Atmospheric box models (part 2)

1. Solving a box model numerically
2. Euler’s method - forward
3. Mid-point method
4. Runge-Kutta 4th order method
5. Euler’s method - backward
1. Numerical solutions

Methods for approximating exact solutions

Useful for real-world simulations

eg. changes in emissions, for example $S = f(t)$
changes in loss rate constants, for example $k = f(t)$
coupled reactions, complex kinetics

A valuable resource:

Numerical Recipes:
The art of scientific computing
Press et al.

Chapter 16. Integration of ordinary differential equations

http://www.nr.com/oldverswitcher.html
2. Euler’s method – forward (explicit)

mass balance equation: \[ \frac{\partial m_x}{\partial t} = S - L = S - km_x \]

let \( \Delta t \) be a fixed time step

\[ m_{x_{t_n+1}} = m_{x_{t_n}} + \left( \frac{\partial m_x}{\partial t} \right)_{t_n, m_{x_{t_n}}} \cdot \Delta t \]

or

\[ m_{x_{n+1}} = m_{x_n} + \frac{\partial m_x}{\partial t} \cdot \Delta t \]

uses derivative at initial point only

requires small time steps

always check that result is independent of time step
3. Midpoint method

\[
\frac{\partial m_x}{\partial t} = S - L = S - km_x
\]

full time step at \( t_n \) :

\[
k_1 = \left( \frac{\partial m_x}{\partial t} \right)_{t_n, m_{x,t_n}} \cdot \Delta t
\]

full time step at mid-point:

\[
k_2 = \left( \frac{\partial m_x}{\partial t} \right)_{t_n+\Delta t/2, m_{x,t_n+k_1/2}} \cdot \Delta t
\]

final result:

\[
m_{x,t_{n+1}} = m_{x,t_n} + k_2
\]
4. Runge-Kutta, 4th order

calculate \( k_\# = \left( \frac{\partial m_x}{\partial t} \right) \cdot \Delta t \)

at each of the following:

step 1: \( t_n, m_{x_n} \)

step 2: \( t_n + \Delta t/2, m_{x_n} + k_1/2 \)

step 3: \( t_n + \Delta t/2, m_{x_n} + k_2/2 \)

step 4: \( t_n + \Delta t, m_{x_n} + k_3 \)

final result: \( m_{x_{t_n+1}} = m_{x_{t_n}} + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} \)
2. Euler’s method – backward (implicit)

mass balance equation: \[ \frac{\partial m_x}{\partial t} = S - L = S - km_x \]

time step = \( \Delta t \)

\[ m_{x_{t_1}} = m_{x_{t_2}} - \Delta t \cdot \left( \frac{\partial m_x}{\partial t} \right)_{t_2, m_{x_{t_2}}} \]

\[ m_{x_{t_1}} = m_{x_{t_2}} - \Delta t \cdot (S - km_{x_{t_2}}) \]

\[ m_{x_{t_1}} = m_{x_{t_2}} (1 - k\Delta t) - S\Delta t \]

\[ m_{x_{t_2}} = \frac{m_{x_{t_1}} + S\Delta t}{1 - k\Delta t} \]

Conundrum - we need to know the answer first, in order to evaluate the derivative!
5. Model performance issues ...

1. accuracy
2. stability
3. computational efficiency (speed)
4. “stiff” equations (a problem with very different time scales)