2 Problem set #2: ODE's

Seriously Suggested Reading Numerical Recipes, 2nd Edition chapter 16 (you might also want to look at Chapters 17). The relevant chapters in the 1st edition are 15 and 16. You might also want to look at `http://www.ldeo.columbia.edu/~mspieg/UserCalc/` as an example of more complex U-series decay problems.

Purpose and preliminaries The purpose of this problem set is to give you some hands-on-experience with both analytic and numerical solutions of simple (but useful) systems of ordinary differential equations stemming from the problem of radioactive decay chains (e.g. the U-series). This entire problem set can be done using Matlab, however, I provide a set of example codes in matlab, C, and Fortran for solving various problems. These can be downloaded as a gzipped tar file from `http://www.ldeo.columbia.edu/~mspieg/mmm/probset2.tar.gz` Use these as templates and working examples but it probably doesn't hurt to try and write the simplest problems from scratch.

The Problem: U-series decay chains

Consider a decay chain of radioactive nuclides $c_1 \rightarrow c_2 \rightarrow c_3 \ldots$ where component c_1 decays to c_2 which decays to c_3 etc. (For example ^{238}U $\rightarrow$ ^{230}Th $\rightarrow$ ^{226}Ra). In a closed system, we can write the behaviour of the concentrations as a coupled system of first order ODE's as

$$\begin{aligned}
\frac{dc_1}{dt} &= -\lambda_1 c_1 \\
\frac{dc_2}{dt} &= \lambda_1 c_1 - \lambda_2 c_2 \\
\frac{dc_3}{dt} &= \lambda_2 c_2 - \lambda_3 c_3 \\
&\vdots \\
\frac{dc_n}{dt} &= \lambda_{n-1} c_{n-1} - \lambda_n c_n
\end{aligned}$$

where $\lambda_i = \ln 2 / t_{1/2_i}$ is the decay constant for nuclide i with half-life $t_{1/2_i}$. In the case of the ^{238}U decay series, the half lives of ^{238}U, ^{230}Th, and ^{226}Ra are 4.468×10^9, 75,380 and 1600 years respectively. We now want to solve this system analytically and numerically and compare. Do the following:

1. Write the 3-nuclide problem in matrix-vector form as

$$\frac{d\mathbf{c}}{dt} = A\mathbf{c}$$

and find a general solution for initial condition $\mathbf{c}_0$ in terms of the eigenvalues and eigenvectors of A. Find the eigenvalues and eigenvectors of A numerically using matlab (*extra credit*: find them analytically and compare... it's not hard).

2. Scaling time! It turns out that solving for the concentrations directly is not the best idea numerically, because the half-lives are so different. It is much better to solve for the *activities* of each nuclide $a_i = \lambda_i c_i$ which is just the rate of decay.

   (a) First rewrite the problem in terms of the activities as

   $$\frac{d\mathbf{a}}{dt} = A'\mathbf{a}$$

   and find A'

   (b) Now rescale this new problem such that a dimensionless time of $t' = 1$ corresponds to the time it takes ^{226}Ra to decay by a factor of e (i.e. scale to the decay rate of the fastest decaying species). What is the new matrix?

3. Using the analytic eigenvector solution, solve this system for $\mathbf{a}(t)$ after 50,000, 100,000, 1 Million years for initial activities $\mathbf{a}_0 = (1, 2, 3)$. What is the solution as $t \to \infty$? (this is a state known as secular equilibrium). Roughly how long does it take this system to attain secular equilibrium?

4. Now solve this problem numerically using `ode45` and compare to the analytic solution. (you can use the solve_lorenz.m as a template program). Play around with the relative and absolute tolerances. What do these need to be to get a true error less than 10^{-6} after 50,000 years?

5. Stiff equation time... The actual decay chain from ^{238}U to ^{226}Ra, is really $^{238}\mathrm{U} \to ^{234}\mathrm{Th} \to ^{234}\mathrm{Pa} \to ^{234}\mathrm{U} \to ^{230}\mathrm{Th} \to ^{226}\mathrm{Ra}$ where the half-lives of ^{234}Th, ^{234}Pa and ^{234}U are 24.1 *days*, 6.7 *hours*, and 245,500 years respectively. This is a desperately stiff system.

   (a) Crudely estimate the number of time steps an explicit stepper like ODE45 would take to integrate the full problem for 50,000 years.

   (b) Test your idea using ODE45 (but you might not want to run the entire problem...). (hint: to watch your poor integrator go (or go nowhere), use the options 'OutputFcn','odeplot' in odeset. see `help odeset` in matlab.

   (c) Choose one or more of the stiff solvers in matlab and try again. Can you get the answers to agree with the simpler system after 50,000 years for an initial condition $\mathbf{a}_0 = (1, 2, 3, 1, 2, 3)$.