Problem 1 : Solving systems of ODEs

You are given an Isothermal Plug Flow Reactor in which components A and C are fed in equimolar amounts, and the follow- ing reaction takes place $2A \rightarrow B$, with C as a third non-reacting component. You assume that the reaction takes place in the liquid phase and that the volumetric flow rate remains constant even when reaction occurs.

The equations for all three species in the plug flow reactor are:

$$u\frac{dC_i}{dz} = r_j, \text{j} = 1,...,3$$

C_A (0) = 2 kmol/m3, C_B (0) = 0, C_C (0) = 2 kmol/m^3 and we take u = 0.5 m/s, k = 0.3 m^3/kmol s, and the total reactor length as z = 2.4 m.

1. Find and plot:

$$C_A(z), C_B(z), C_C(z)$$

Problem 2: Solving algebraic equations

The heat of reaction for a certain reaction is given by
$$\Delta H_r^o(T) = -24097 - 0.26 \cdot T + 1.69 \cdot 10^{-3}T^2 + 1.5 \cdot 10^5/T \text{ cal/mol.}$$

1. Compute the temperature at which $\Delta H_r^o(T) = -23505$ cal/mol.
2. For different temperatures plot a graph of $\Delta H_r^o(T)$

use the temperature range from 50 to 999 (you can assume the units to be ^oC

Problem 3 (Bonus)

Solve $\frac{dy}{dt} = f(y,t)$, $y(0) = y_0$, with N_points=40 steps until t=time_finish=10, $y_0 = 1$.

$$f(y,t) = -2 \cdot y$$

1. Calculate and plot the analytical solution to this ODE (with black solid line 'k-')
2. Solve the ODE using the simple Forward Euler scheme. Plot it using red dashed line ('r--')

```
t[k+1] = t[k] + dt
 y[k+1] = y[k] + dt*f(y[k], t[k])
```

   Comment on the accuracy of the simple Forward Euler scheme
3. Solve the equation using Runge-Kutta method. Plot the solution.

```
t[k+1] = t[k] + dt
    y_star = y[k] + dt*f(y[k], t[k])
    y[k+1] = y[k] + 0.5*dt*f(y[k], t[k]) + 0.5*dt*f(y_star, t[k+1])
```

Comment on the accuracy of the Runge-Kutta method.

Problem 4: Data regression

1. Fit the data using $y(x) = A \cdot x^\alpha$.
2. Print the parameters A, α that provide the best fit.
3. Plot the data points with circles and the fitted curve with a solid line
4. Calculate the χ^2 which is the squared difference between each data point and the predicted curve:

$$\chi^2 = \Sigma(data[i] - FittedCurve[i])^2$$

```
xdata = np.array([ 0.          ,   0.34482759,  0.68965517,  1.03448276,  1.37931
034,
        1.72413793,  2.06896552,  2.4137931 ,  2.75862069,  3.10344828,
        3.44827586,  3.79310345,  4.13793103,  4.48275862,  4.82758621,
        5.17241379,  5.51724138,  5.86206897,  6.20689655,  6.55172414,
        6.89655172,  7.24137931,  7.5862069 ,  7.93103448,  8.27586207,
        8.62068966,  8.96551724,  9.31034483,  9.65517241, 10.          ])
ydata = np.array([  167.2725507 ,    106.16914857,     41.10677407,     32.463013
12,
         275.03411655,    -39.24097532,     38.94035796,    226.68658612,
         -14.49192858,    327.88376208,    275.78034462,    951.86094163,
         756.67008989,    791.13135981,   1006.52714599,   1454.26168144,
        1732.68330785,   1721.27067228,   2368.93945747,   2670.11523175,
        3501.20879412,   3799.49281404,   4578.21264294,   5083.0998124 ,
        5586.56113083,   6608.43402806,   7060.16302214,   8208.65917808,
        9045.44567227, 10040.46032897])
```

```
In [5]: from scipy.optimize import curve_fit
        xdata = np.array([ 0.        ,  0.34482759,  0.68965517,  1.03448276,  1.3
        7931034,
                1.72413793,  2.06896552,  2.4137931 ,  2.75862069,  3.10344828,
                3.44827586,  3.79310345,  4.13793103,  4.48275862,  4.82758621,
                5.17241379,  5.51724138,  5.86206897,  6.20689655,  6.55172414,
                6.89655172,  7.24137931,  7.5862069 ,  7.93103448,  8.27586207,
                8.62068966,  8.96551724,  9.31034483,  9.65517241, 10.        ])
        ydata = np.array([  167.2725507 ,   106.16914857,    41.10677407,    32.46
        301312,
                275.03411655,   -39.24097532,    38.94035796,   226.68658612,
                -14.49192858,   327.88376208,   275.78034462,   951.86094163,
                756.67008989,   791.13135981,  1006.52714599,  1454.26168144,
                1732.68330785,  1721.27067228,  2368.93945747,  2670.11523175,
                3501.20879412,  3799.49281404,  4578.21264294,  5083.0998124 ,
                5586.56113083,  6608.43402806,  7060.16302214,  8208.65917808,
                9045.44567227, 10040.46032897])
```