

# Kppa Verification using KPP-generated code

Created: 2 June 2014

John C. Linford <jlinford@paratools.com>

Copyright (c) ParaTools, Inc.

## Description

This notebook shows how differences in Kppa's default values vs. KPP impact the solution vector.

## Instructions

1. Compile and run the `box_model_kppa_defaults` and `box_model_kpp221` codes
2. Execute all cells in this notebook.

```
In [20]: kppa_file = 'box_model_kppa_defaults/box_model.dat'
kpp_file = 'box_model_kpp221/box_model.dat'
```

## Processing Script

Skip down for results

```
In [21]: %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 be
    ginning in field cstart.
    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 [22]: kppa_dat = read_datfile(kppa_file, 0, 3)
         kpp_dat = read_datfile(kpp_file, 1, 1)

```

# Results: Relative Error and SNR

The following cell shows the relative error per concentration for each time step written to the .dat files. A signal-to-noise ratio and relative error count are shown below. As a rule of thumb, SNR  $\geq 250$  indicates solution match to five decimal places. This is a **relative error calculation**, so concentrations close to zero are likely to have high relative error but low absolute error.

```
In [23]: err_dat = calc_err(kpp_dat, kppa_dat)

0.00221674 > 0.0001: 2.05728, 2.05272
0.0211062 > 0.0001: 0.000348496, 0.00034114
0.150604 > 0.0001: 0.0747899, 0.0880506
0.0277657 > 0.0001: 0.0598564, 0.0581944
0.36951 > 0.0001: 3.30119e-06, 5.23591e-06
0.00482968 > 0.0001: 2.1192, 2.10896
0.05623 > 0.0001: 0.00032153, 0.00030345
0.13959 > 0.0001: 0.0621093, 0.0721857
0.0797873 > 0.0001: 0.0491468, 0.0452255
0.383539 > 0.0001: 3.0591e-06, 4.96236e-06
0.00579982 > 0.0001: 2.17601, 2.16339
0.0832211 > 0.0001: 0.000299782, 0.000274834
0.133312 > 0.0001: 0.0523327, 0.0603824
0.120216 > 0.0001: 0.0409397, 0.0360181
0.395384 > 0.0001: 2.83006e-06, 4.68075e-06
0.00641395 > 0.0001: 2.23097, 2.21666
0.104782 > 0.0001: 0.000282194, 0.000252625
0.131085 > 0.0001: 0.0446299, 0.0513628
0.152911 > 0.0001: 0.0345554, 0.0292715
0.406407 > 0.0001: 2.6217e-06, 4.41667e-06
0.00740289 > 0.0001: 2.28502, 2.2681
0.122053 > 0.0001: 0.000267879, 0.000235184
0.129698 > 0.0001: 0.0384707, 0.0442038
0.179207 > 0.0001: 0.0295247, 0.0242337
0.412249 > 0.0001: 2.4366e-06, 4.14564e-06
0.00911115 > 0.0001: 2.33818, 2.31688
0.136307 > 0.0001: 0.000256132, 0.000221219
0.129835 > 0.0001: 0.0334825, 0.0384784
0.200604 > 0.0001: 0.0255088, 0.0203916
0.415482 > 0.0001: 2.27423e-06, 3.89078e-06
0.0116235 > 0.0001: 2.39016, 2.36238
0.147694 > 0.0001: 0.000246406, 0.000210013
0.130086 > 0.0001: 0.029398, 0.0337941
0.217504 > 0.0001: 0.0222636, 0.0174212
0.413788 > 0.0001: 2.13271e-06, 3.63812e-06
0.0149003 > 0.0001: 2.44058, 2.40422
0.157512 > 0.0001: 0.000238279, 0.000200747
0.13134 > 0.0001: 0.0260216, 0.029956
0.231266 > 0.0001: 0.0196128, 0.015077
0.410343 > 0.0001: 2.00973e-06, 3.4083e-06
0.0187687 > 0.0001: 2.48913, 2.44241
```

0.165275 > 0.0001: 0.000231429, 0.000193179  
0.131744 > 0.0001: 0.023208, 0.0267295  
0.241939 > 0.0001: 0.0174269, 0.0132106  
0.401837 > 0.0001: 1.90294e-06, 3.18131e-06  
0.0230849 > 0.0001: 2.53555, 2.47701  
0.170962 > 0.0001: 0.000225604, 0.000187034  
0.131473 > 0.0001: 0.0208472, 0.024003  
0.249164 > 0.0001: 0.0156098, 0.0117204  
0.388206 > 0.0001: 1.81018e-06, 2.9588e-06  
0.027665 > 0.0001: 2.57966, 2.5083  
0.175056 > 0.0001: 0.00022061, 0.000181991  
0.1306 > 0.0001: 0.0188548, 0.0216871  
0.253738 > 0.0001: 0.0140887, 0.0105139  
0.370573 > 0.0001: 1.7295e-06, 2.74773e-06  
0.0323264 > 0.0001: 2.62139, 2.53665  
0.177513 > 0.0001: 0.000216294, 0.000177899  
0.128746 > 0.0001: 0.0171647, 0.0197011  
0.255791 > 0.0001: 0.0128081, 0.00953188  
0.348427 > 0.0001: 1.65921e-06, 2.54647e-06  
0.0368994 > 0.0001: 2.66068, 2.5625  
0.178427 > 0.0001: 0.000212538, 0.000174615  
0.125706 > 0.0001: 0.015725, 0.0179859  
0.255583 > 0.0001: 0.0117245, 0.00872789  
0.321896 > 0.0001: 1.59785e-06, 2.35635e-06  
0.041232 > 0.0001: 2.69755, 2.58632  
0.177843 > 0.0001: 0.000209244, 0.000172032  
0.121278 > 0.0001: 0.0144939, 0.0164943  
0.253298 > 0.0001: 0.0108037, 0.00806718  
0.291038 > 0.0001: 1.54417e-06, 2.17807e-06  
0.0451921 > 0.0001: 2.73205, 2.60858  
0.175825 > 0.0001: 0.000206339, 0.000170059  
0.115309 > 0.0001: 0.013438, 0.0151895  
0.249111 > 0.0001: 0.0100187, 0.00752292  
0.256141 > 0.0001: 1.49709e-06, 2.0126e-06  
0.0486667 > 0.0001: 2.76425, 2.62973  
0.172444 > 0.0001: 0.000203759, 0.000168622  
0.10768 > 0.0001: 0.0125298, 0.0140418  
0.243176 > 0.0001: 0.00934736, 0.00707431  
0.217693 > 0.0001: 1.45571e-06, 1.86079e-06  
0.0515615 > 0.0001: 2.79425, 2.65018  
0.167789 > 0.0001: 0.000201455, 0.000167653  
0.0983088 > 0.0001: 0.0117467, 0.0130274  
0.235637 > 0.0001: 0.00877192, 0.00670494  
0.176461 > 0.0001: 1.41923e-06, 1.72333e-06  
0.053799 > 0.0001: 2.82216, 2.67033  
0.161964 > 0.0001: 0.000199386, 0.000167093  
0.0871375 > 0.0001: 0.0110701, 0.0121268  
0.226639 > 0.0001: 0.00827775, 0.00640168  
0.133514 > 0.0001: 1.38699e-06, 1.6007e-06  
0.0553174 > 0.0001: 2.84807, 2.69052  
0.15509 > 0.0001: 0.000197519, 0.000166886  
0.074127 > 0.0001: 0.0104845, 0.0113239  
0.216343 > 0.0001: 0.00785273, 0.00615385  
0.0902213 > 0.0001: 1.35842e-06, 1.49313e-06  
0.0560706 > 0.0001: 2.87211, 2.71107  
0.147303 > 0.0001: 0.000195826, 0.00016698

```

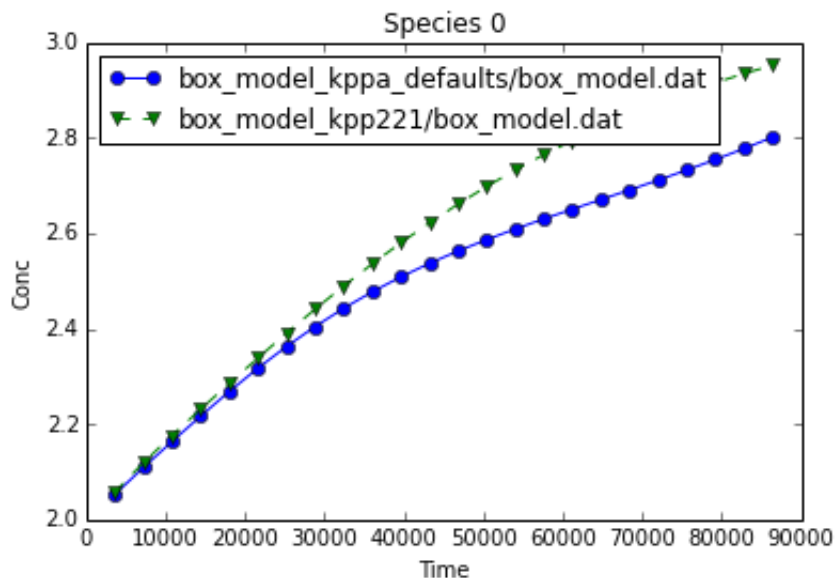
0.0592384 > 0.0001: 0.00997678, 0.010605
0.20494 > 0.0001: 0.00748682, 0.00595247
0.0481962 > 0.0001: 1.33303e-06, 1.40053e-06
0.0560276 > 0.0001: 2.89438, 2.73222
0.138758 > 0.0001: 0.000194283, 0.000167325
0.0424093 > 0.0001: 0.009536, 0.00995832
0.192686 > 0.0001: 0.00717159, 0.00578972
0.00916745 > 0.0001: 1.31039e-06, 1.32252e-06
0.0551734 > 0.0001: 2.915, 2.75417
0.129621 > 0.0001: 0.000192872, 0.000167872
0.023522 > 0.0001: 0.00915284, 0.00937332
0.179926 > 0.0001: 0.00689994, 0.00565846
0.0245812 > 0.0001: 1.29016e-06, 1.25845e-06
0.0535099 > 0.0001: 2.93408, 2.77707
0.120076 > 0.0001: 0.000191576, 0.000168572
0.00236897 > 0.0001: 0.00881943, 0.00884037
0.167135 > 0.0001: 0.0066659, 0.00555179
0.0508008 > 0.0001: 1.27201e-06, 1.20739e-06
0.0510571 > 0.0001: 2.9517, 2.801
0.110314 > 0.0001: 0.00019038, 0.000169379
0.020928 > 0.0001: 0.00852902, 0.00835053
0.154933 > 0.0001: 0.00646437, 0.00546283
0.0696517 > 0.0001: 1.25569e-06, 1.16823e-06
SNR: 55.557451db
120 samples with relative error > 0.0001

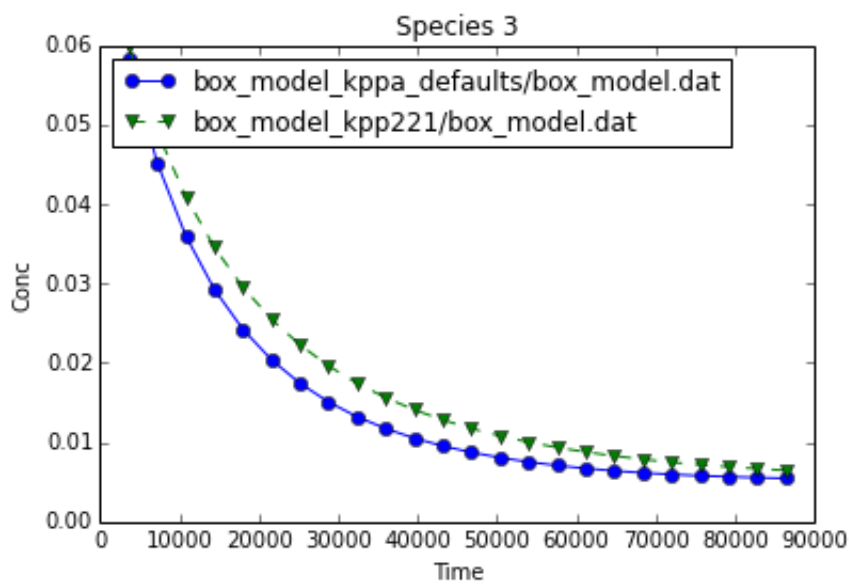
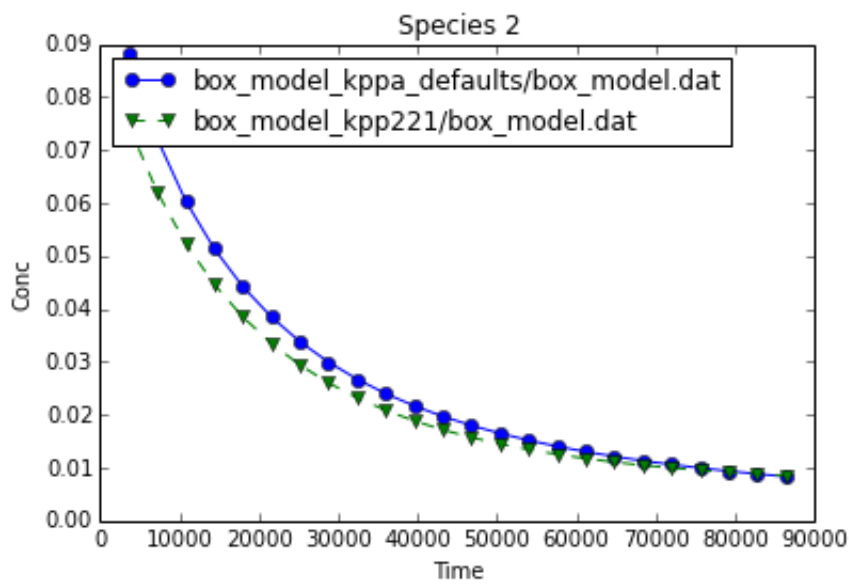
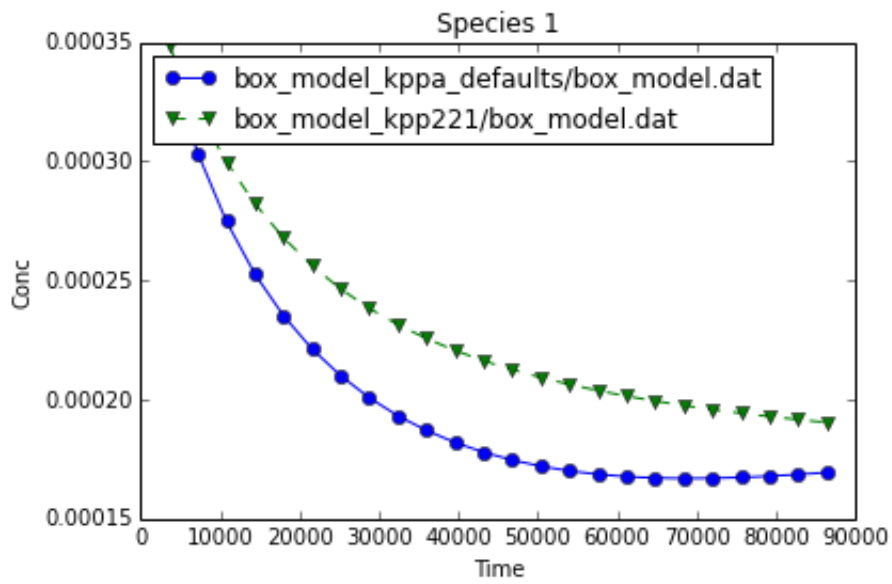
```

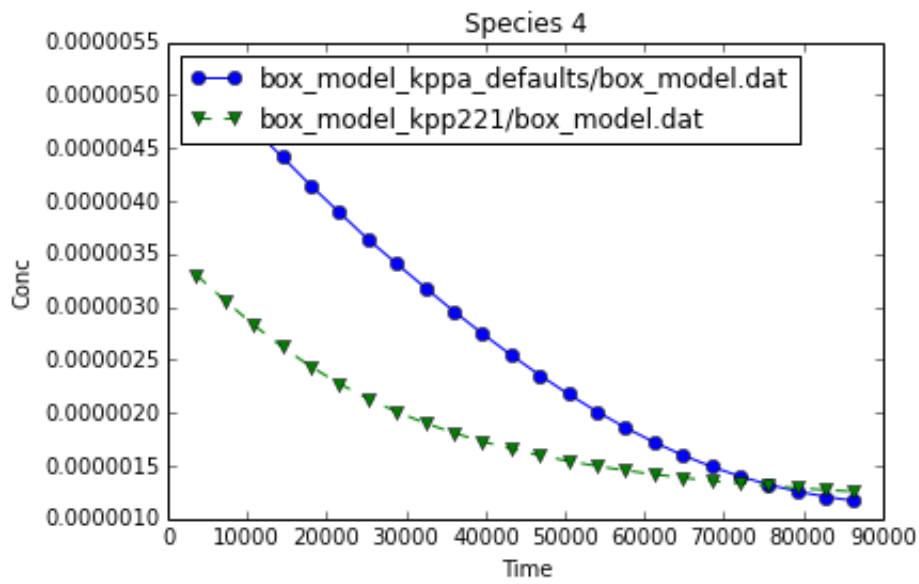
## Results: Concentration vs. Time

The following charts show concentrations from both files vs. time. The species name is not recorded in the .dat file, but you can replace None in the plot\_dat call with an ordered list of names to set the plot titles.

```
In [24]: plot_dat([kppa_dat, kpp_dat], names=[kppa_file, kpp_file], titles=None)
```



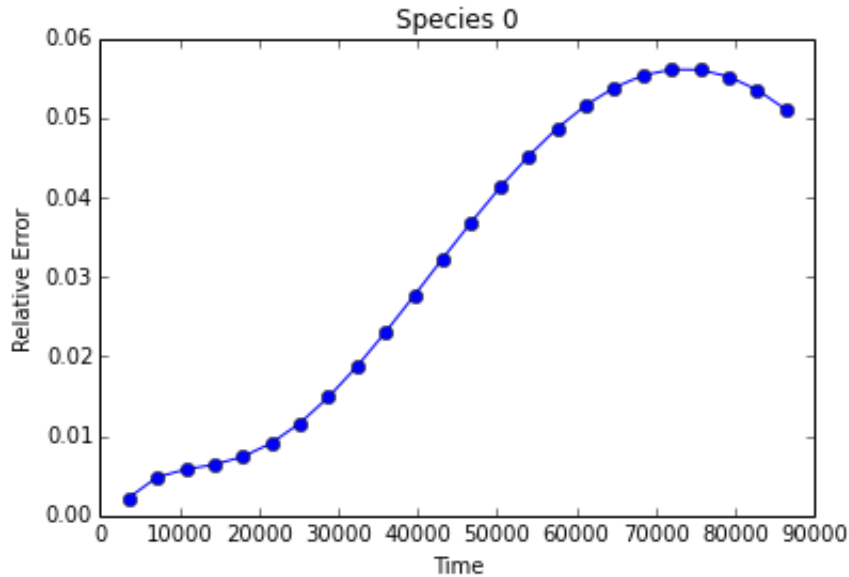




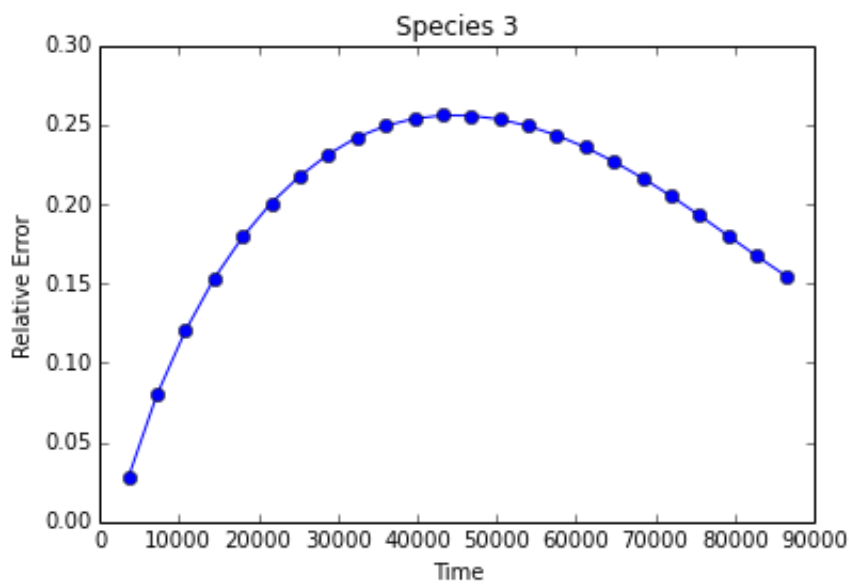
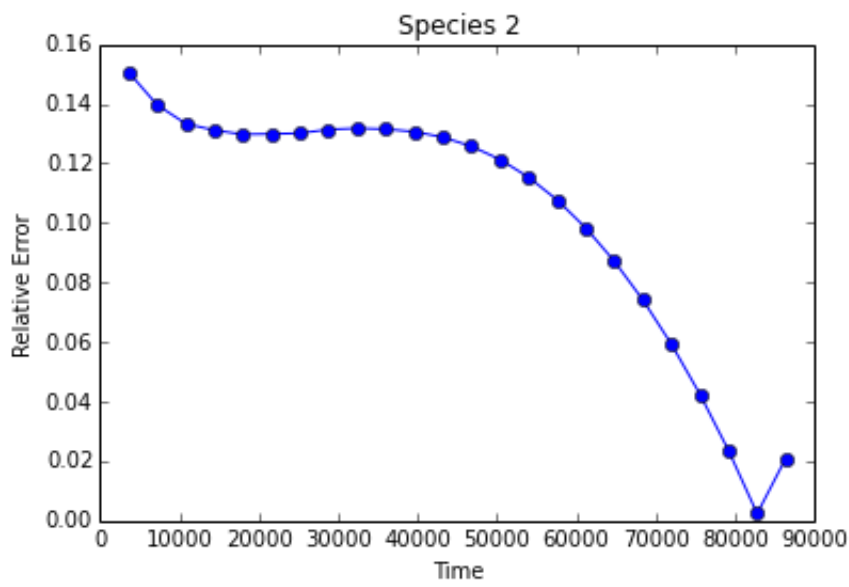
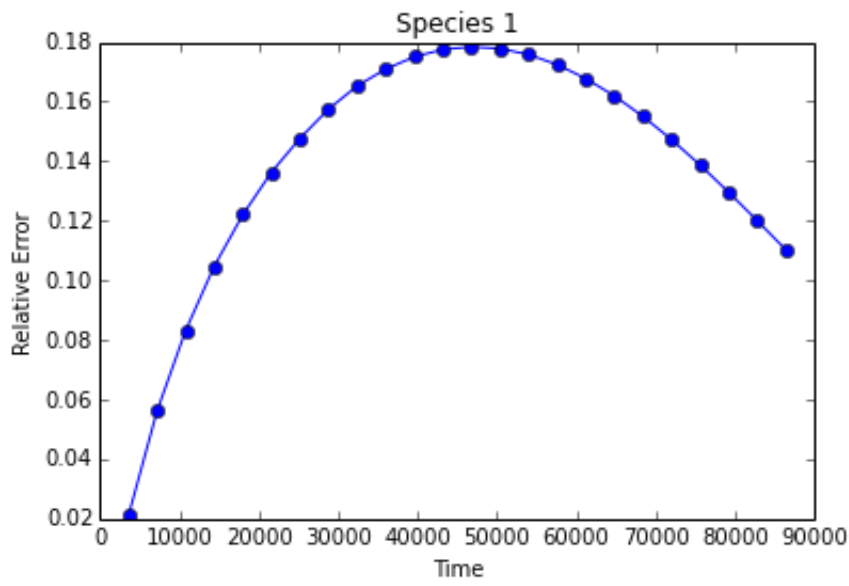
## Results: Error vs. Time

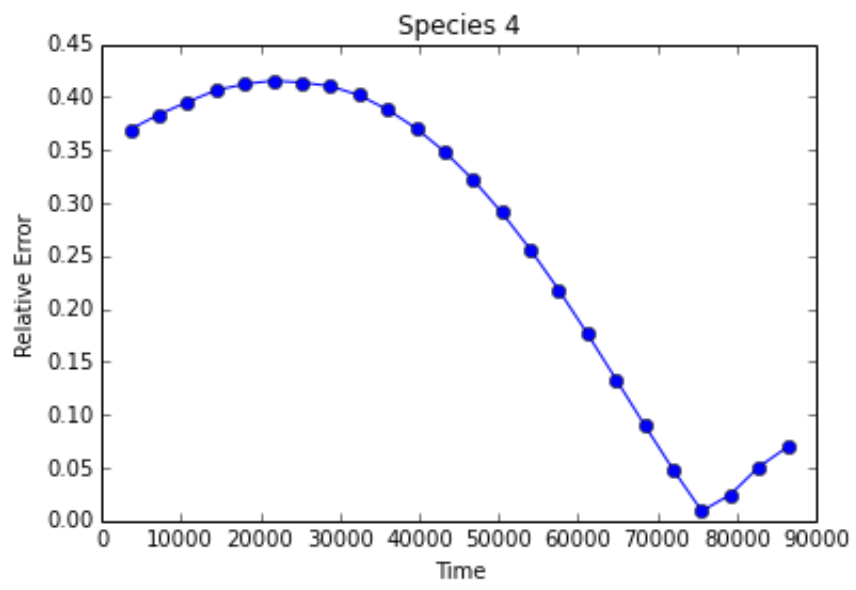
The following charts show relative error between the two data files vs. time.

```
In [25]: plot_dat([err_dat], ylabel='Relative Error')
```









In [25]: