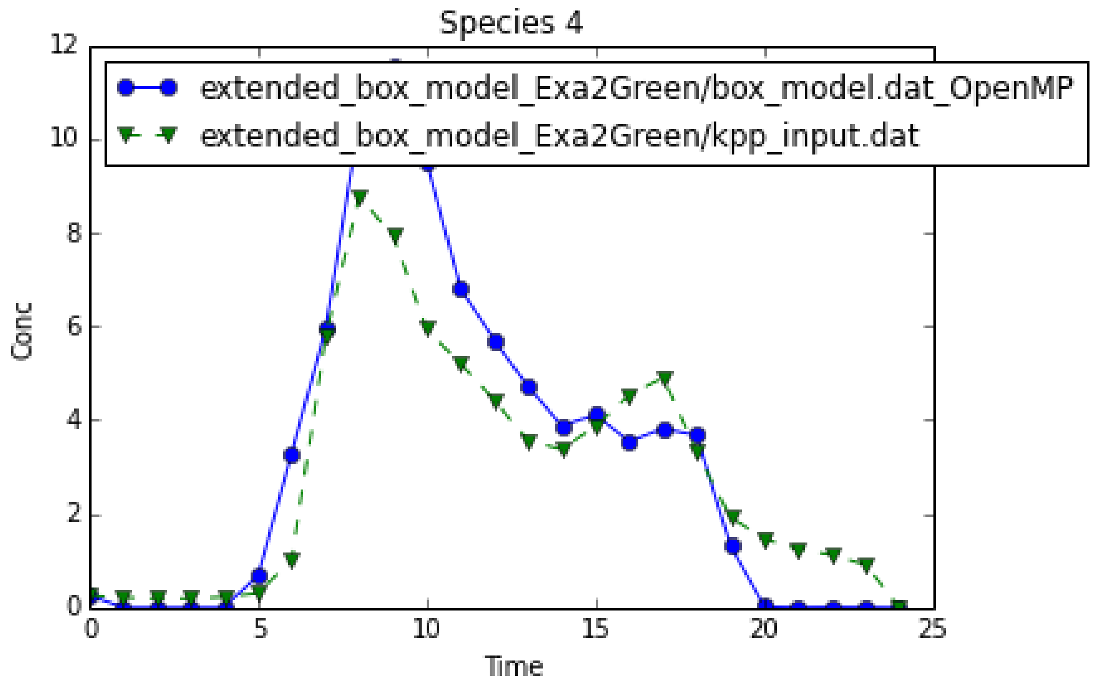
```
In [11]: plot_dat([kpp_dat_1, kpp_dat_2], names=[kpp_file_1, kpp_file_2], titles=None)
```

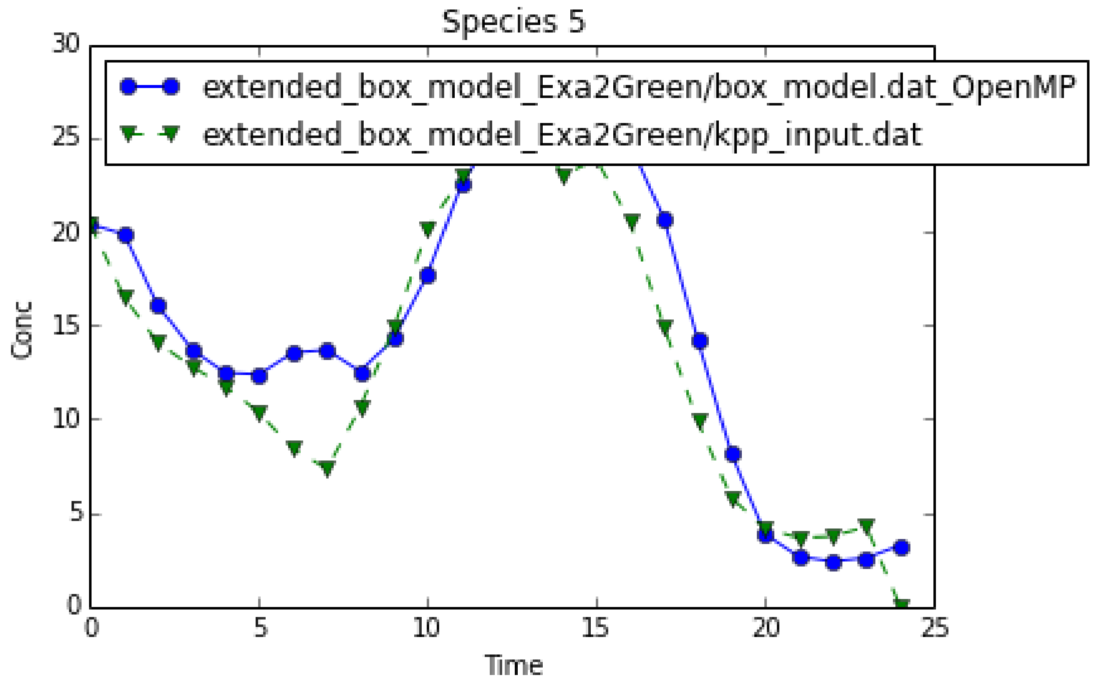

Species 0
extended_box_model_Exa2Green/box_model.dat_OpenMP
extended_box_model_Exa2Green/kpp_input.dat
Conc
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
Time
0
5
10
15
20
25

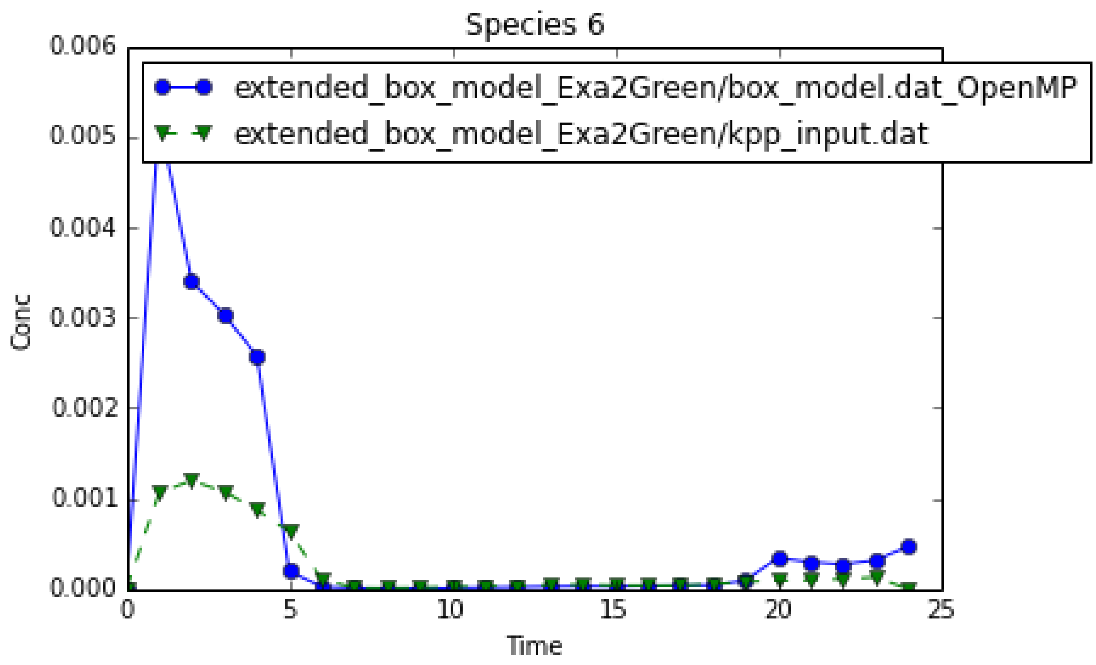

Species 1
extended_box_model_Exa2Green/box_model.dat_OpenMP
extended_box_model_Exa2Green/kpp_input.dat
Conc
0.10
0.08
0.06
0.04
0.02
0.00
Time
0
5
10
15
20
25

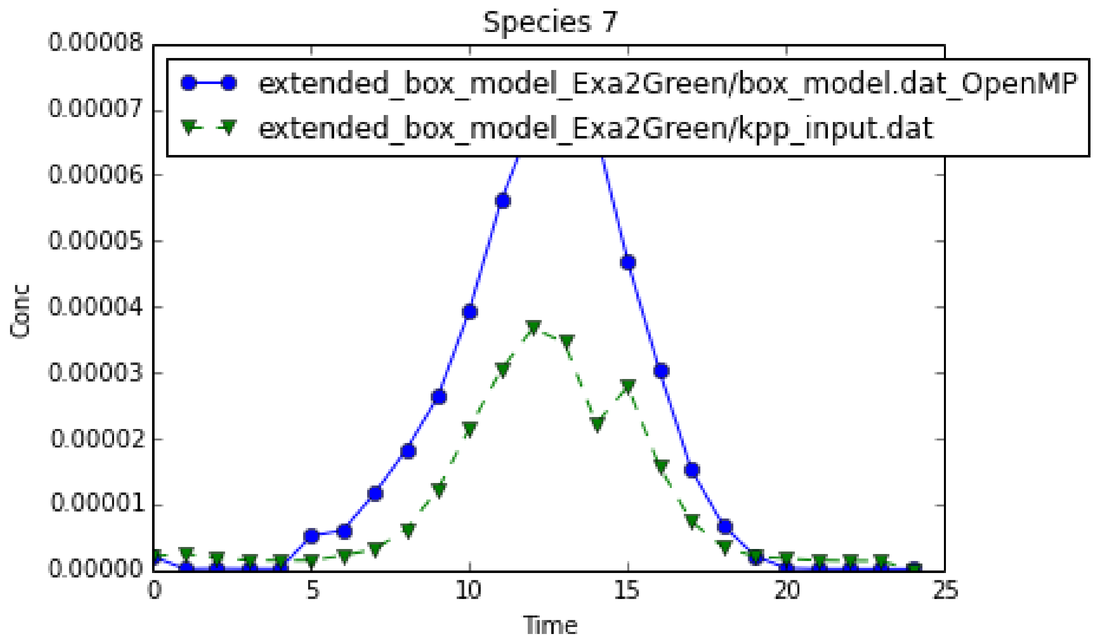

Species 2
extended_box_model_Exa2Green/box_model.dat_OpenMP
extended_box_model_Exa2Green/kpp_input.dat
Conc
Time
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20
25


Species 3
extended_box_model_Exa2Green/box_model.dat_OpenMP
extended_box_model_Exa2Green/kpp_input.dat
Conc
Time
0
5
10
15
20
25
30
0
5
10
15
20
25


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

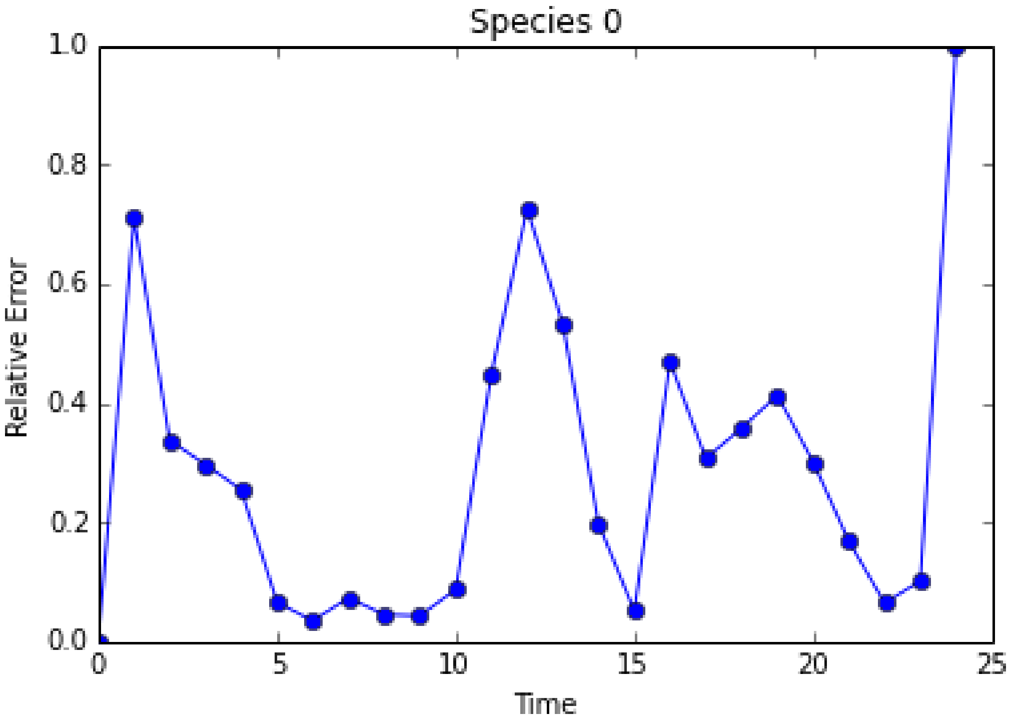

Species 5
extended_box_model_Exa2Green/box_model.dat_OpenMP
extended_box_model_Exa2Green/kpp_input.dat
Conc
Time
0
5
10
15
20
25
30
0
5
10
15
20
25

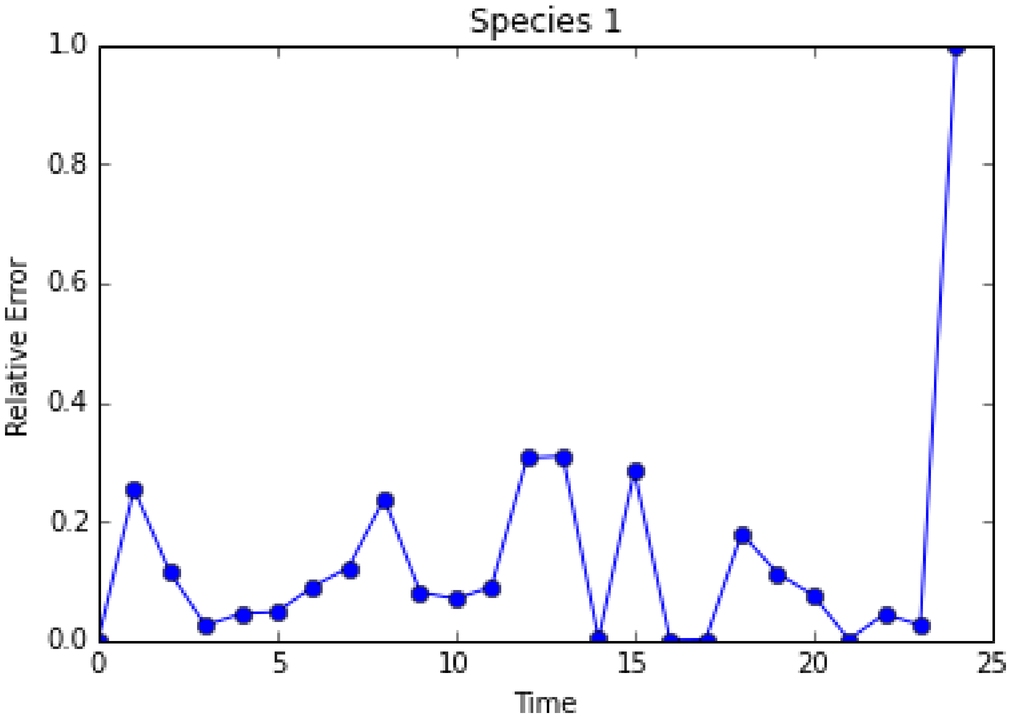


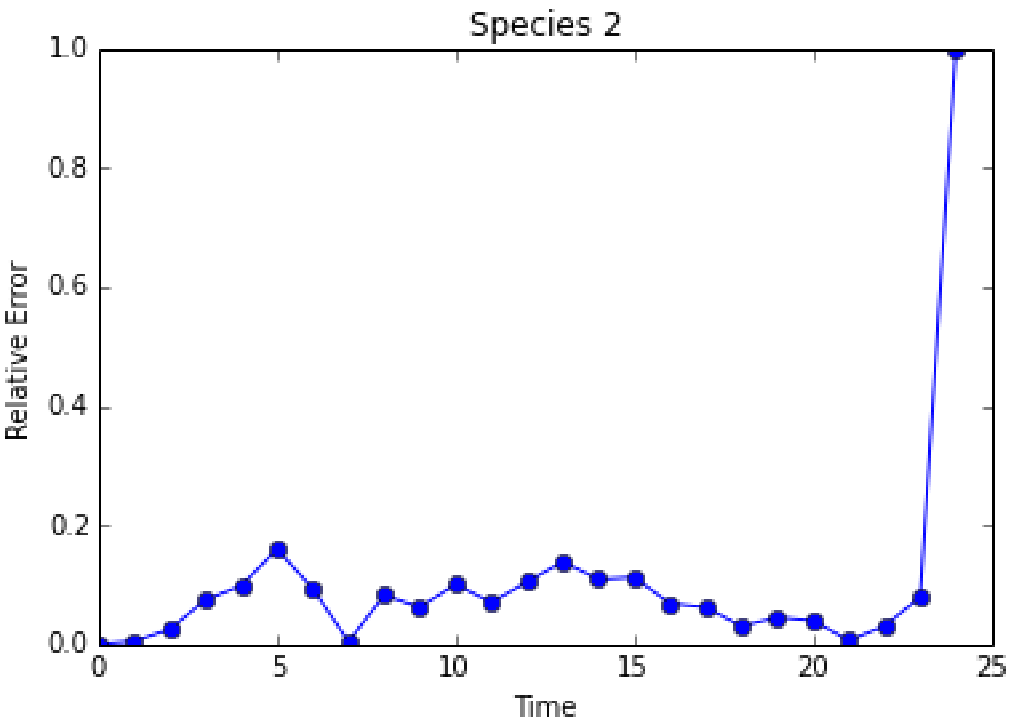


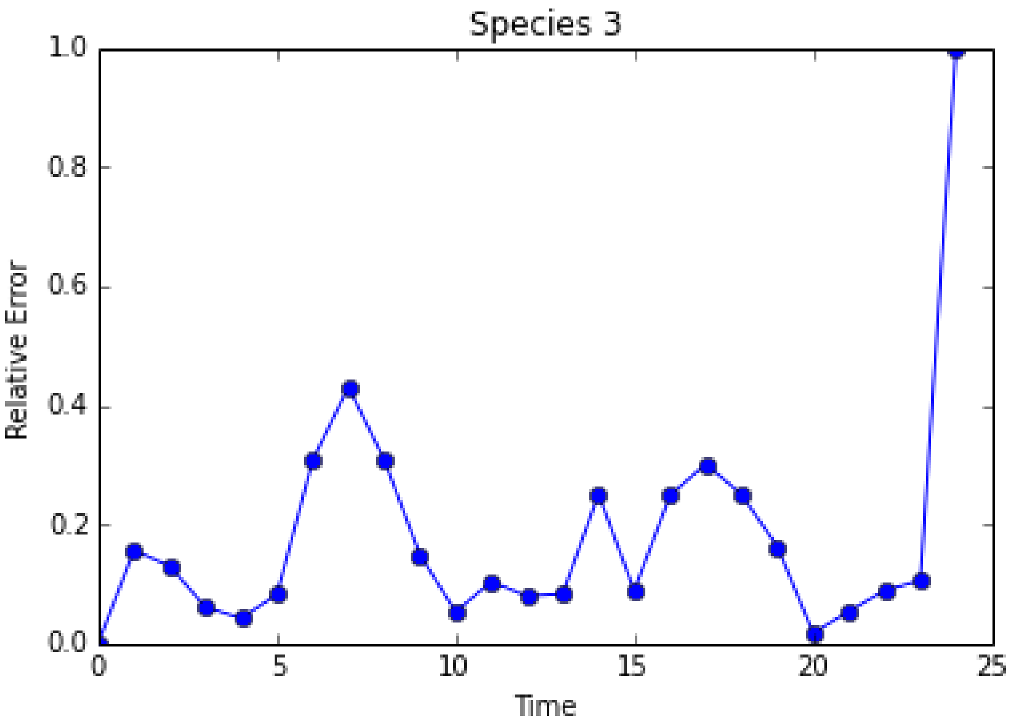
In [12]:

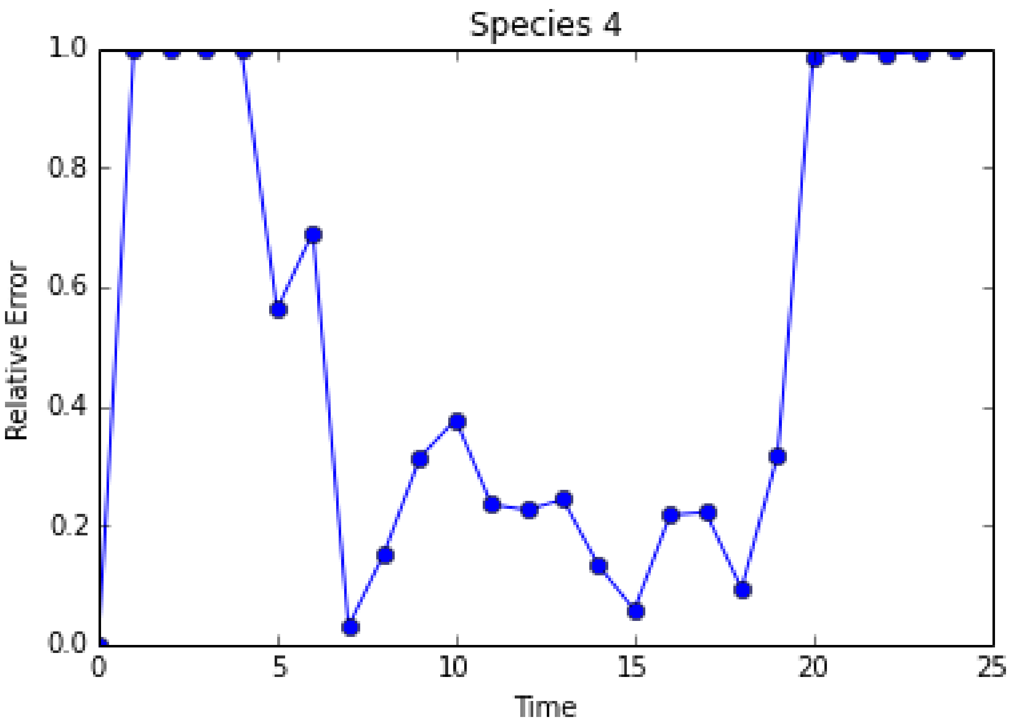
```
plot_dat([err_dat], ylabel='Relative Error')
```


Species 0

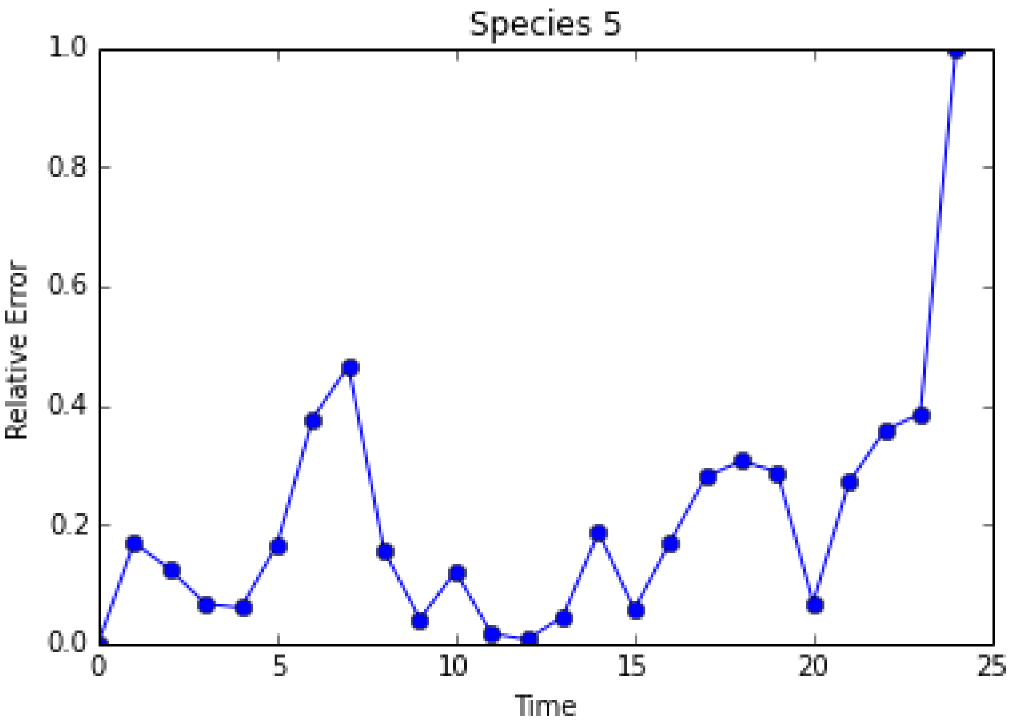

Species 1


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

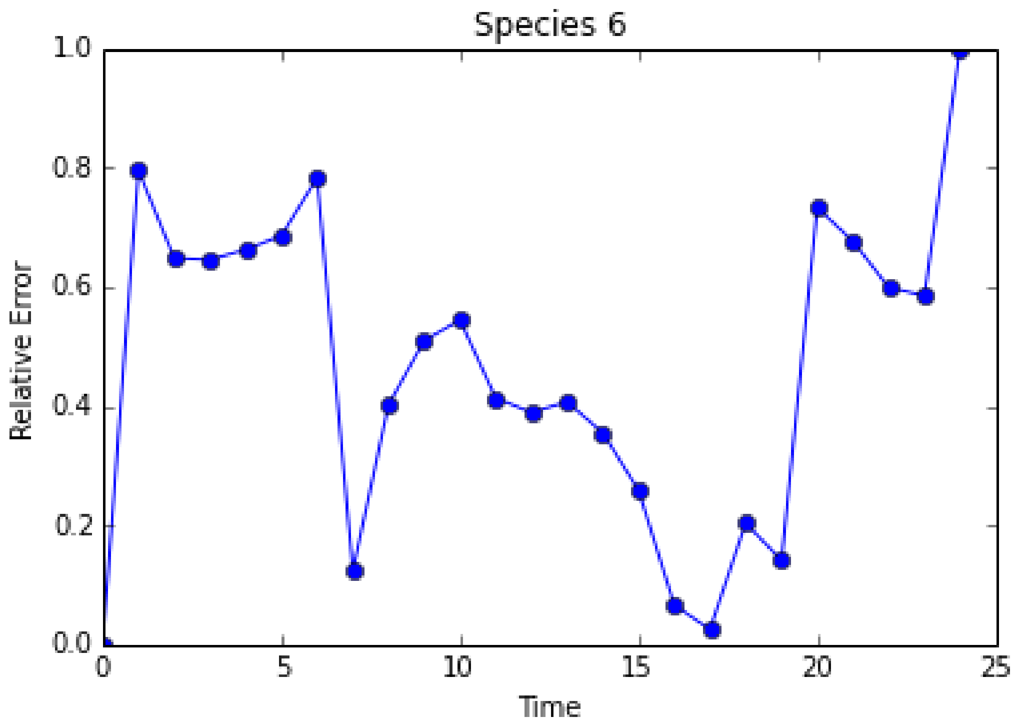

Species 3
Relative Error
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
Time

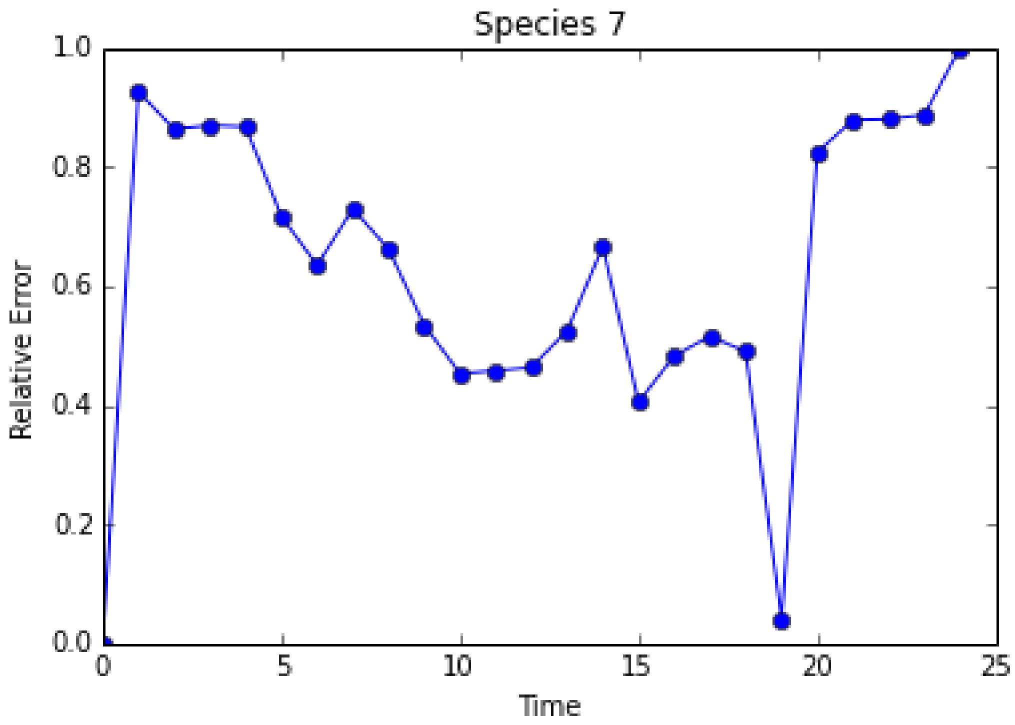


Species 4



Species 5


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


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

```
In [12]:
```