Errors

What are possible sources for errors in computed solutions?

- Rounding error (finite digit limitation in computer arithmetic)
- Accumulation error (rounding errors accumulated when additional calculations are done)
- Truncation error (termination of an infinite process)

  $\lim_{n\to\infty} x_n = x$. Approximate x with x_N. TE = $|x - x_N|$
- Discretization error (discrete approximation to continuous systems)

Floating Point Number & Error

Computers use a finite subset of the rational numbers to approximate any real number

Let x be any real number.

Infinite decimal expansion : $x \;=\; \pm .x_1 x_2 \cdots x_d \cdots 10^e$

Truncated floating point number : $x \;\approx fl(x) = \;\pm .x_1 x_2 \cdots x_d 10^e$

where $x_1 \neq 0, 0 \leq x_i \leq 9,$

d : an integer, precision of the floating point system

e : an bounded integer

Floating point or **roundoff error** : $fl(x) - x$

Error Propagation

When additional calculations are done, there is an accumulation of these floating point errors.

Example: Let $x = -0.6667$ and $fl(x) = -0.667\ 10^0$ where $d = 3$.

$$\text{Floating point error: } fl(x) - x = ?$$

$$\text{Error propagation: } fl(x)^2 - x^2 = ?$$

Error Propagation

When additional calculations are done, there is an accumulation of these floating point errors.

Example : Let $x = -0.6667$ and $fl(x) = -0.667\ 10^0$ where $d = 3$.

Floating point error : $fl(x) - x = -0.0003$

Error propagation : $fl(x)^2 - x^2 = 0.00040011$

Accumulation Error

Let $U_0 = fl(u_0)$, $A = fl(a)$, and $B = fl(b)$ so that

$$U_k = AU_{k-1} + B + \overline{R}_k = aU_{k-1} + b + R_k$$

where $\overline{R}_k$ is the roundoff error at k-th step

and R_k includes $\overline{R}_k$ and round errors assoicated with a and b.

Accumulation Error: $U_k - u_k$

Accumulation Error

- Can we guarantee that the accumulation error can keep reasonably small as the number of time steps increases?
- Can we provide a rough estimate on the accumulation error?

Accumulation Error

Accumulation Error Theorem

Consider the first order finite difference algorithm.

If $r = |a| < 1$ and the roundoff errors are uniformly bounded, *i.e.*,

$$\left|\overline{R}_k\right| \le R < \infty,$$

then the accumulated error is uniformly bounded, *i.e.*,

$$|U_k - u_k| \le r^k |U_0 - u_0| + R\frac{1-r^k}{1-r} \le \left(1 + \frac{1}{1-r}\right)R.$$

Summary on Cooling of Coffee

Model

Based on the discrete Newton cooling Law,

$$u_{k+1} = au_k + b$$

Steady State Theorem:

if $|a| < 1$, then

$u_{k+1} \to u$, where $u = au + b$ (i.e., $u = u_{sur}$)

Accumulation Error Theorem:

if $|a| < 1 \,\&\, |\overline{R}_{k+1}| \le R < \infty$, then

$|U_{k+1} - u_{k+1}| \le M < \infty$, where M independen t of k

Finally, What-if thinking

For example, what if the coffee is not well-stirred?

Finally, What-if thinking

What if $\Delta t \to 0$?

$$
\begin{aligned}
u_{k+1} - u_k &= c\Delta t\left(u_{sur} - u_k\right) \\
\frac{u_{k+1} - u_k}{\Delta t} &= c\left(u_{sur} - u_k\right)
\end{aligned}
$$

Finally, What-if thinking

Discrete Model:

$$u_{k+1} - u_k = c\Delta t\left(u_{sur} - u_k\right)$$

$$\frac{u_{k+1} - u_k}{\Delta t} = c\left(u_{sur} - u_k\right)$$

$$\Rightarrow$$

Continuous Model:

$$\text{As } \Delta t \to 0, \quad \frac{u_{k+1} - u_k}{\Delta t} \to \frac{du}{dt}$$

$$\therefore \quad \frac{du}{dt} = c\left(u_{sur} - u\right)$$

Case 1 More on Cooling Coffee

Discrete Time-Space Models: Convergence Analysis

YORK
UNIVERSITÉ
UNIVERSITY
redefine THE POSSIBLE.

Initial Value Problem

Continuous Model

$$\begin{cases} \dfrac{du}{dt} = c\left(u_{sur} - u(t)\right) \\ u(0) = u_0 \end{cases}$$

Actually, we can solve it exactly

$$u(t) = u_{sur} + \left(u(0) - u_{sur}\right)e^{-ct}$$

Forward Euler's Method

Let $u = u(t)$. The general form of an initial value problem is

$$\begin{cases} u_t = f(t,u) \\ u(0) = u_0 \text{ is given} \end{cases}$$

In general, we may not be able to solve it exactly.

Therefore, we approximate $u_t \approx \dfrac{u_{k+1} - u_k}{\Delta t}$ and then solve u discretely

where $u_k \approx u(k\Delta t)$:

$$\boxed{\text{Forward Euler's Method:} \quad \frac{u_{k+1} - u_k}{\Delta t} = f(k\Delta t, u_k)}$$

Euler approximations with size of Δt

Applying Forward Euler (or Euler) Method to the cooling of coffee problem as an example:

$$\frac{u_{k+1} - u_k}{\Delta t} = c\left(u_{sur} - u_k\right)$$

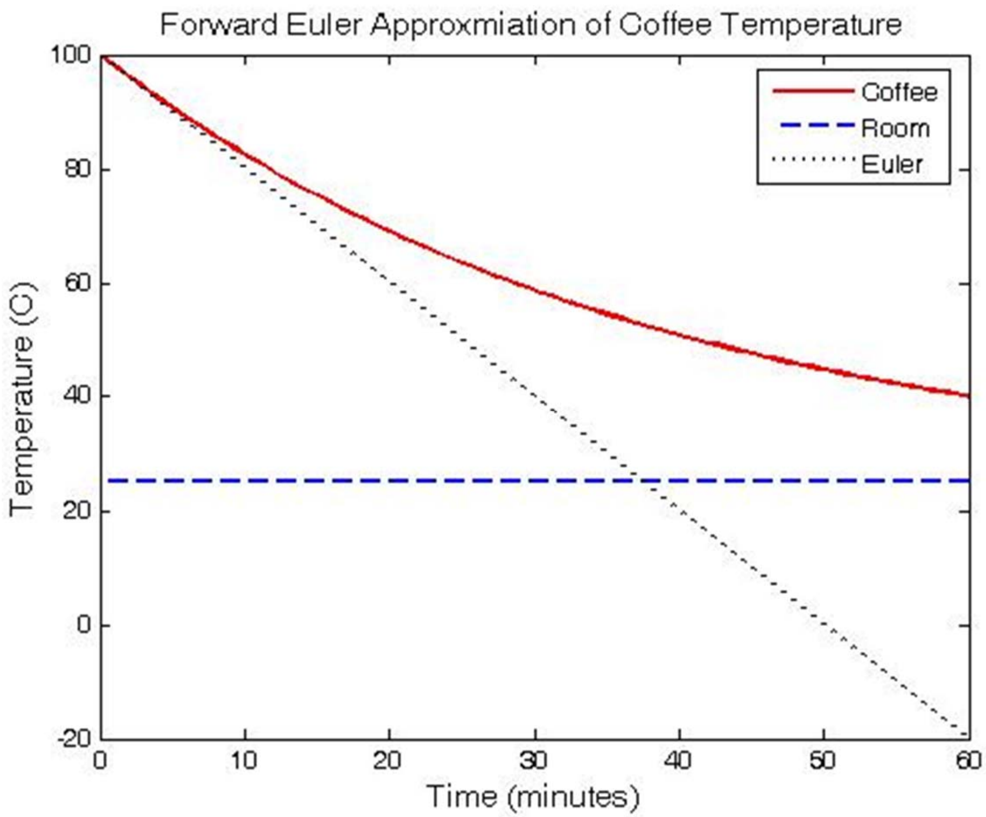
$$\therefore \qquad u_{k+1} = a\,u_k + b$$

where $a = 1 - c\Delta t$ and $b = c\,\Delta t\,u_{sur}$

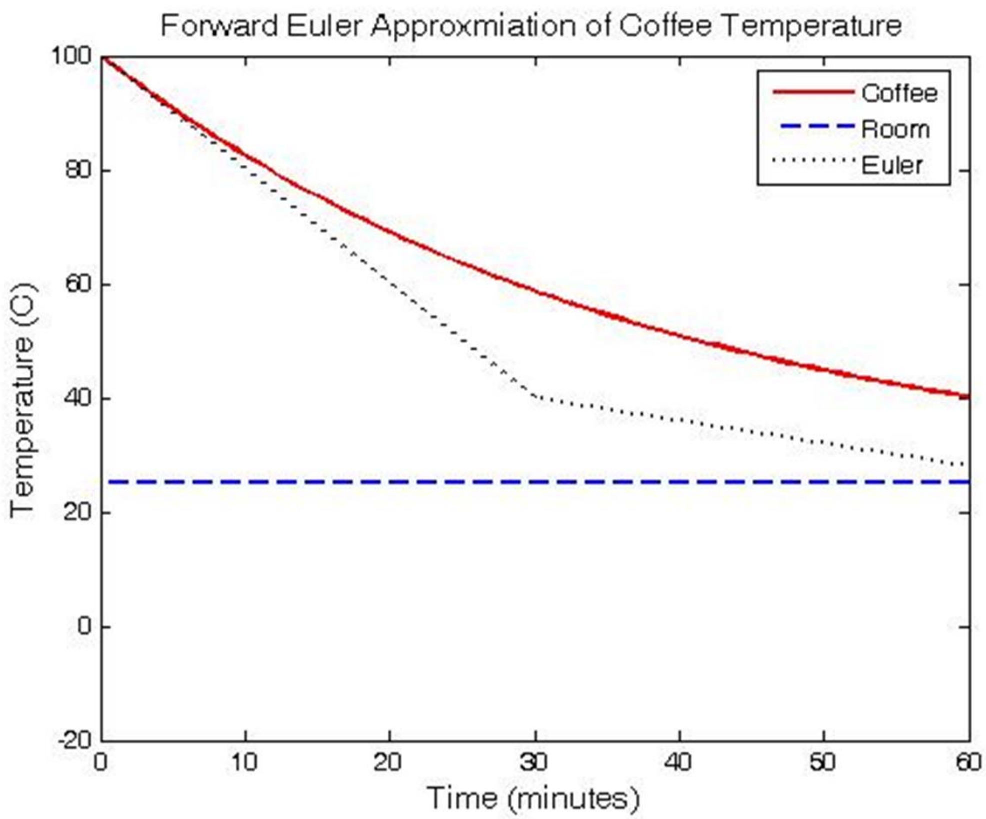
Consider to estimate $u(t)$ at $t = 60$ minutes.

Fix all the parameters except Δt. We change the time step size $\Delta t = 60,\ 30,\ 15, 1$.

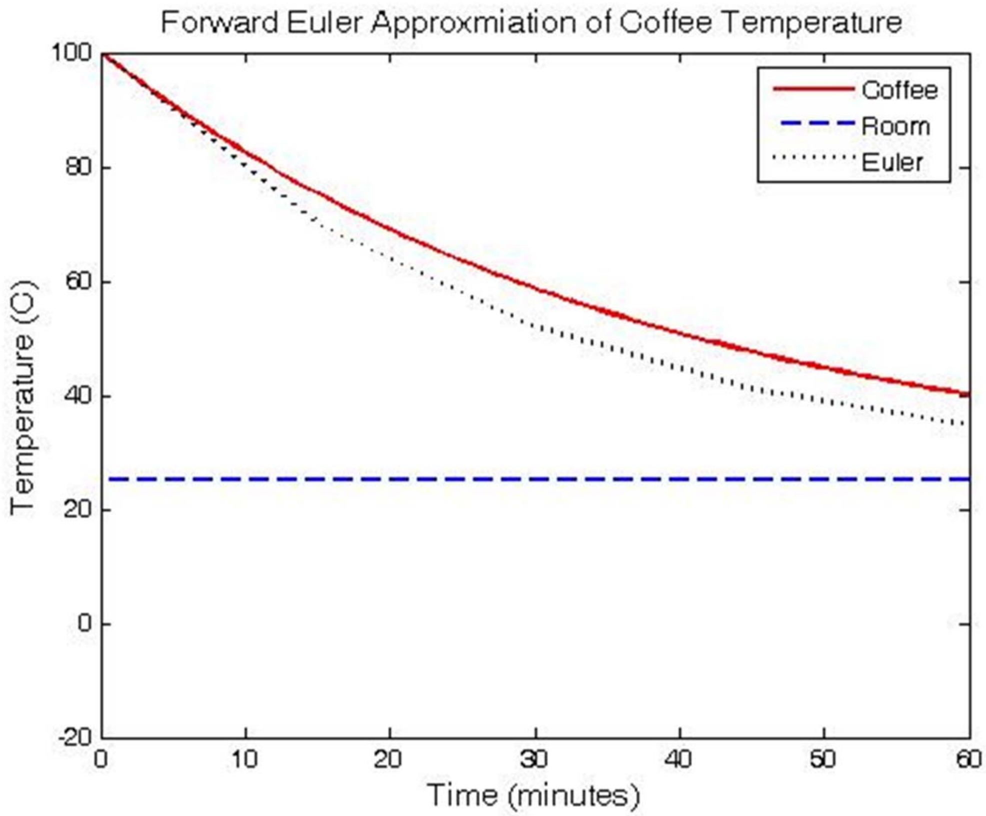
Euler Approximation with Δt=60 minutes



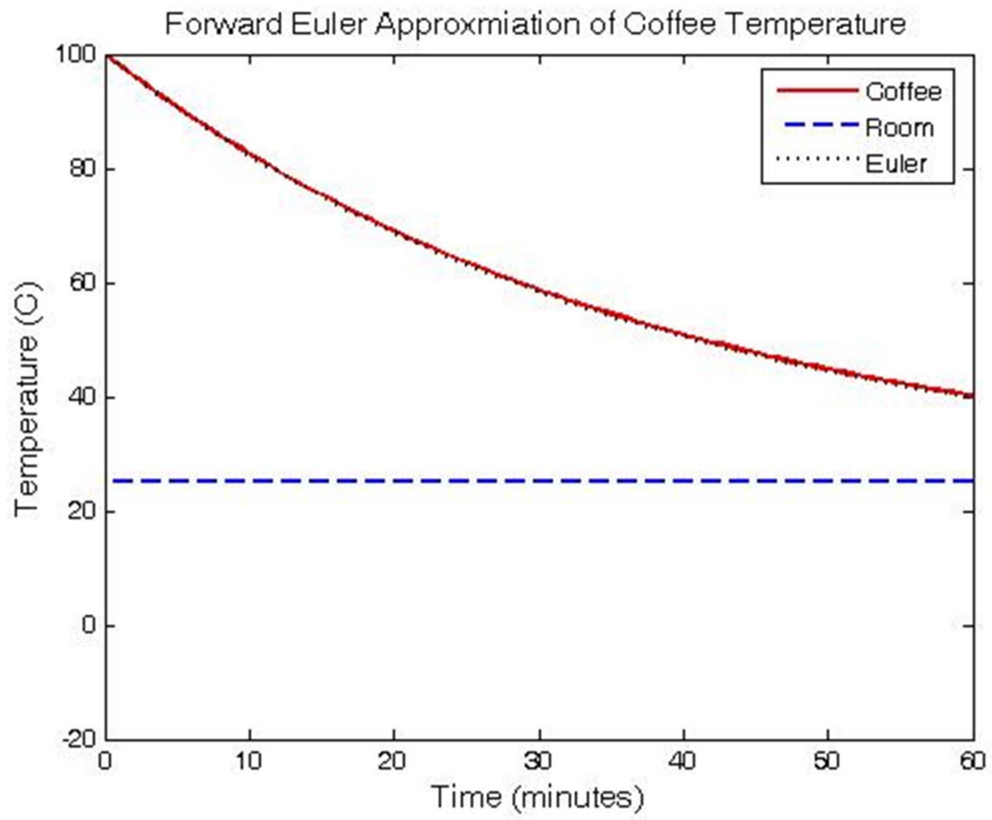
Euler Approximation with Δt=30 minutes



Euler Approximation with Δt=15 minutes



Euler Approximation with Δt=1 minutes



Euler approximations with size of Δt

Consider Euler approximation error at a time $t = 60$ minutes.

We change the time step size $\Delta t = 60,\ 30,\ 15, 1$ minutes, respectively

Δt	Exact (c)	Forward Euler (c)	Abs Error (c)
60	40.14	-20.00	60.14
30	40.14	28	12.14
15	40.14	34.72	5.42
1	40.14	39.82	0.33

Euler approximations with size of Δt

- What can we conclude as Δt becomes smaller?

$$\text{As } \Delta t \ \downarrow, \quad \text{the error} \downarrow$$

- How is the numerical error exactly related to the size of Δt ?

Assessment: Numerical error

Two major types of numerical errors existing for numerical solutions of an initial value problem:

a) Accumulation error $\equiv E_r^k = R_k = U_k - u_k$

b) Discretization error $\equiv E_d^k = u_k - u(k\,\Delta t)$

where $u(k\Delta t)$: exact continous solution

u_k : from Euler's algorithm with no roundoff error

U_k : from Euler's algorithm with roundoff error

$\therefore$ Overall error $\equiv E_r^k + E_d^k = U_k - u(k\Delta t)$

Assessment: Numerical error

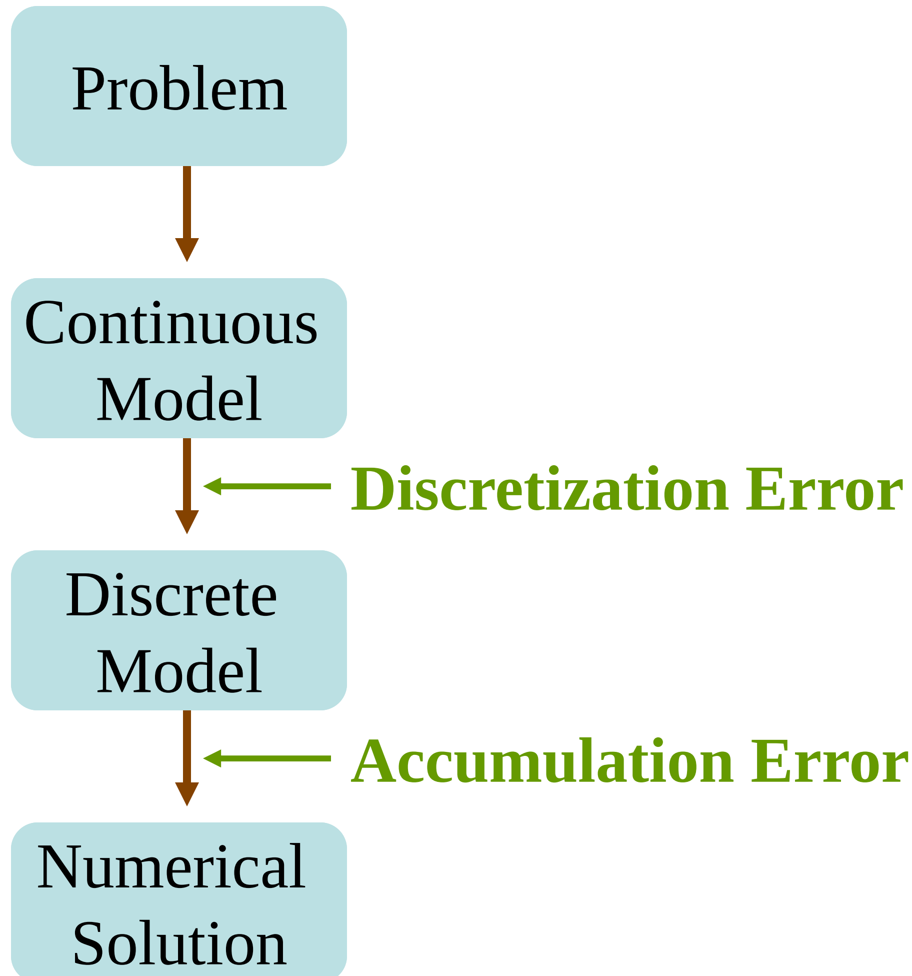
Let $\quad u$: exact continous solution

$\quad U$: an approximation to u

Absolute error $\equiv |U-u|$

Relative error $\equiv |U-u|/u, \text{ if } u \neq 0$

Assessment: Computational error



Accumulation Error Revisit

Accumulation Error Theorem (Cooling of Coffee)

If $r = |a| < 1$ and the roundoff errors are uniformly bounded,

then $$|U_k - u_k| \le R\left(1 + \frac{1}{1-r}\right).$$

Numerical Stability Condition for Euler Algorithm :

$$|a| < 1 \text{ and } c > 0 \Rightarrow 0 < \Delta t < 2/c \text{ and } c > 0$$

An important restriction on Δt so that the accumulation error won't be out of control!

Assessment: Discretization Error

What is the error when $u_t(k\Delta t) \approx \dfrac{u((k+1)\Delta t) - u(k\Delta t)}{\Delta t}$?

Reminder:

Taylor Expansion of a function $u(t)$ about a point $t = a$:

$$f(t) = f(a) + f_t(a)(t-a) + \frac{f_{tt}(\hat{t})}{2}(t-a)^2$$

where $\hat{t}$ is between a and t.

Rewritten the Taylor expansion:

$$\frac{f(t) - f(a)}{(t-a)} = f_t(a) + \frac{f_{tt}(\hat{t})}{2}(t-a)$$

Assessment: Euler Error Theorem

Theorem 1.6.3

Consider the continuous and discrete Newton cooling models.

Assume that the solution have the 1st, 2nd derivatives on $[0, T]$.

If $|u''(t)| \leq M$ and $E_d^0 = 0$ and $a = 1\text{-}c\Delta t > 0$ and $c, \Delta t > 0$, then

$$\left|E_d^{k+1}\right| \leq \frac{M}{2c}\Delta t.$$

Backward Euler's Method

Backward Euler's Method : $\dfrac{u_{k+1} - u_k}{\Delta t} = f\big((k+1)\Delta t, u_{k+1}\big)$

where the evaluation of function f depends on the unknown value u_{k+1}. Sometime, we call it "implicite method".

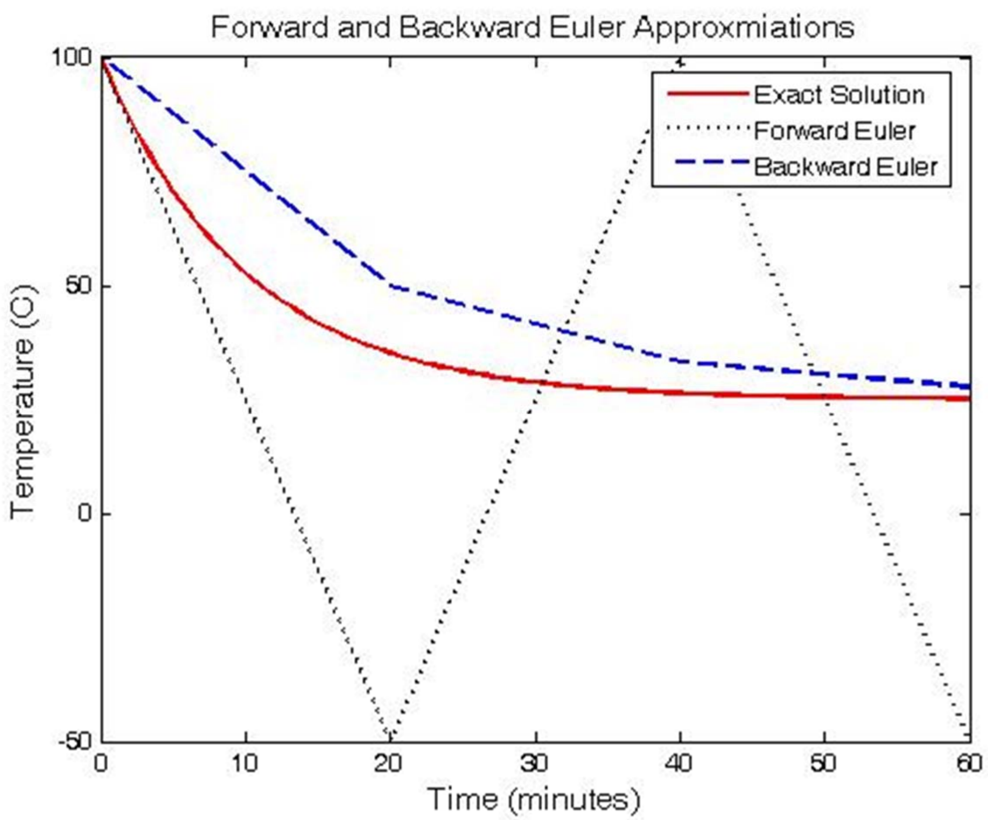
Remarks:

The numerical computation is always stable.

So there is no restriction on Δt!

(Leave as an assignment :-)

Backward Euler Approximation with Δt=20 minutes and a = 1



Backward Euler approximation Error is O(Δt)

Consider Euler approximation error at a time $t = 60$ minutes.

We change the time step size $\Delta t = 60,\ 30,\ 15, 1$ minutes, respectively

Δt	Forward Euler Error	Backward Euler Error
60	60.14	13.70
30	12.14	8.01
15	5.42	4.38
1	0.33	0.32

Improved Euler Method

It can be roughly considered as the average of forward and backward Euler method (Semi - Explicite) :

$$\begin{cases} \dfrac{u_{temp} - u_k}{\Delta t} = f\left(k\Delta t, u_k\right) \\ \dfrac{u_{k+1} - u_k}{\Delta t} = \dfrac{f\left(k\Delta t, u_k\right) + f\left(\left(k+1\right)\Delta t, u_{temp}\right)}{2} \end{cases}$$

Assessment:
Stable and Discretization Error O(Δt^2)

Theorem 1.6.3′

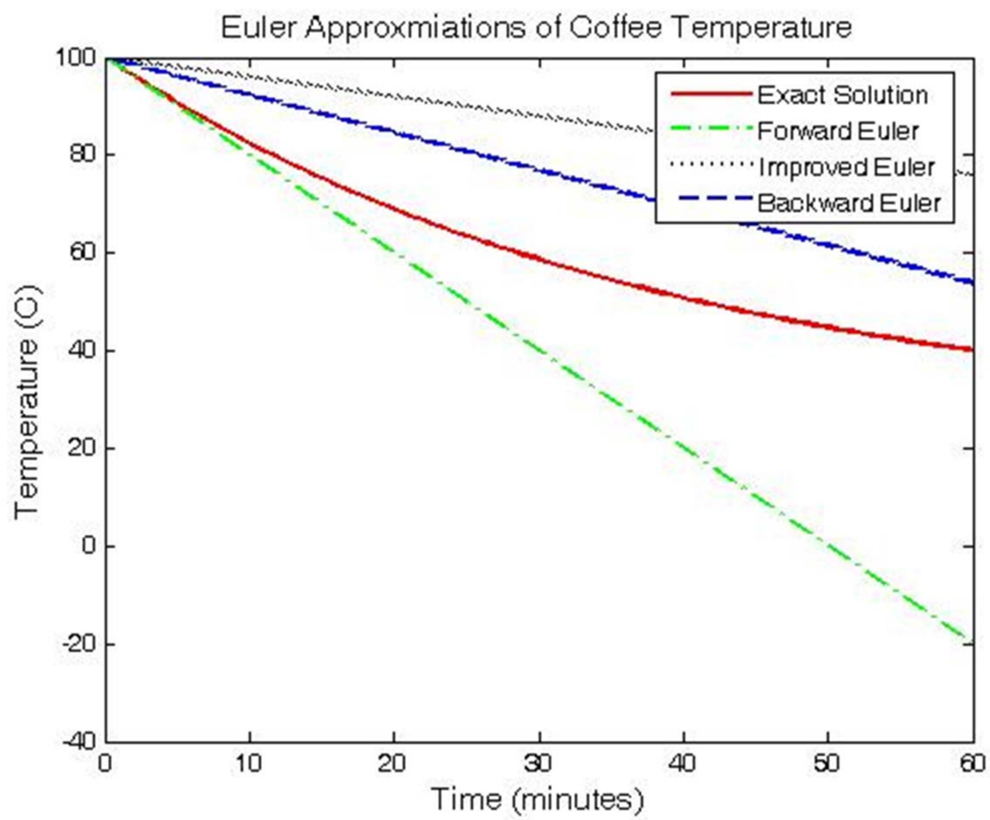
Consider the continuous and discrete Newton cooling models with improved Euler method.

Assume that the solution have the 1st, 2nd, and 3rd derivatives on $[0, T]$.

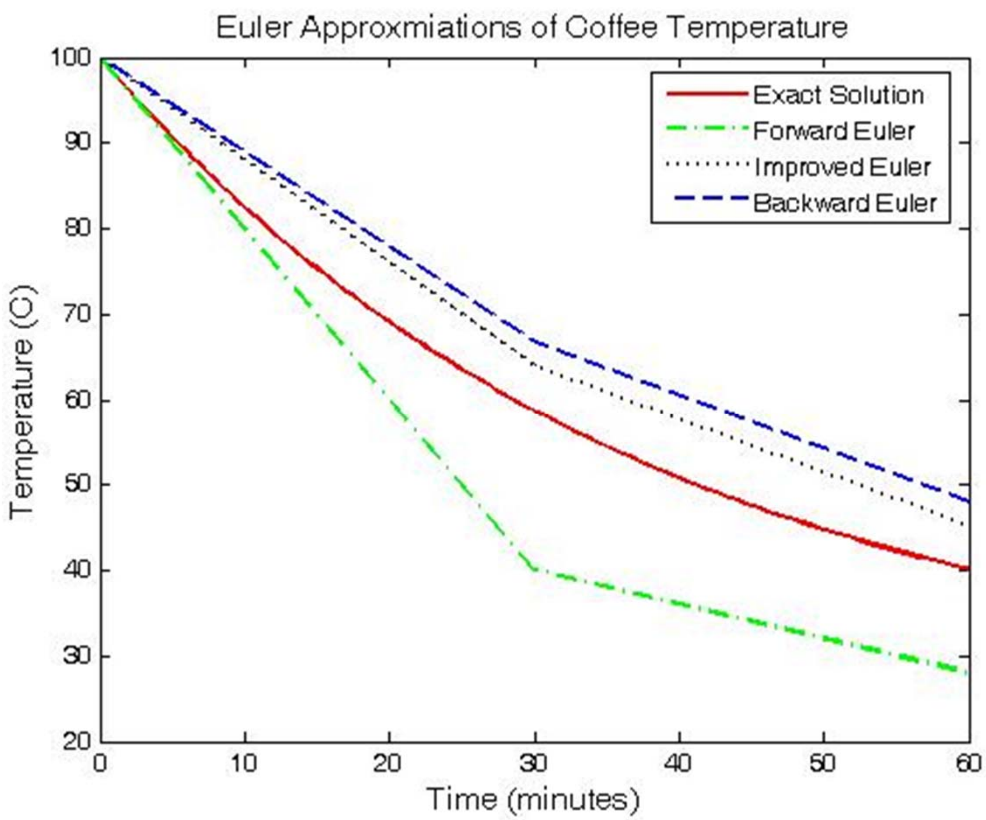
If $\left|u^{(3)}(t)\right| \le M$ and $E_d^0 = 0$ and a = 1-$c\Delta t$>0 and $c, \Delta t$ >0, then

$$\left|E_d^{k+1}\right| \le \frac{M}{3c}(\Delta t)^2.$$

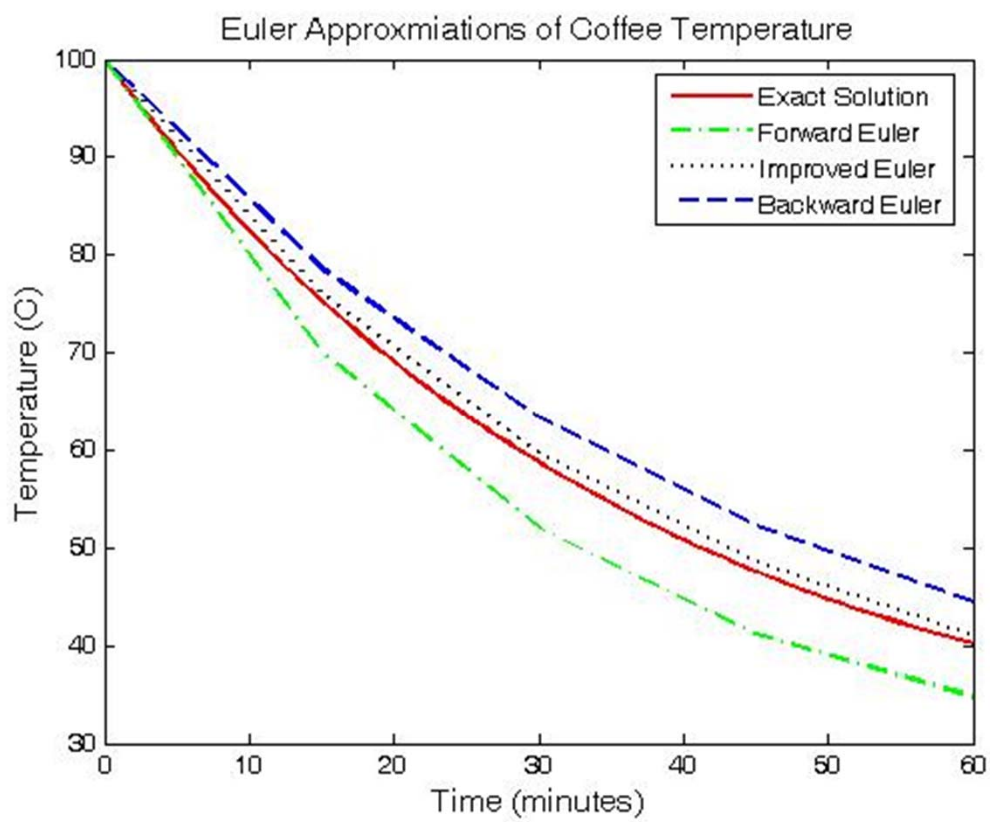
Improved Euler Approximation with Δt=60 minutes



Improved Euler Approximation with Δt=30 minutes



Improved Euler Approximation with Δt=15 minutes



Summary Initial Value Problem

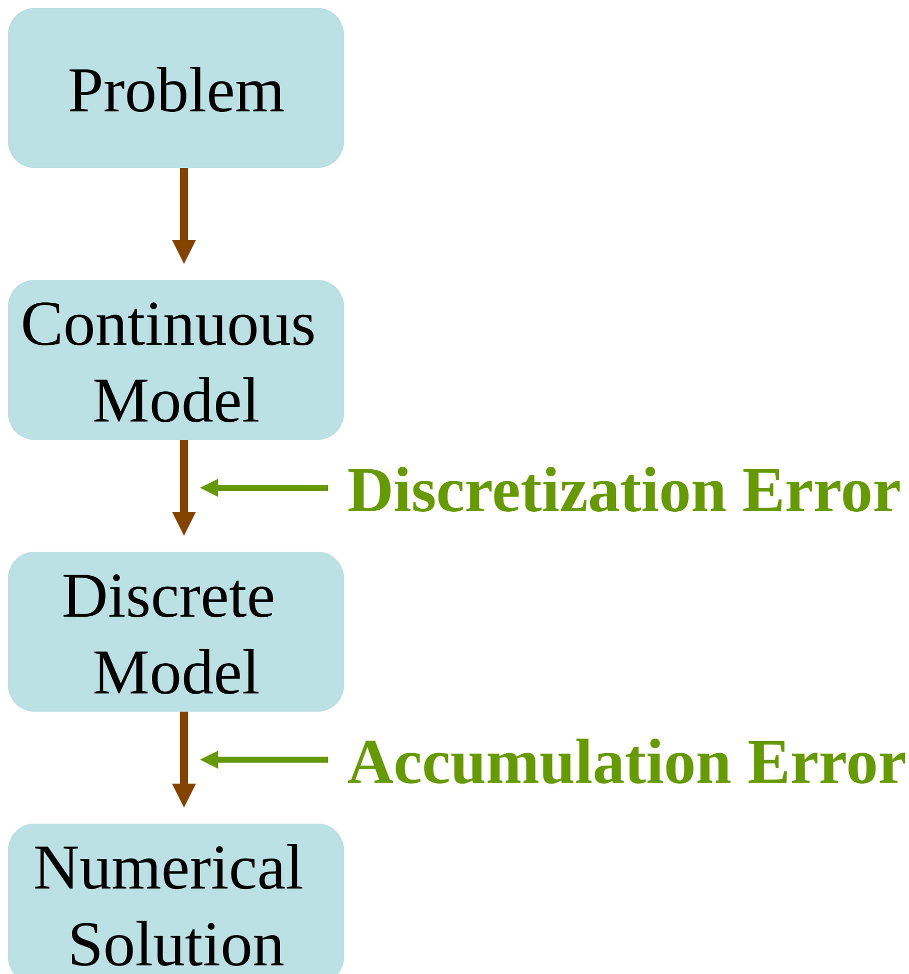
An initial value problem :

$$\begin{cases} u_t = f(t,u) \\ u(0) = u_0 \text{ is given} \end{cases}$$

In general, the close form cannot be found.

Numerical approximation is needed

Assessment: Numerical error



Summary on Numerical errors

There are two types of numerical errors existing for numerical solutions of an initial value problem:

a) Accumulation error $\equiv E_r^k = R_k = U_k - u_k$

b) Discretization error $\equiv E_d^k = u_k - u(k\,\Delta t)$

where $\quad u(k\Delta t)$: exact continous solution

u_k : from Euler's algorithm with no roundoff error

U_k : from Euler's algorithm with roundoff error

$\therefore$ Overall error $\equiv E_r^k + E_d^k = U_k - u(k\Delta t)$

Summary on Euler methods

Forward Euler's Method: $\dfrac{u_{k+1}-u_k}{\Delta t}=f\left(k\Delta t,u_k\right)$

explicit, restriction on Δt, discretization error $O(\Delta t)$

Backward Euler's Method: $\dfrac{u_{k+1}-u_k}{\Delta t}=f\left((k+1)\Delta t,u_{k+1}\right)$

implicit, always stable, discretization error $O(\Delta t)$

Improved Euler's Method (≈ Average of Forward and Backward Euler):

$$\begin{cases} \dfrac{u_{temp}-u_k}{\Delta t}=f\left(k\Delta t,u_k\right) \\ \dfrac{u_{k+1}-u_k}{\Delta t}=\dfrac{f\left(k\Delta t,u_k\right)+f\left((k+1)\Delta t,u_{temp}\right)}{2} \end{cases}$$

explicit, restriction on Δt, discretization error $O\left((\Delta t)^2\right)$

Further Applications-Finance

- The heat equation arises in the modeling of a number of phenomena and is often used in financial mathematics in the modeling of options.
- The famous Black–Scholes option pricing model's differential equation can be transformed into the heat equation allowing relatively easy solutions from a familiar body of mathematics.
- Many of the extensions to the simple option models do not have closed form solutions and thus must be solved numerically to obtain a modeled option price.

Further Applications-Image Processing

- Diffusion problems dealing with Dirichlet, Neumann and Robin boundary conditions have closed form analytic solutions (Thambynayagam 2011)
- The heat equation is also widely used in image analysis (Perona & Malik 1990) and in machine-learning as the driving theory behind scale-space or graph Laplacian methods
- The heat equation can be efficiently solved numerically using the Crank–Nicolson method of (Crank & Nicolson 1947). This method can be extended to many of the models with no closed form solution, see for instance (Wilmott, Howison & Dewynne 1995)