# kpp_vs_kppa

**Unknown Author**

November 3, 2014

In [19]:
```
kppa_file = 'extended_box_model_Exa2Green/box_model.dat_OpenMP'
kpp_file = 'extended_box_model_Exa2Green/kpp-2.2.1.dat'
```

In [20]:
```
%matplotlib inline
import re
from itertools import cycle
from pylab import *
from matplotlib.markers import MarkerStyle
import matplotlib.pyplot as plt

ATOL = 1.0e-3
RTOL = 1.0e-4
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
```

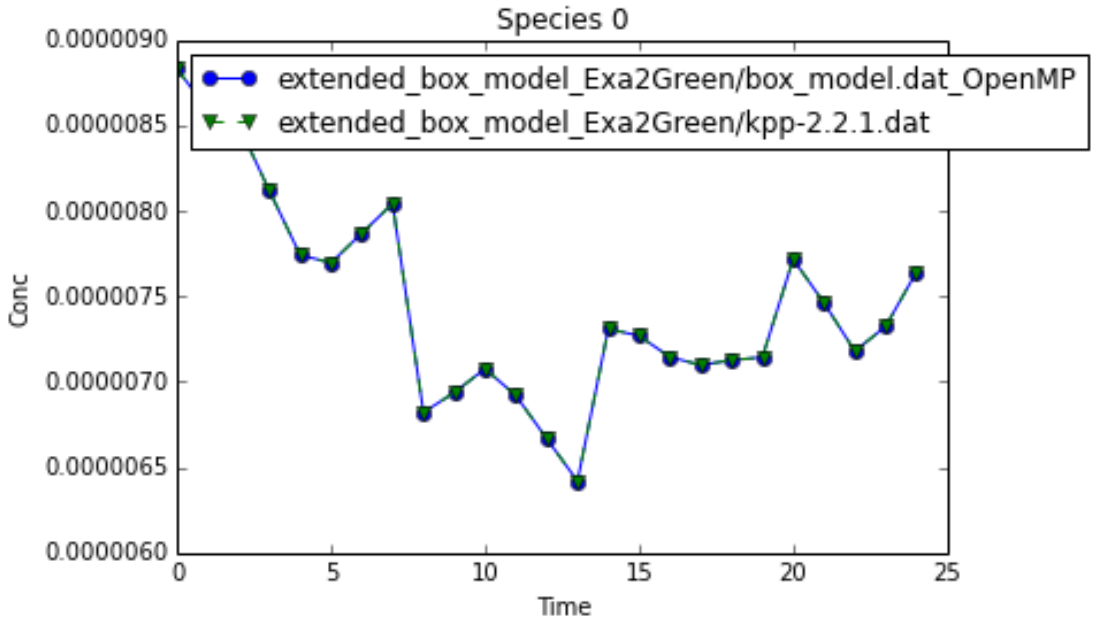
In [21]:
```
kppa_dat = read_datfile(kppa_file, 0, 1)
kpp_dat = read_datfile(kpp_file, 0, 1)
```

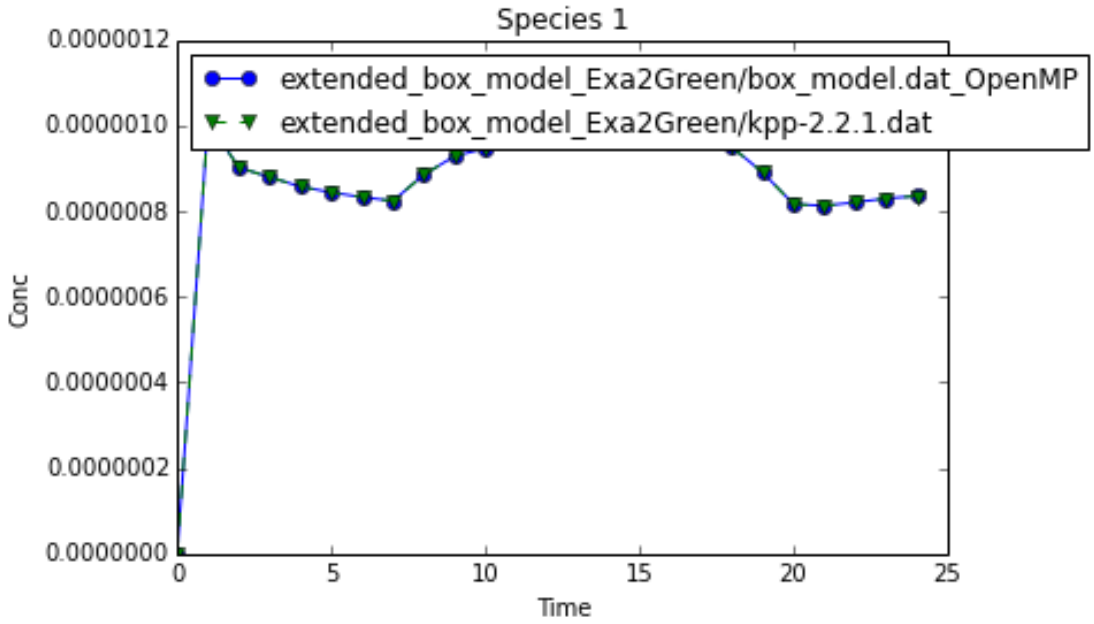
In [22]:
```
err_dat = calc_err(kpp_dat, kppa_dat)
```

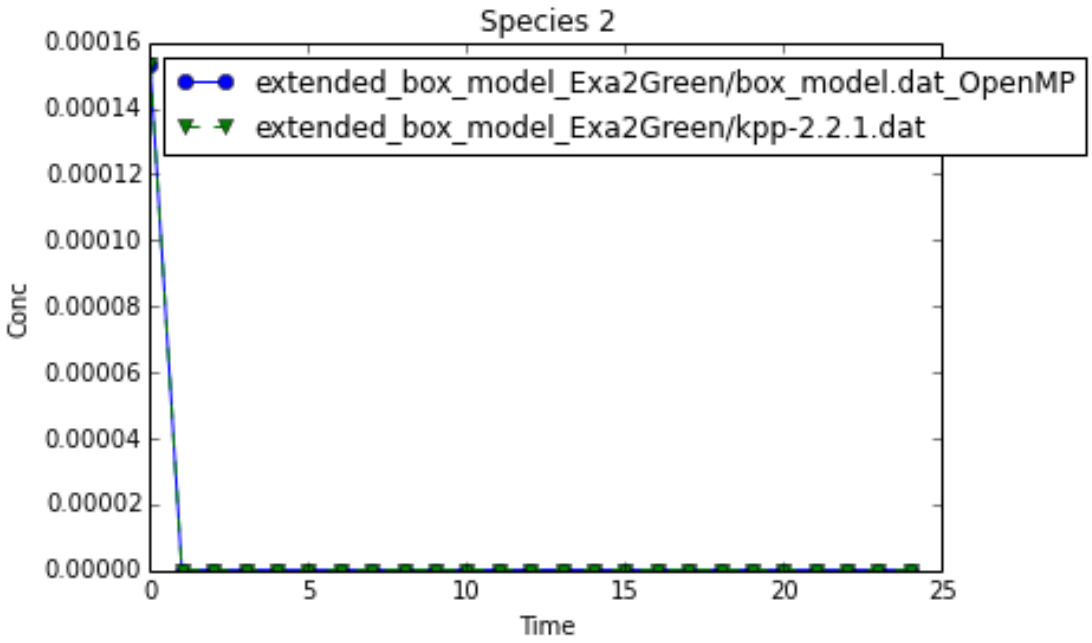
```
0.000456859 > 0.0001: 1.00877e-06, 1.00923e-06
0.00045137 > 0.0001: 4.80594e-12, 4.80811e-12
0.000117084 > 0.0001: 9.01542e-07, 9.01648e-07
0.000147543 > 0.0001: 5.083e-13, 5.08375e-13
0.000115618 > 0.0001: 8.79144e-07, 8.79246e-07
0.000133387 > 0.0001: 4.79072e-13, 4.79136e-13
0.000161565 > 0.0001: 8.56961e-07, 8.571e-07
0.000166133 > 0.0001: 4.51617e-13, 4.51692e-13
0.000153734 > 0.0001: 8.4341e-07, 8.43539e-07
0.000159377 > 0.0001: 4.46025e-13, 4.46096e-13
0.0001372 > 0.0001: 8.32331e-07, 8.32445e-07
0.000145914 > 0.0001: 4.42988e-13, 4.43053e-13
0.000112778 > 0.0001: 8.23455e-07, 8.23548e-07
0.000134729 > 0.0001: 4.40927e-13, 4.40986e-13
0.000148162 > 0.0001: 8.84422e-07, 8.84554e-07
0.000169431 > 0.0001: 6.13703e-13, 6.13807e-13
0.000186275 > 0.0001: 9.27975e-07, 9.28148e-07
0.000216142 > 0.0001: 5.53923e-13, 5.54043e-13
0.000204924 > 0.0001: 9.46166e-07, 9.4636e-07
0.000232464 > 0.0001: 4.72812e-13, 4.72922e-13
0.000207704 > 0.0001: 9.82611e-07, 9.82815e-07
0.00023357 > 0.0001: 5.8514e-13, 5.85276e-13
0.000231726 > 0.0001: 1.02127e-06, 1.02151e-06
0.000263182 > 0.0001: 7.06366e-13, 7.06552e-13
0.00046547 > 0.0001: 1.05199e-06, 1.05248e-06
0.000460749 > 0.0001: 8.301e-13, 8.30482e-13
0.000265676 > 0.0001: 1.05298e-06, 1.05326e-06
0.000303453 > 0.0001: 8.38412e-13, 8.38667e-13
0.000211264 > 0.0001: 1.03471e-06, 1.03493e-06
0.000236005 > 0.0001: 7.26819e-13, 7.2699e-13
0.000211938 > 0.0001: 1.00673e-06, 1.00695e-06
0.000238239 > 0.0001: 6.11022e-13, 6.11168e-13
0.00020812 > 0.0001: 9.84239e-07, 9.84444e-07
0.000236558 > 0.0001: 6.26826e-13, 6.26975e-13
0.000207092 > 0.0001: 9.49351e-07, 9.49548e-07
0.000240844 > 0.0001: 6.32445e-13, 6.32597e-13
0.000320208 > 0.0001: 8.89662e-07, 8.89946e-07
0.000318725 > 0.0001: 6.17852e-13, 6.18049e-13
0.000229548 > 0.0001: 8.16063e-07, 8.16251e-07
0.000293947 > 0.0001: 4.6124e-13, 4.61375e-13
0.000255894 > 0.0001: 8.12122e-07, 8.1233e-07
0.000329246 > 0.0001: 4.34576e-13, 4.34719e-13
```

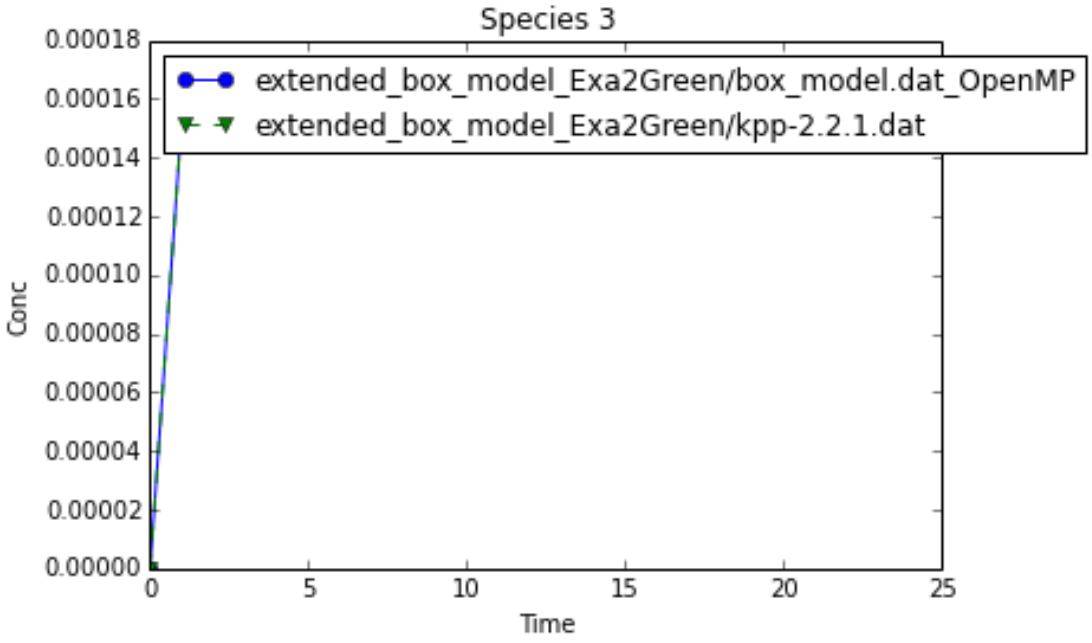
```
0.000130015 > 0.0001: 8.20189e-07, 8.20296e-07
0.000171543 > 0.0001: 4.13684e-13, 4.13755e-13
0.000240358 > 0.0001: 8.29226e-07, 8.29425e-07
0.000237046 > 0.0001: 4.40238e-13, 4.40342e-13
0.000155225 > 0.0001: 8.34922e-07, 8.35052e-07
0.000160438 > 0.0001: 4.67564e-13, 4.67639e-13
SNR: 233.844508db
48 samples with relative error > 0.0001
```

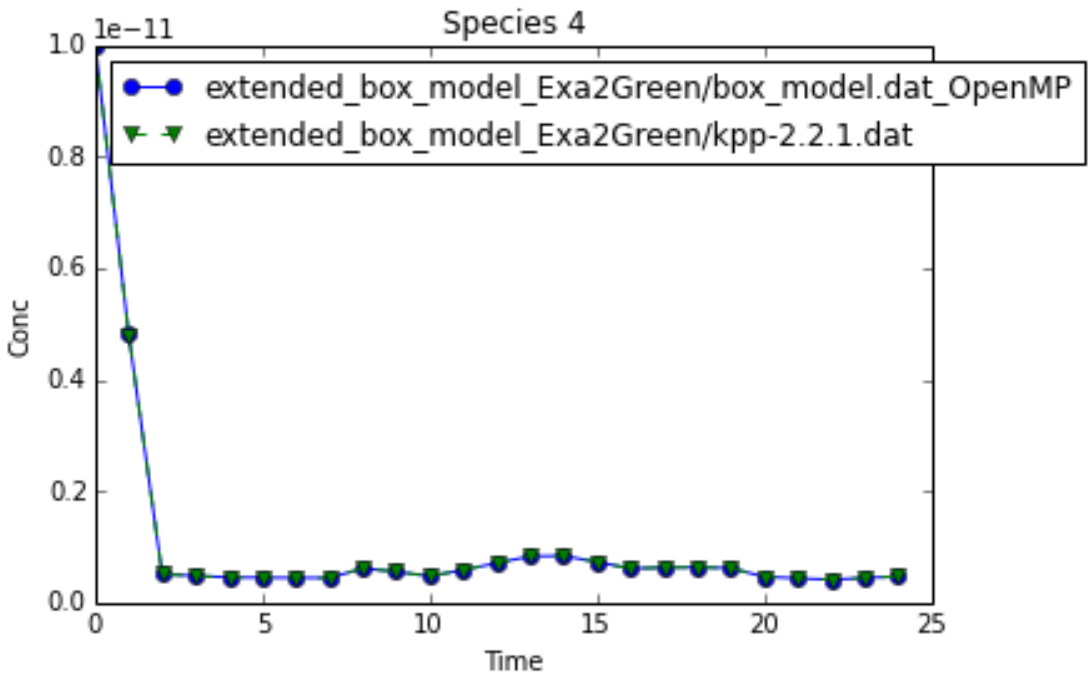
In [23]:
```
plot_dat([kppa_dat, kpp_dat], names=[kppa_file, kpp_file], titles=None)
```



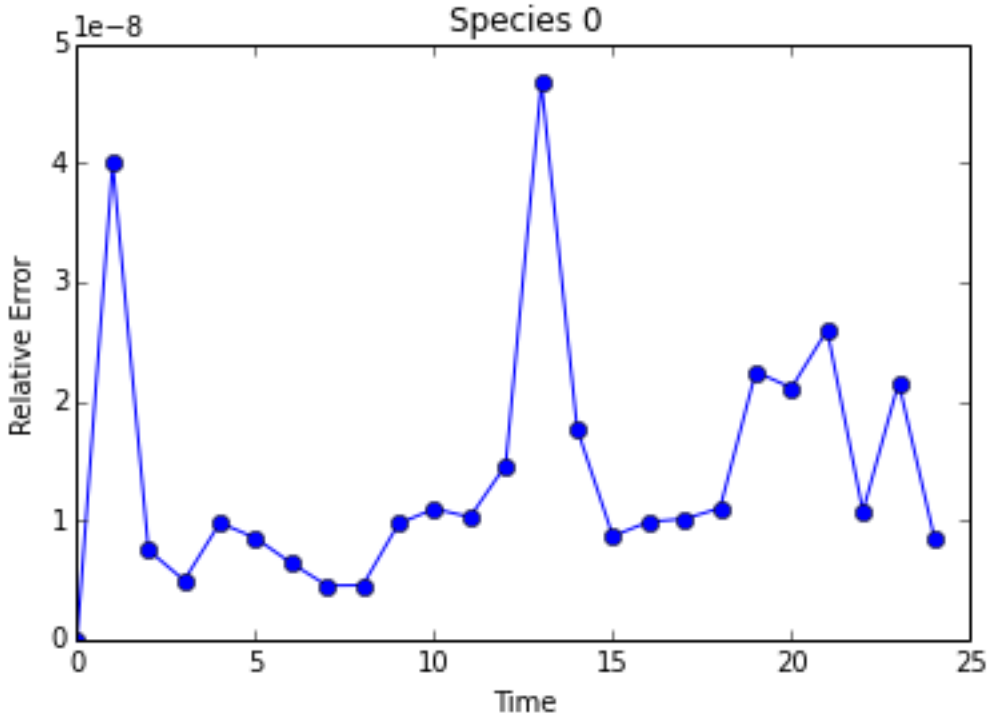



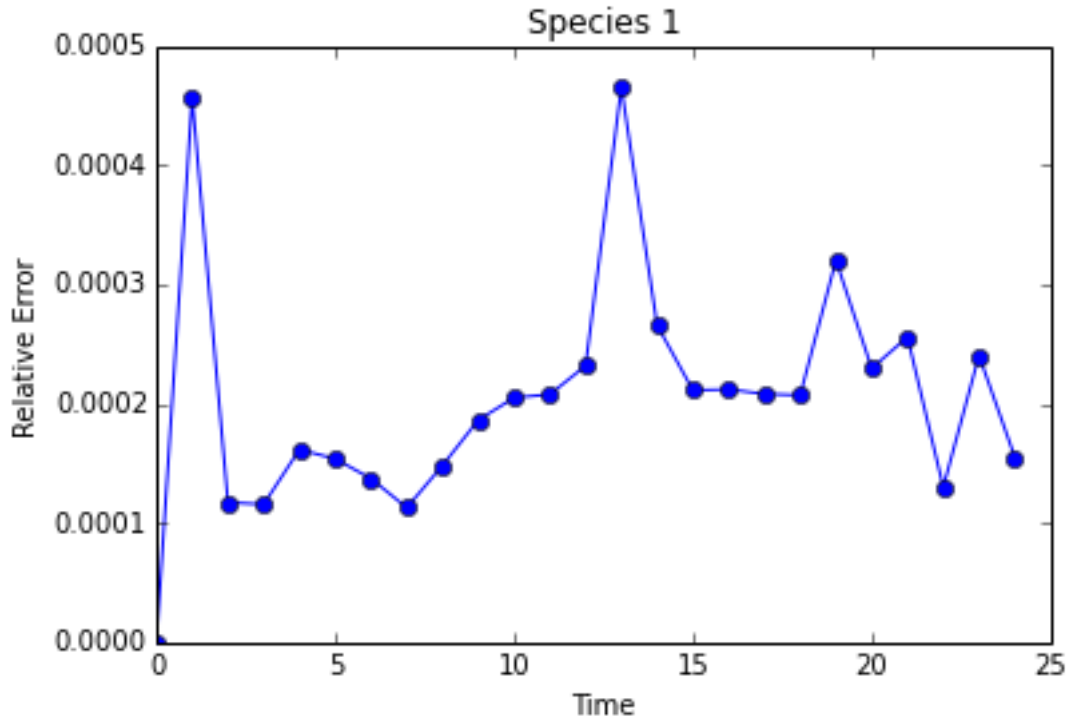

Species 2
extended_box_model_Exa2Green/box_model.dat_OpenMP
extended_box_model_Exa2Green/kpp-2.2.1.dat
Conc
Time
0.00016
0.00014
0.00012
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
0
5
10
15
20
25

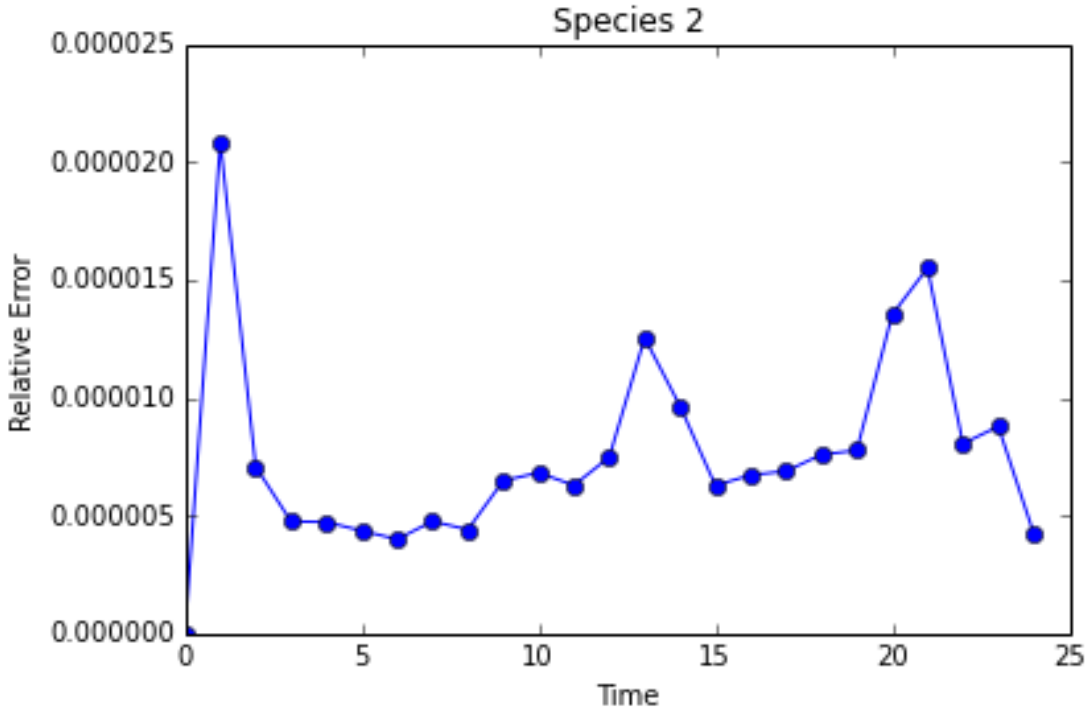

Species 3
extended_box_model_Exa2Green/box_model.dat_OpenMP
extended_box_model_Exa2Green/kpp-2.2.1.dat
Conc
Time
0.00018
0.00016
0.00014
0.00012
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
0
5
10
15
20
25

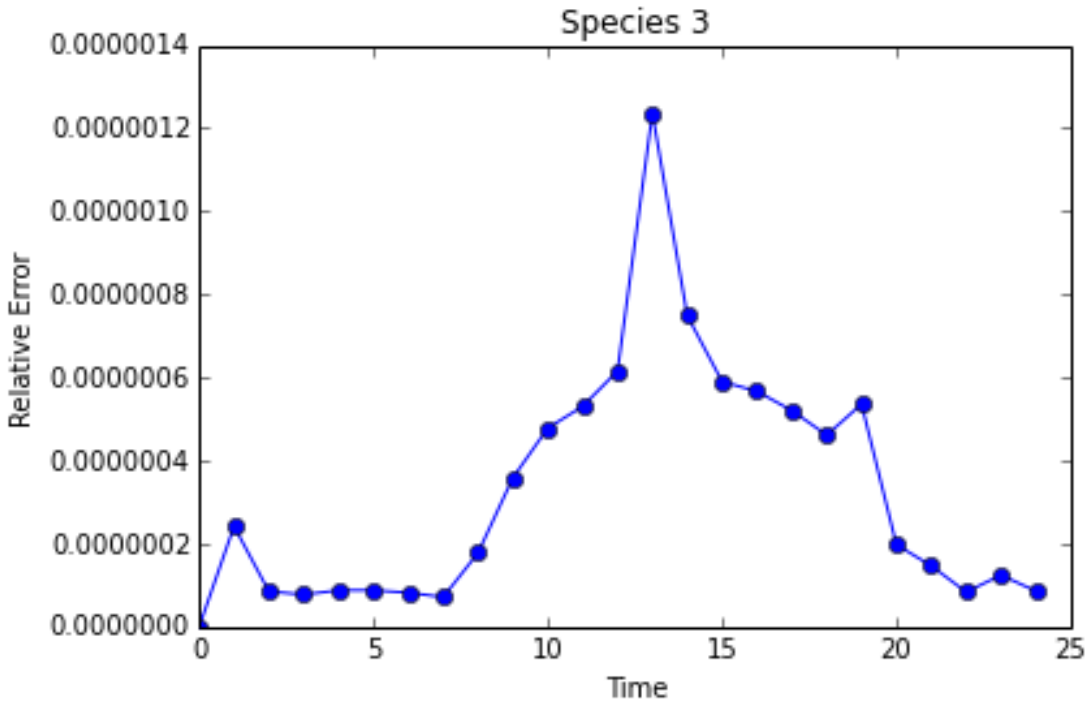


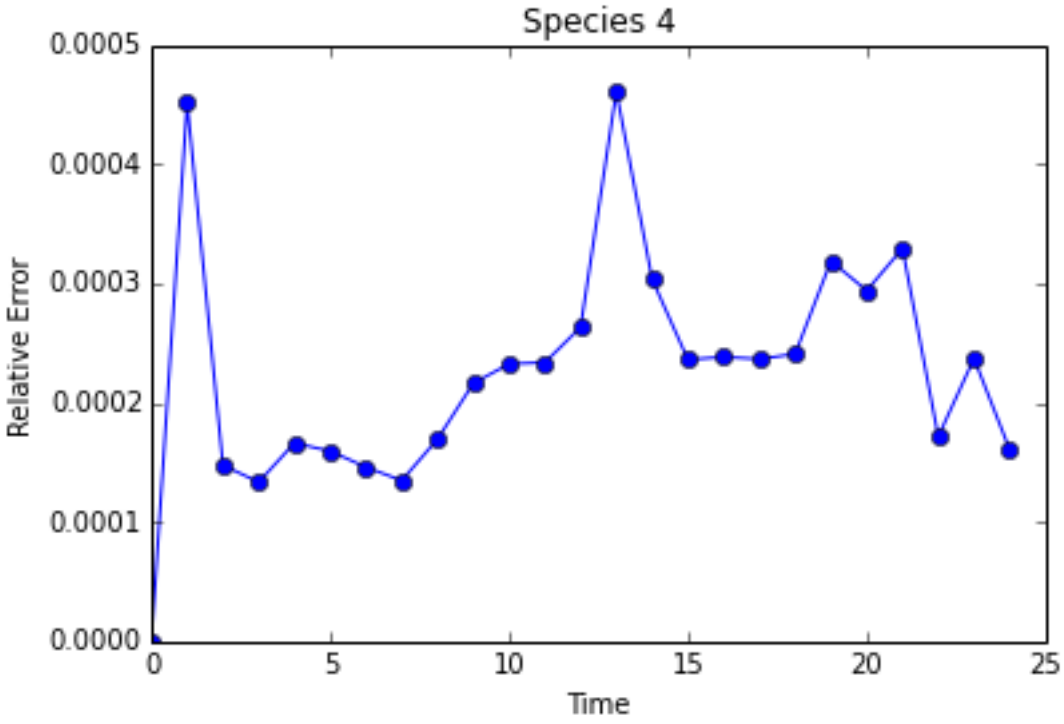

In [24]:
```
plot_dat([err_dat], ylabel='Relative Error')
```




Species 1
Relative Error
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
0
5
10
15
20
25
Time


Species 2
Relative Error
0.000025
0.000020
0.000015
0.000010
0.000005
0.000000
0
5
10
15
20
25
Time


Species 3
Relative Error
Time
0.0000014
0.0000012
0.0000010
0.0000008
0.0000006
0.0000004
0.0000002
0.0000000
0
5
10
15
20
25


Species 4
Relative Error
Time
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
0
5
10
15
20
25

In [24]: