Explicit Methods

Implicit Methods

Newtonian Cooling

Numerical solution to first order ordinary differential equation

$$\frac{dT}{dt} = f(T,t)$$

We can not simply integrate this equation. We have to solve it numerically! First we need to discretise time:

$$t_j = t_0 + j\,dt$$

and for Temperature T

$$T_j = T(t_j)$$

Let us try a forward difference:

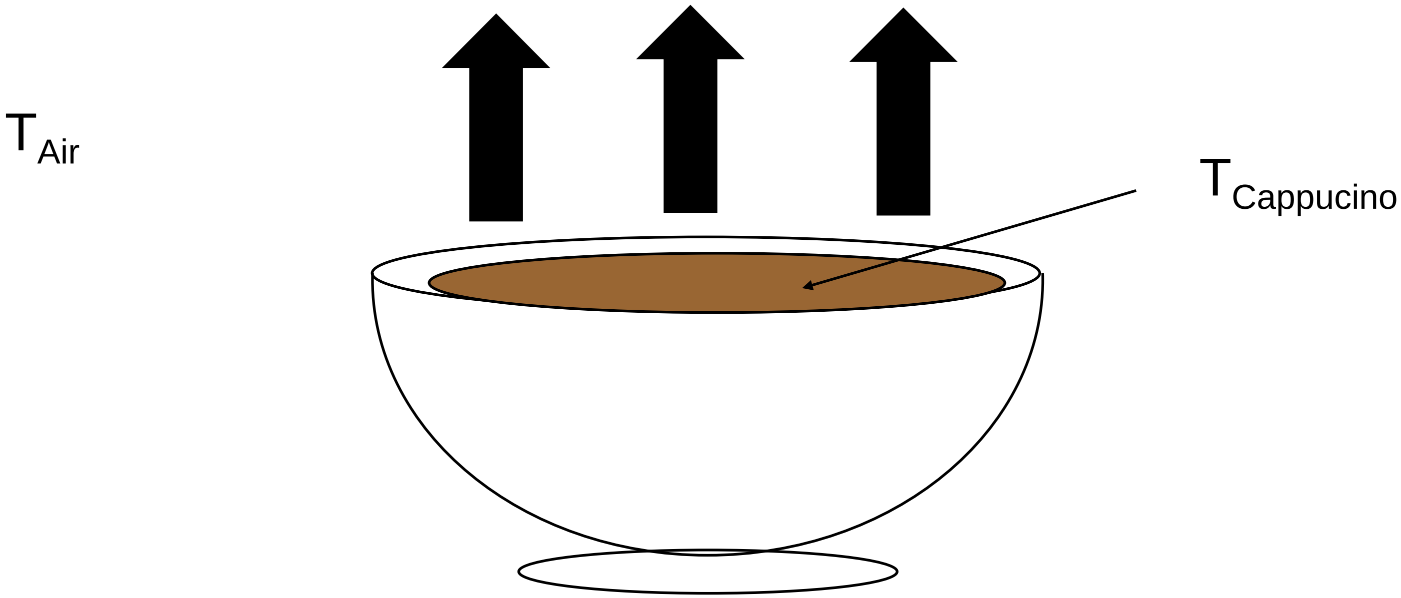
$$\left.\frac{dT}{dt}\right|_{t=t_j} = \frac{T_{j+1} - T_j}{dt} + O(dt)$$

... which leads to the following explicit scheme :

$$T_{j+1} \approx T_j + \mathrm{dt} f\,(T_j, t_j)$$

This allows us to calculate the Temperature T as a function of time and the *forcing* inhomogeneity f(T,t). Note that there will be an error O(dt) which will accumulate over time.

Let's try to apply this to the Newtonian cooling problem:



How does the temperature of the liquid evolve as a function of time and temperature difference to the air?

Newtonian Cooling

The rate of cooling (dT/dt) will depend on the temperature difference (T_{cap}-T_{air}) and some constant (thermal conductivity). This is called **Newtonian Cooling**.

With $T = T_{cap} - T_{air}$ being the temperature difference and τ the time scale of cooling then $f(T,t) = -T/\tau$ and the differential equation describing the system is

$$\frac{dT}{dt} = -T/\tau$$

with initial condition $T=T_i$ at $t=0$ and $\tau>0$.

Newtonian Cooling

This equation has a simple analytical solution:

$$T(t) = T_i \exp(-t/\tau)$$

How good is our finite-difference appoximation?
For what choices of dt will we obtain a stable solution?

Our FD approximation is:

$$T_{j+1} = T_j - \frac{dt}{\tau} T_j = T_j (1 - \frac{dt}{\tau})$$

$$\boxed{T_{j+1} = T_j (1 - \frac{dt}{\tau})}$$

Newtonian Cooling

$$T_{j+1} = T_j (1 - \frac{dt}{\tau})$$

1. Does this equation approximation converge for dt -> 0?
2. Does it behave like the analytical solution?

With the initial condition $T=T_0$ at $t=0$:

$$T_1 = T_0 (1 - \frac{dt}{\tau})$$

$$T_2 = T_1 (1 - \frac{dt}{\tau}) = T_0 (1 - \frac{dt}{\tau})(1 - \frac{dt}{\tau})$$

leading to : $$T_j = T_0 (1 - \frac{dt}{\tau})^j$$

Newtonian Cooling

$$T_j = T_0(1-\frac{dt}{\tau})^j$$

Let us use dt=t_j/j where t_j is the total time up to time step j:

$$T_j = T_0\left(1+\left[-\frac{t}{j\tau}\right]\right)^j$$

This can be expanded using the *binomial theorem*

$$T_j = T_0\left[1^j + 1^{j-1}\left[-\frac{t}{j\tau}\right]\binom{j}{1} + 1^{j-2}\left[-\frac{t}{j\tau}\right]^2\binom{j}{2} + \ldots\right]$$

Newtonian Cooling

... where
$$\binom{j}{r} = \frac{j!}{(j-r)!\,r!}$$

we are interested in the case that dt-> 0 which is equivalent to j->◎

$$\frac{j!}{(j-r)!} = j(j-1)(j-2)...(j-r+1) \rightarrow j^r$$

as a result

$$\binom{j}{r} \rightarrow \frac{j^r}{r!}$$

Newtonian Cooling

substituted into the series for T_j we obtain:

$$T_j \rightarrow T_0 \left[1 + \frac{j}{1!} \left[- \frac{t}{j\tau} \right] + \frac{j^2}{2!} \left[- \frac{t}{j\tau} \right]^2 + \dots \right]$$

which leads to

$$T_j \rightarrow T_0 \left[1 + \left[- \frac{t}{\tau} \right] + \frac{1}{2!} \left[- \frac{t}{\tau} \right]^2 + \dots \right]$$

... which is the Taylor expansion for

$$T_j = T_0 \exp(-t / \tau)$$

Newtonian Cooling - *Convergence*

So we conclude:

For the Newtonian Cooling problem, the numerical solution converges to the exact solution when the time step dt gets smaller.

How does the numerical solution behave?

$$T_j = T_0 \exp(-t/\tau)$$

The analytical solution decays monotonically!

$$T_{j+1} = T_j (1 - \frac{dt}{\tau})$$

What are the conditions so that $T_{j+1} < T_j$?

Newtonian Cooling - *Convergence*

$$T_{j+1} = T_j \left(1 - \frac{dt}{\tau}\right)$$

$T_{j+1} < T_j$ requires

$$0 \le 1 - \frac{dt}{\tau} < 1$$

or

$$0 \le dt < \tau$$

The numerical solution decays only montonically for a limited range of values for dt! Again we seem to have a *conditional stability*.

Newtonian Cooling - *Convergence*

$$T_{j+1} = T_j(1 - \frac{dt}{\tau})$$

if $\tau < dt < 2\tau$ then $(1 - \frac{dt}{\tau}) < 0$

→ the solution oscillates but converges as $|1-dt/\tau|<1$

if $dt > 2\tau$ then $dt/\tau > 2$

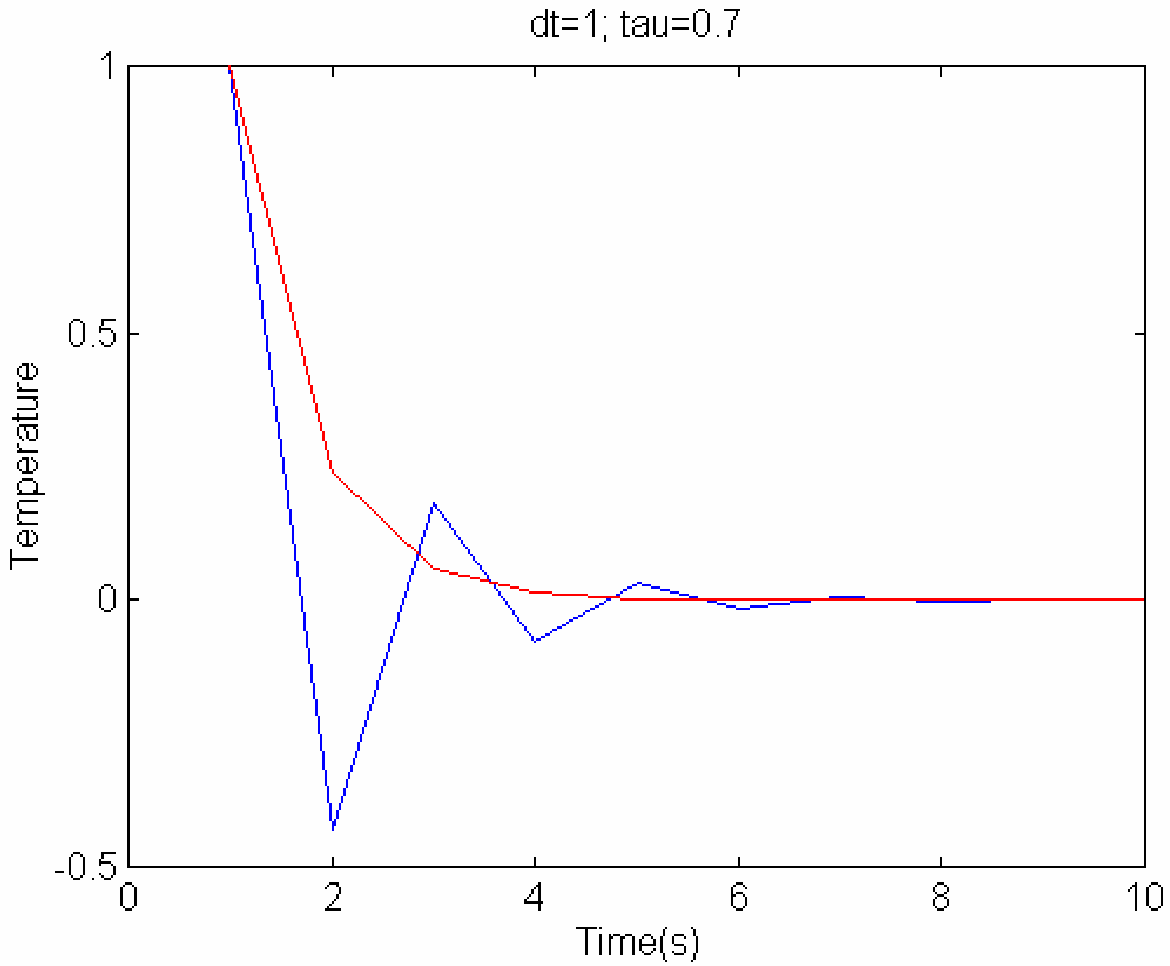
→ $1-dt/\tau<-1$ and the solution oscillates and diverges

... now let us see how the solution looks like ....

Newtonian Cooling - *Convergence*

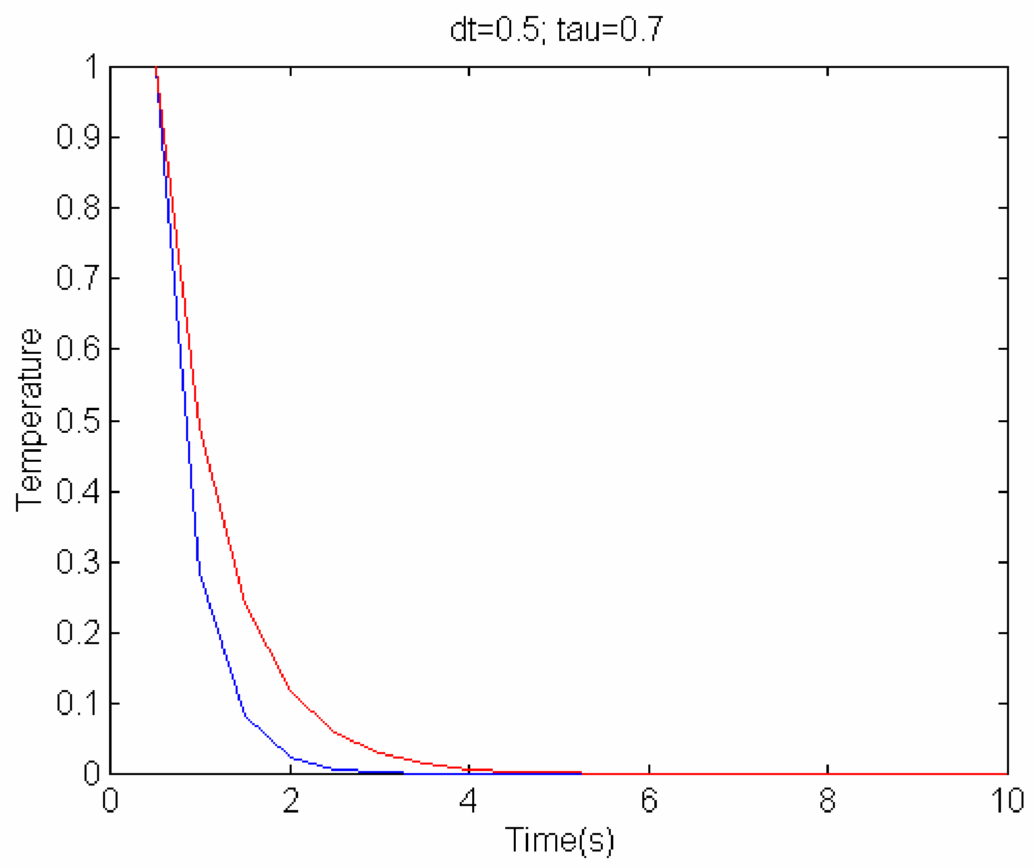
```
% Matlab Program - Newtonian Cooling

% initialise values
nt=10;
t0=1.
tau=.7;
dt=1.

% initial condition
T=t0;

% time extrapolation
for i=1:nt,
T(i+1)=T(i)-dt/tau*T(i);
end

% plotting
plot(T)
```

Newtonian Cooling - *Convergence*

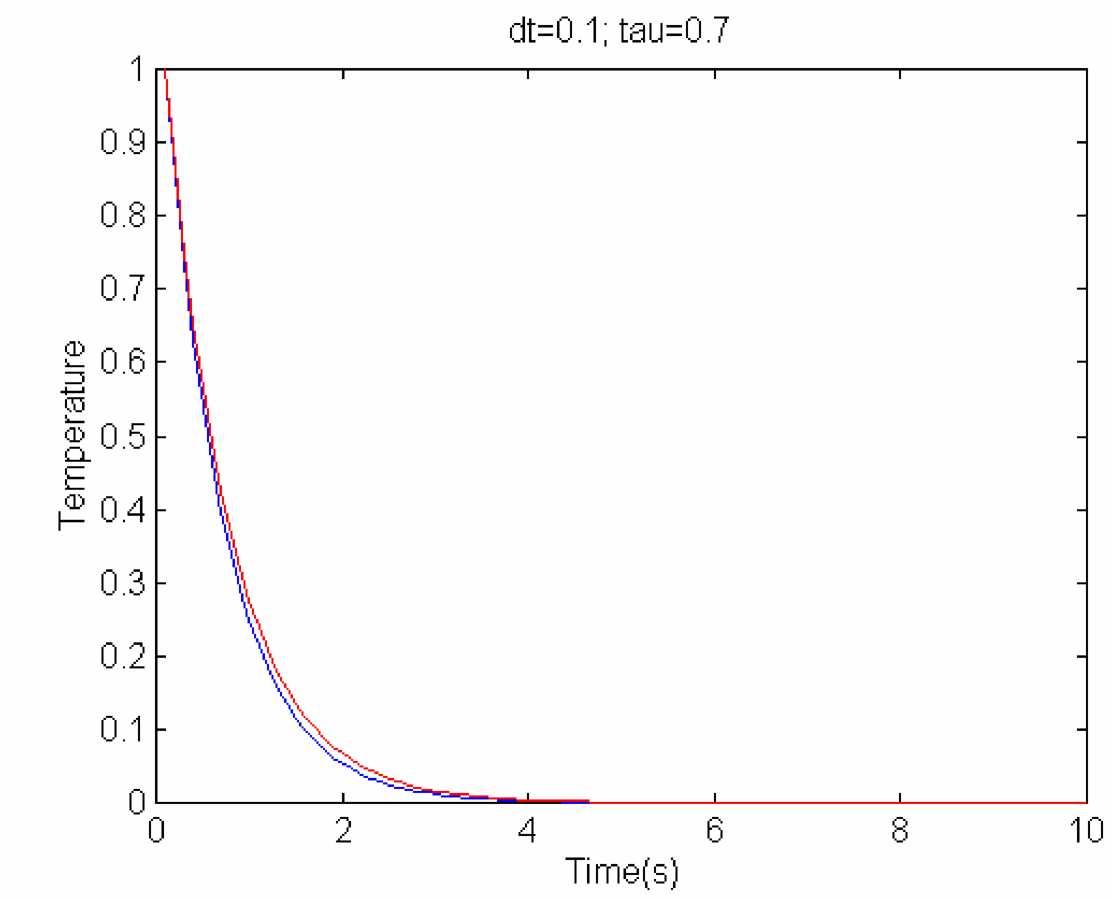


Newtonian Cooling - *Convergence*

Solution converges but does not have the right time-dependence

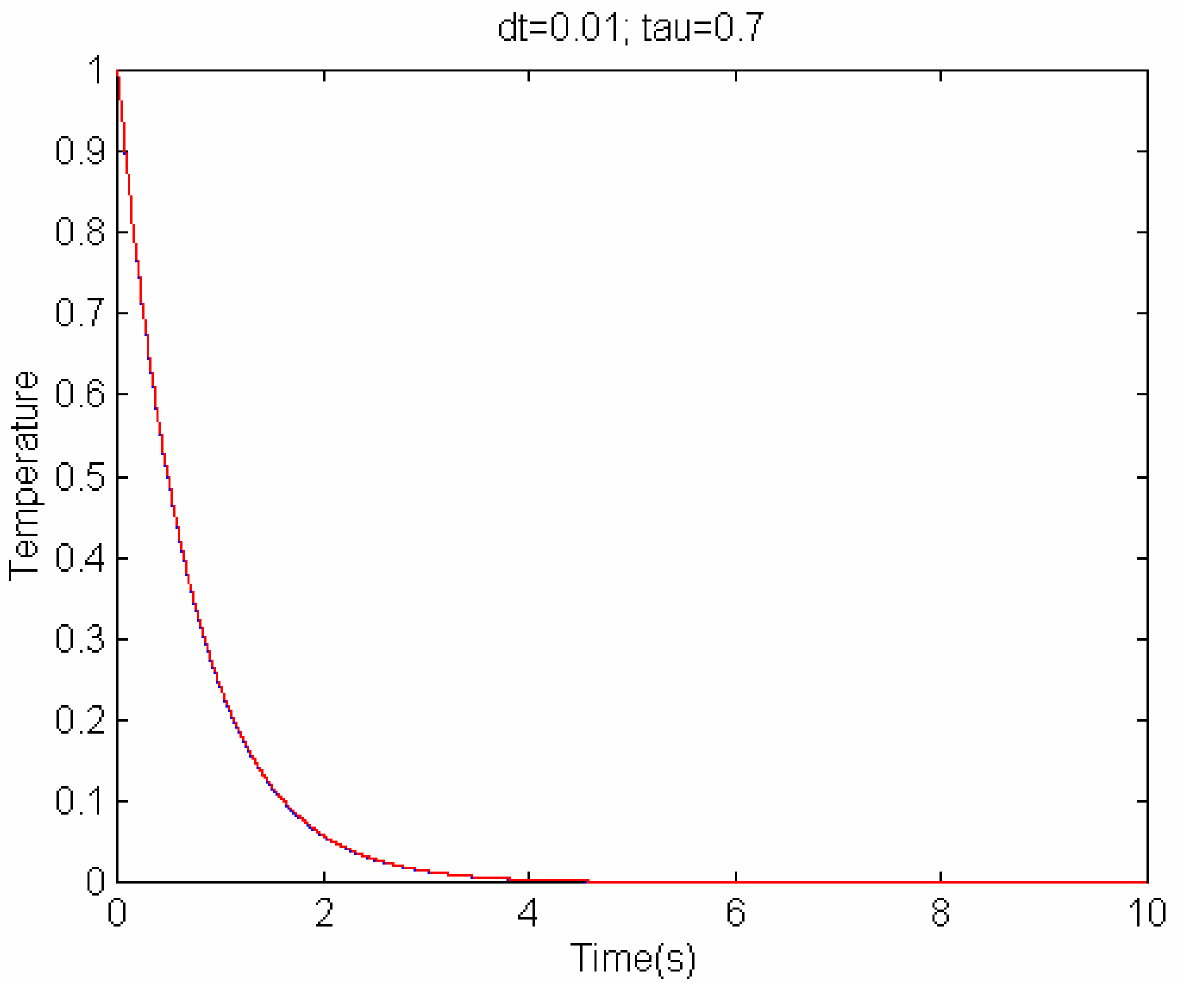


... only slight error of the time-dependence - acceptable solution ...



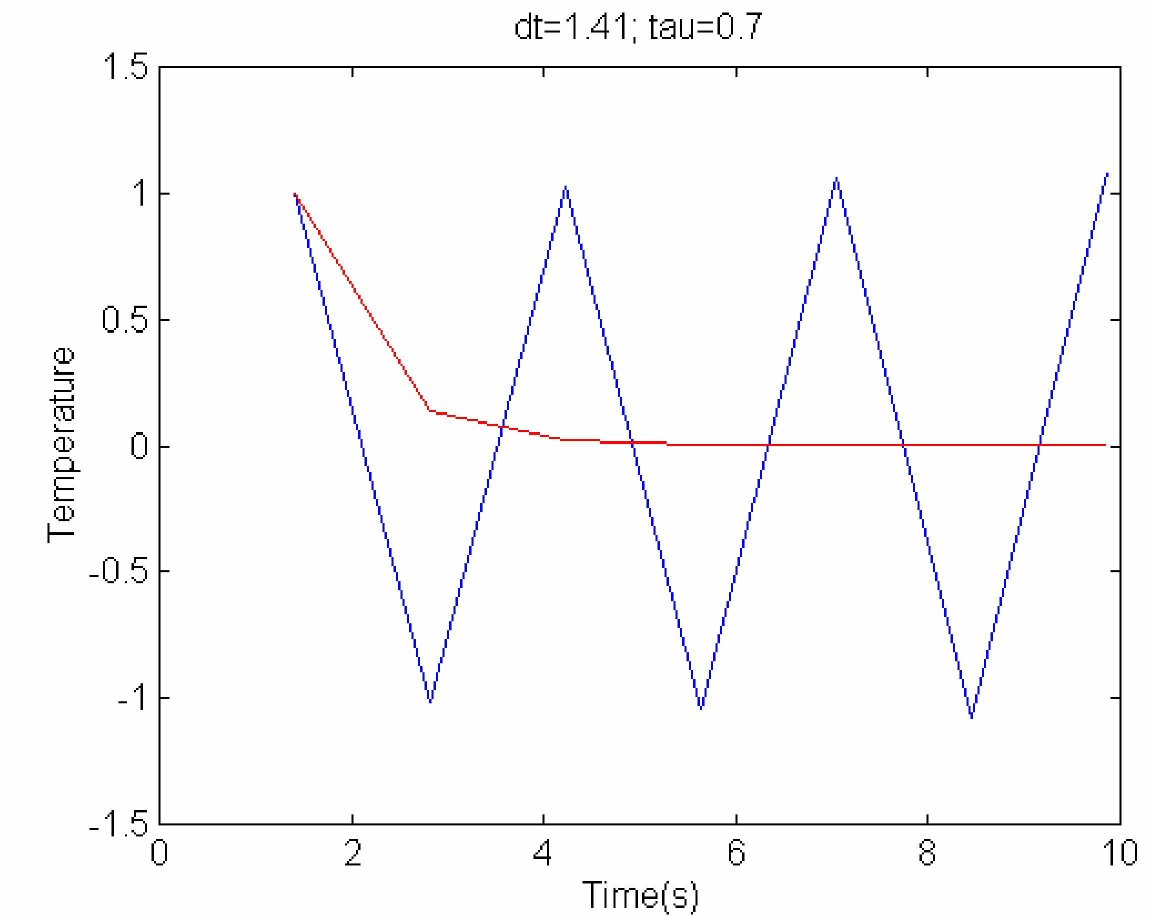
Newtonian Cooling - *Convergence*

.. very accurate solution which we pay by a fine sampling in time ...



Newtonian Cooling - *Convergence*

... this solution is wrong and unstable !



Numerical Methods in Geophysics:
Implicit Methods

What is an implicit scheme?

Explicit vs. implicit scheme for Newtonian Cooling

Crank-Nicholson Scheme (mixed explicit-implicit)

Explicit vs. implicit for the diffusion equation

Relaxation Methods

What is an implicit method?

Let us recall the *ODE*:

$$\frac{dT}{dt} = f(T,t)$$

Before we used a forward difference scheme, what happens if we use a backward difference scheme?

$$\frac{T_j - T_{j-1}}{dt} + O(dt) = f(T_j, t_j)$$

$$\Rightarrow \boxed{T_j \approx T_{j\text{-}1} + \mathrm{dt} f(T_j, t_j)}$$

What is an implicit method?

or

$$T_j \approx T_{j-1}(1+\frac{dt}{\tau})^{-1}$$

$$T_j \approx T_0(1+\frac{dt}{\tau})^{-j}$$

Is this scheme *convergent*?

Does it tend to the exact solution as dt->0? YES, it does (exercise)

Is this scheme *stable,* i.e. does T decay monotonically? This requires

$$0 < \frac{1}{1+\frac{dt}{\tau}} < 1$$

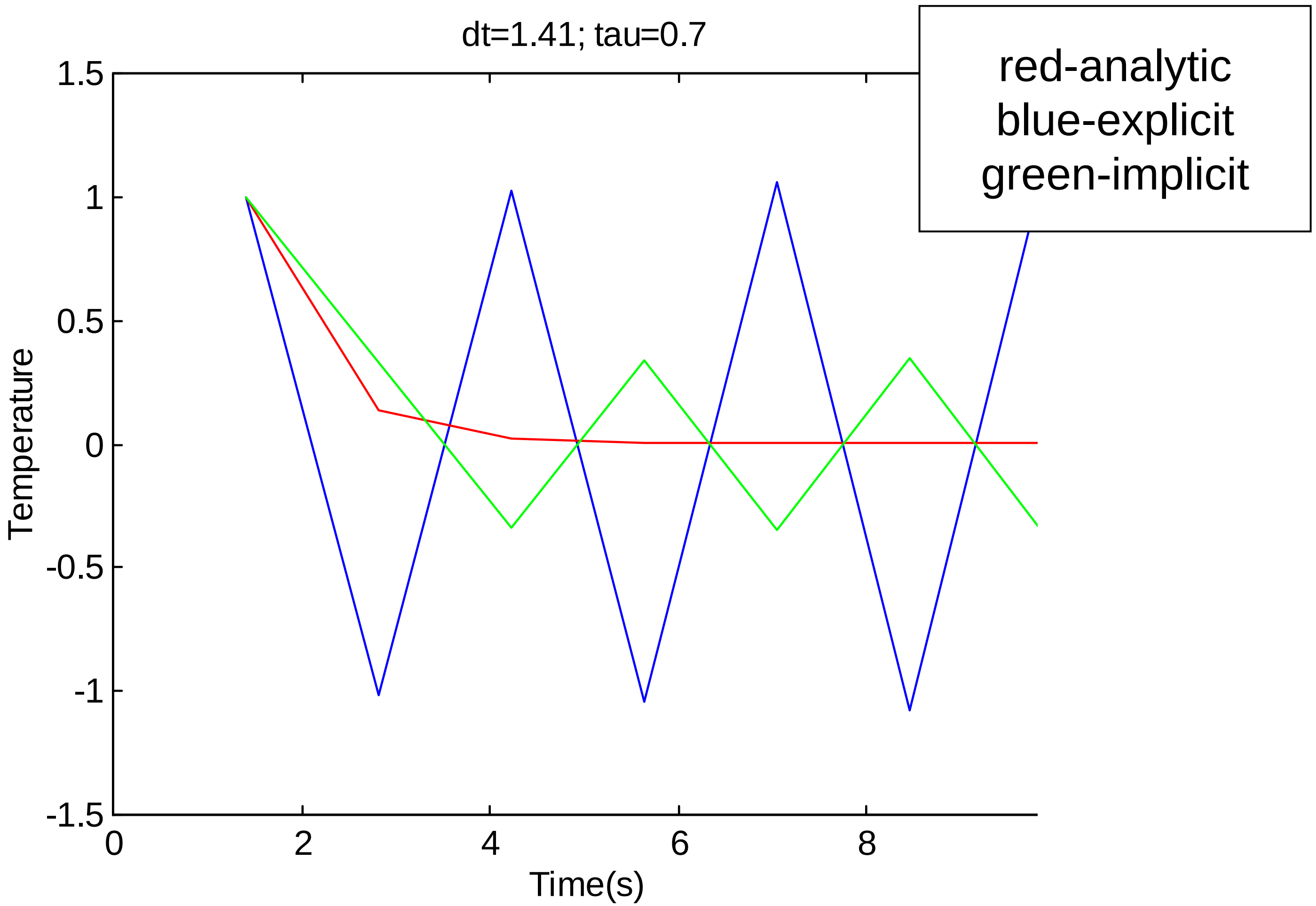
What is an implicit method?

$$0 < \frac{1}{1 + \frac{dt}{\tau}} < 1$$

This scheme is always stable! This is called
unconditional stability
... which doesn't mean it's accurate!
Let's see how it compares to the explicit method...

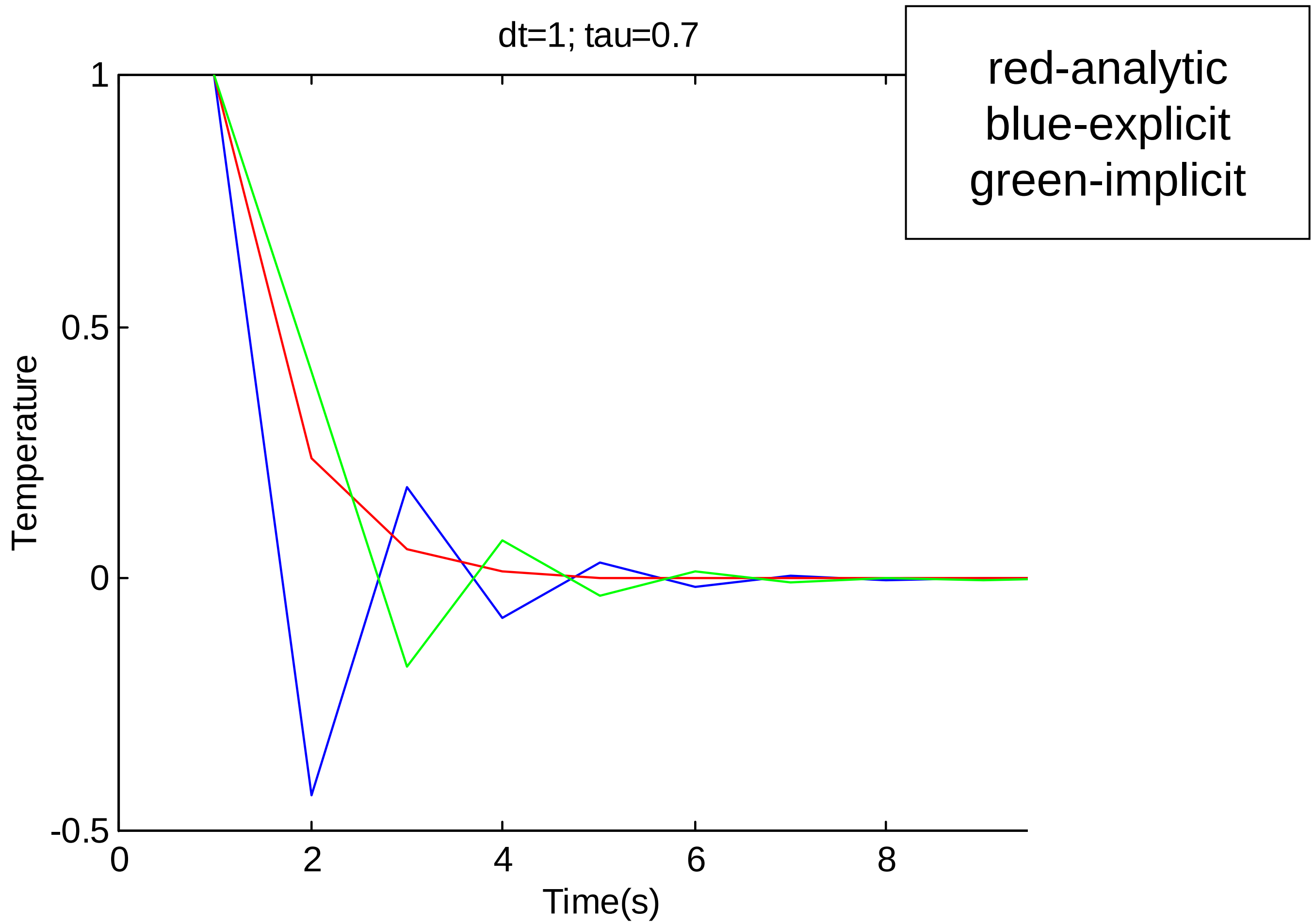
What is an implicit method?

Explicit unstable - implicit stable - both inaccurate



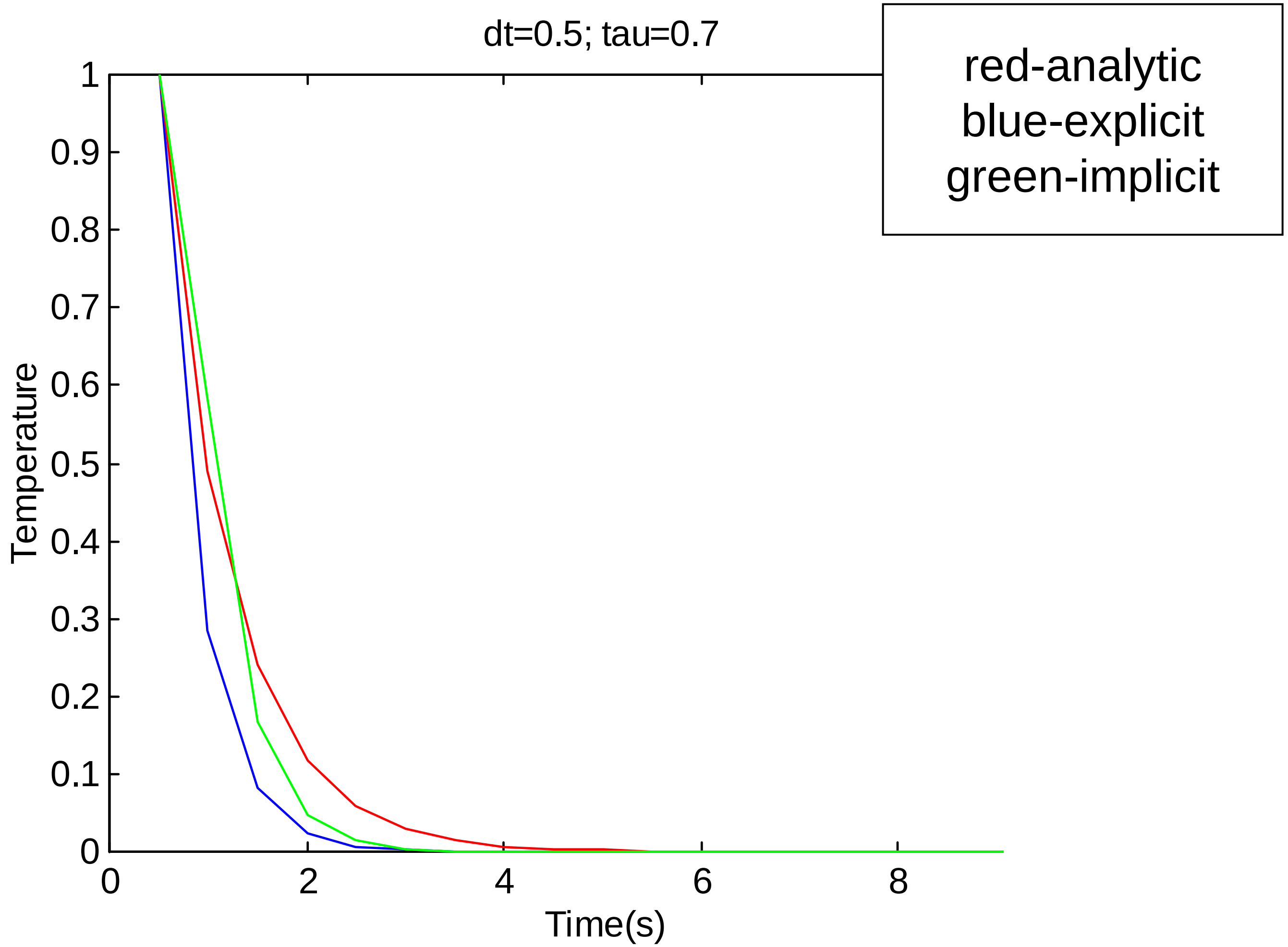
What is an implicit method?

Explicit stable - implicit stable - both inaccurate



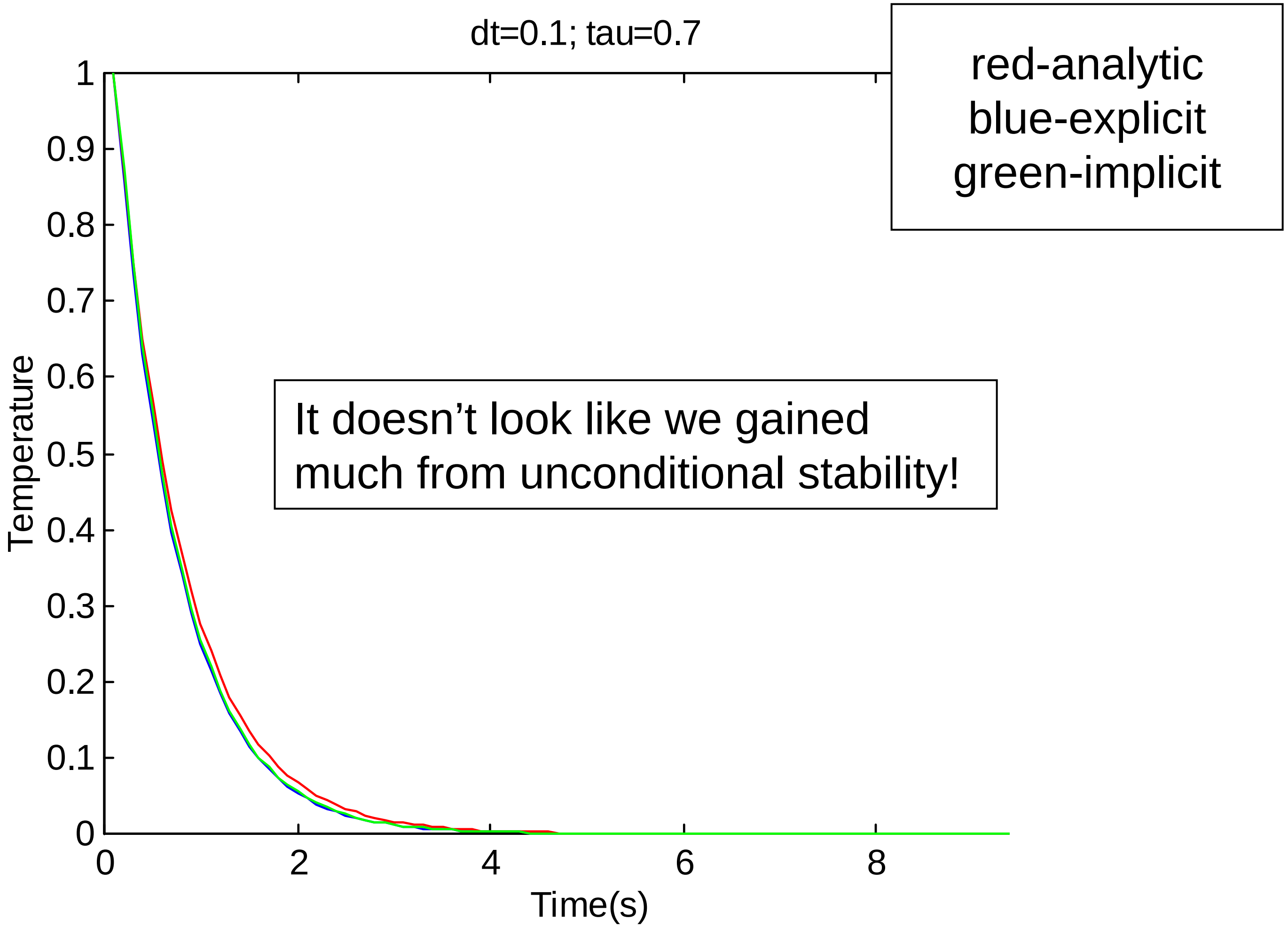
What is an implicit method?

Explicit stable - implicit stable - both inaccurate



What is an implicit method?

Explicit stable - implicit stable - both *accurate*



Mixed implicit-explicit schemes

We start again with ...

$$\frac{dT}{dt} = f(T,t)$$

Let us interpolate the right-hand side to j+1/2 so that both sides are defined at the same location in time ...

$$\frac{T_{j+1} - T_j}{dt} \approx \frac{f(T_{j+1}, t_{j+1}) + f(T_j, t_j)}{2}$$

Let us examine the accuracy of such a scheme using our usual tool, the *Taylor series*.

Mixed implicit-explicit schemes

... we learned that ...

$$\frac{T_{j+1}-T_j}{\Delta t}=\left(\frac{dT}{dt}\right)_j+\frac{\Delta t}{2}\left(\frac{d^2T}{dt^2}\right)_j+\frac{\Delta t^2}{6}\left(\frac{d^3T}{dt^3}\right)_j+O(\Delta t^3)$$

... also the interpolation can be written as ...

$$\frac{1}{2}\left(f_j+f_{j+1}\right)=\frac{1}{2}\left[2f_j+\Delta t\left(\frac{df}{dt}\right)_j+\frac{\Delta t^2}{2}\left(\frac{d^2f}{dt^2}\right)_j+O(\Delta t^3)\right]$$

since $\frac{dT}{dt}=f(T,t)$ => $\frac{d^2T}{dt^2}=\frac{df(T,t)}{dt}$

Mixed implicit-explicit schemes

... it turns out that ...
this mixed scheme is accurate to **second** order!
The previous schemes (explicit and implicit) were
all first order schemes.

Now our cooling experiment becomes:

$$\frac{T_{j+1} - T_j}{dt} \approx -\frac{1}{2\tau}(T_{j+1} + T_j)$$

$$\Rightarrow T_{j+1}(1 + \frac{dt}{2\tau}) \approx T_j(1 - \frac{dt}{2\tau})$$

leading to the extrapolation scheme

$$\Rightarrow T_{j+1} \approx T_j \left[\frac{1 - \frac{dt}{2\tau}}{1 + \frac{dt}{2\tau}} \right]$$

How stable is this scheme?
The solution decays if ...

$$-1 < \left[\frac{1 - \frac{dt}{2\tau}}{1 + \frac{dt}{2\tau}} \right] < 1$$

Mixed implicit-explicit schemes

$$-1 < \left[\frac{1 - \dfrac{dt}{2\tau}}{1 + \dfrac{dt}{2\tau}} \right] < 1$$

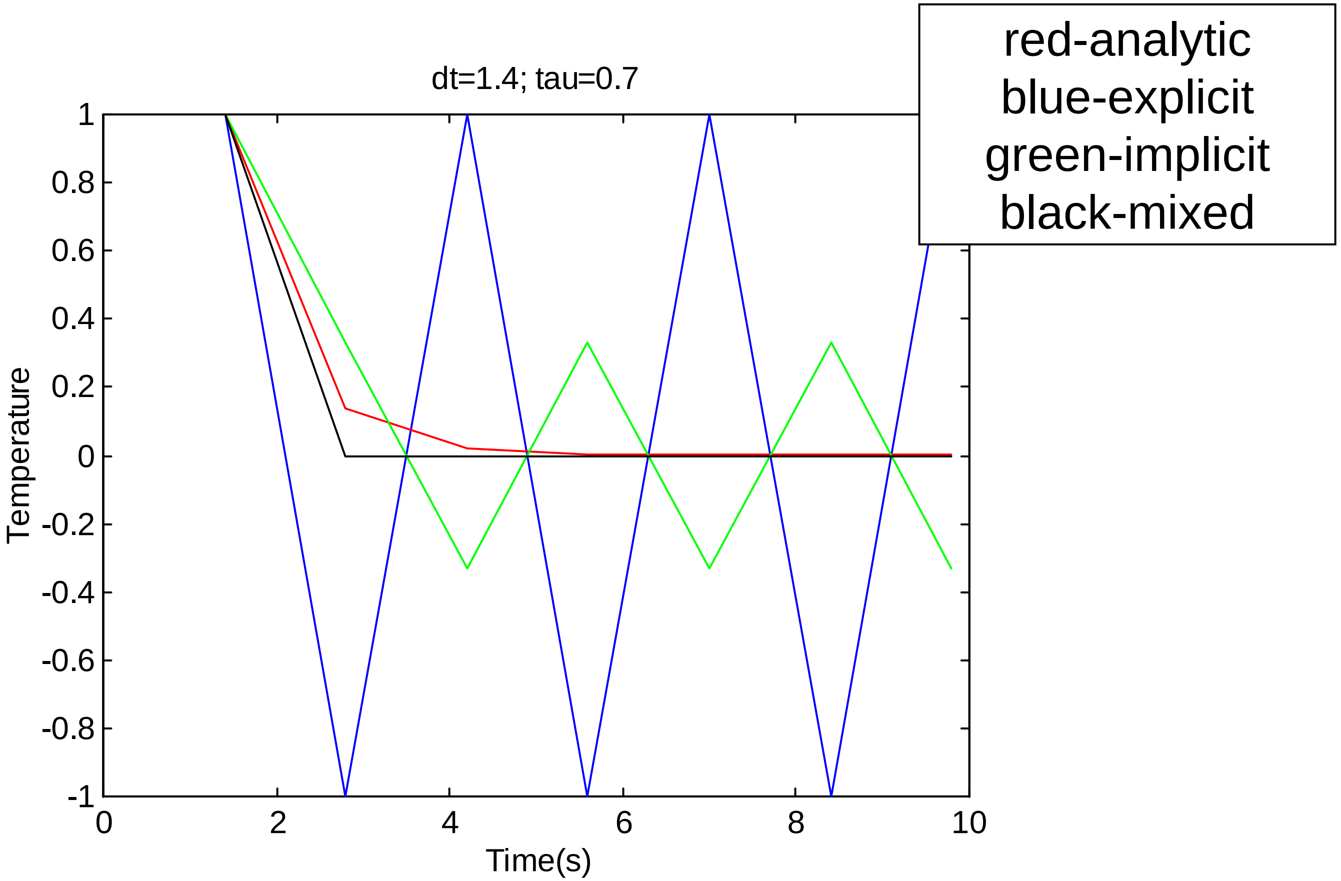
This scheme is always stable for positive dt and τ!
If dt>2 τ, the solution decreases monotonically!

Let us now look at the Matlab code and then compare it to the other approaches.

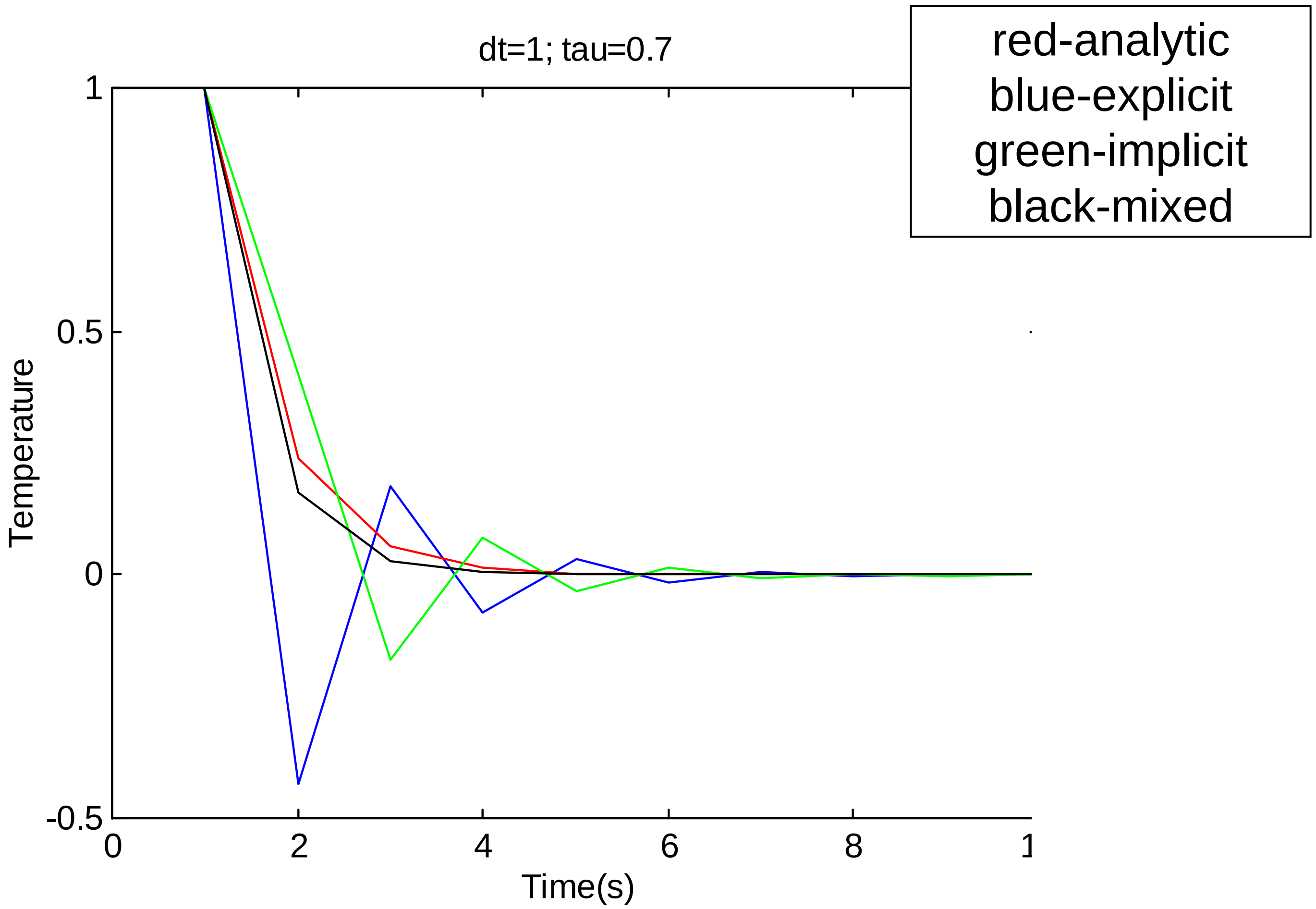
Mixed implicit-explicit schemes

```
t0=1.
tau=.7;
dt=.1;
dt=input(' Give dt : ');

nt=round(10/dt);

T=t0;
Ta(1)=1;
Ti(1)=1;
Tm(1)=1;

for i=1:nt,
t(i)=i*dt;
T(i+1)=T(i)-dt/tau*T(i);                      % explicit forward
Ta(i+1)=exp(-dt*i/tau);                       % analytic solution
Ti(i+1)=T(i)*(1+dt/tau)^(-1);                 % implicit
Tm(i+1)=(1-dt/(2*tau))/(1+dt/(2*tau))*Tm(i); % mixed
end

plot(t,T(1:nt),'b-',t,Ta(1:nt),'r-',t,Ti(1:nt),'g-',t,Tm(1:nt),'k-')
xlabel('Time(s)')
ylabel('Temperature')
```

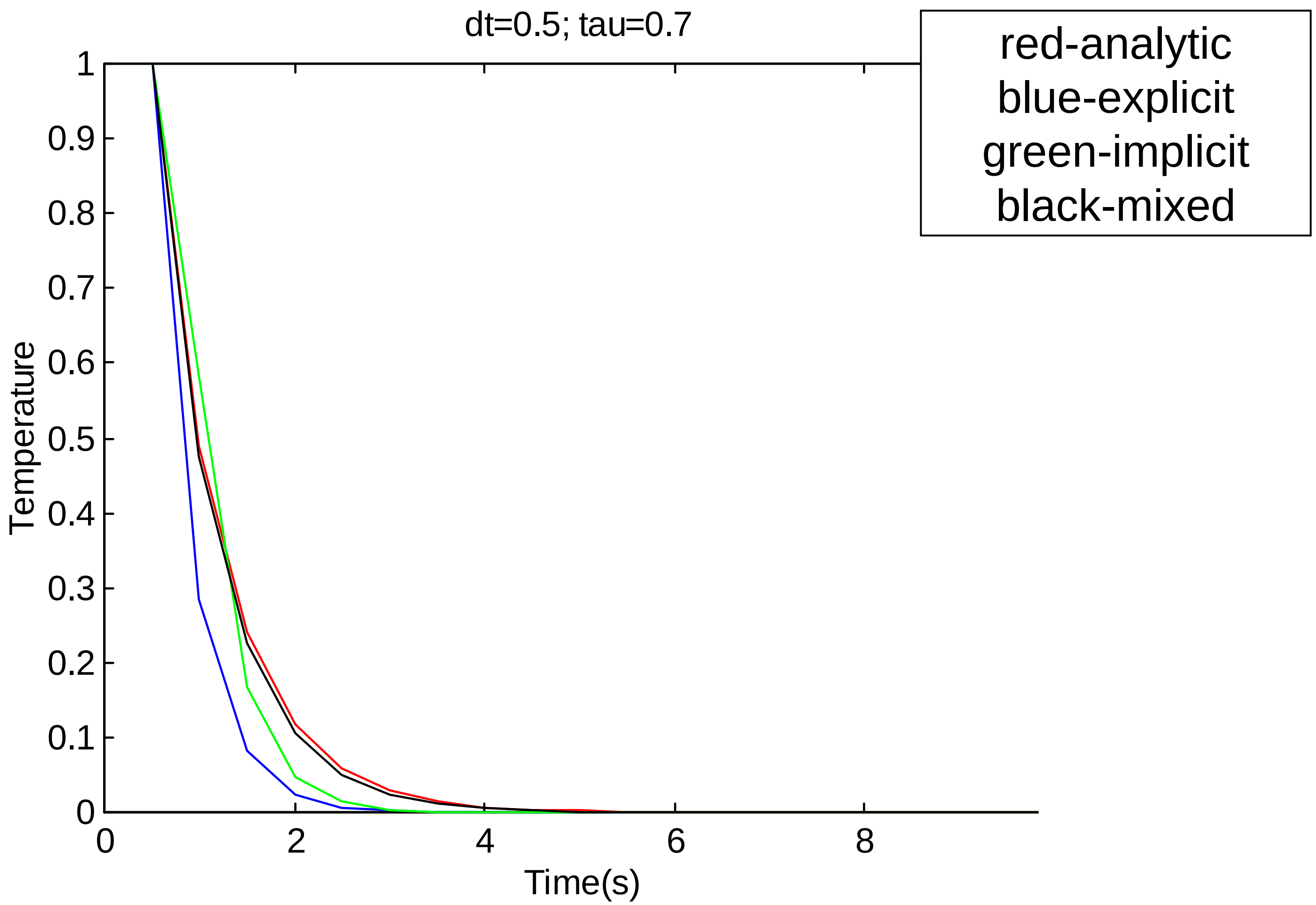
Mixed implicit-explicit schemes



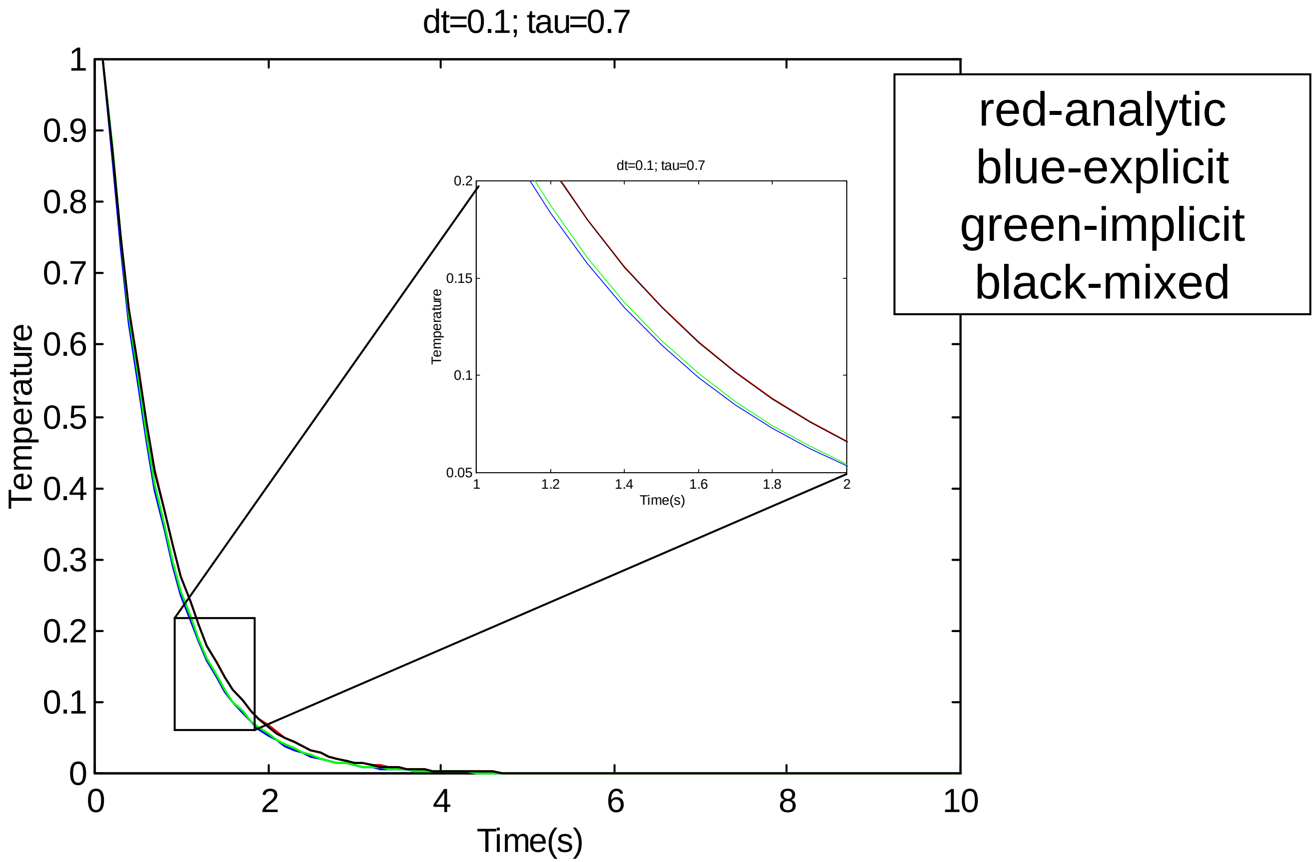
Mixed implicit-explicit schemes



Mixed implicit-explicit schemes



Mixed implicit-explicit schemes



The mixed scheme is a clear ***winner***!

Explicit - Implicit Methods - Summary

Certain FD approximations to time-dependent partial differential equations lead to implicit solutions. That means to propagate (extrapolate) the numerical solution in time, a linear system of equations has to be solved.

The solution to this system usually requires the use of matrix inversion techniques.

The advantage of some implicit schemes is that they are unconditionally stable, which however does not mean they are very accurate.

It is possible to formulate mixed explicit-implicit schemes (e.g. Crank-Nickolson or trapezoidal schemes) , which are more accurate than the equivalent explicit or implicit schemes.