Finite Difference (Forward Problem)

In this toturial we will consider the IVP:

$$\dot{x} = \sin(t) - 2x, x(0) = 0.$$

which has the analytical solution:

$$x(t) = -\frac{1}{5}\cos(t) + \frac{2}{5}\sin(t) + \frac{1}{5}e^{-2t}$$

We will provide a brief illustration of the accuracy of some numerical methods in solving the IVP. Then we will discuss the derivative estimation from time series.

Initial Value Problem

```
clearvars; close all; clc;
```

First, we write the ODE function:

```
xdot = @(t,x) sin(t)-2*x;
```

and the true solution $x(t)$

```
xt   = @(t)   -0.2*cos(t) +0.4*sin(t) +0.2*exp(-2*t);
```

We choose different time steps to explore the accuracy of different methods for different time step.

```
tau = logspace(-4,0,20)';
```

Initialize your variables:

```
E   = zeros(length(tau),4); %To store the error in the estimation
                            %since we will estimate the error for each tau
                            %then they have the same number of rows
                            % E has 4 columns to store the error for 4
                            % different methods, so we will store the error
                            % in the following order:
                            % E [Forward_Diff,  Backward_Diff,  Central_Diff,
                            %    Runge-Kutta]


x0 = 0;         %initial condition, given from the IVP
Tfinal = 10;    %Final time of the time series
```

Now, we start our main loop which will consider different value for the time step each time

```
for i=1:length(tau)
    t = (0:tau(i):Tfinal)';     %The time span, meaning that it is the
                                % time steps or the time values that we will
                                % obtain the system state at.
```

Forward Difference Method

For each time step value, we use the forward difference method given by:

$$x(t+\tau) = x(t) + \tau\dot{x}(t)$$

```
    FD = x0; %initialize the estimated solution FD with the initial condition
    for j=1:length(t)-1
        x = FD(end) + tau(i)*xdot(t(j),FD(end));
        FD = [FD; x]; %#ok
    end
```

Backward Difference

In the basic form, we can write the backward Euler method as:

$$\dot{x}(t, x(t)) = \frac{x(t) - x(t-\tau)}{\tau}$$

which gives:

$$x(t) = x(t-\tau) + \tau\dot{x}(t, x(t))$$

Method 1

You see that $x(t)$ is in both sides of the equation. Then, we try to find $w = x(t)$ that satisfy the algebraic equation above. This can be done by finding the root of the function:

$$f(w) = w - x(t-\tau) - \tau\dot{x}(t, w)$$

It is one dimentional optimization problem, and we already have a very close (Good) intital guess which is $x(t-\tau)$. We can implement this method as the following:

```
    % Backward difference
    BD1 = x0; %BD1 store the backward difference solution using the first method.
    for j=1:length(t)-1
        f = @(w) w - BD1(end) - tau(i)*xdot(t(j+1),w);
        x = fzero(f,BD1(end)); %Find a root with initial guess.
        BD1 = [BD1; x]; %#ok
    end
```

Method 2

However, since our ode function is linear (in x), we can re-write the equation:

$$x(t) = x(t-\tau) + \tau\dot{x}(t, x(t))$$

by substituting the expression of $\dot{x}$, then we have

$$x(t) = x(t-\tau) + \tau(\sin(t) - 2x(t))$$

solve for $x(t)$ we will have:

$$x(t) = \frac{x(t-\tau) + \tau\sin(t)}{1+2\tau}$$

Then, we can implement this solution as the following:

```
BD2 = x0; %BD2 store the backward difference solution using the second method.
for j=1:length(t)-1
    x = (1/(1+2*tau(i)))*(BD2(end) + tau(i)*sin(t(j+1)));
    BD2 = [BD2; x]; %#ok
end
```

Keep in mined that this works just because the ode function is linear. Method 1 above is the more common and general method to solve the IVP with the backward difference.

Central Difference

The central difference is given by:

$$\dot{x}(t, x(t)) = \frac{x(t+\tau) - x(t-\tau)}{2\tau}$$

However, this form of the central difference is unstable and can result with high error. Instead, the midpoint method or the second order Runge-Kutta (RK2) can be equivelent and more stable. Keeping in mind that by considering a midpoint, we can write the above equation as:

$$\dot{x}(t+\frac{\tau}{2}, x(t+\frac{\tau}{2})) = \frac{x(t+\tau) - x(t)}{\tau}$$

and to find $\dot{x}$ above, we need to obtain $x(t+\frac{\tau}{2})$ by:

$$x(t+\frac{\tau}{2}) = x(t) + \frac{\tau}{2}\dot{x}(t, x(t))$$

This combination to solve the central difference is practically the second order Runge-Kutta method:

```
RK2 = x0;
for j=1:length(t)-1
    k1 = RK2(end) + 0.5*tau(i)*xdot(t(j),RK2(end));%last equation above
    k2 = tau(i)*xdot(t(j)+0.5*tau(i),k1);
    x  = RK2(end) + k2;
    RK2 = [RK2; x]; %#ok
end
```

Runge-Kutta (RK4)

```
    % Runge-Kutta (RK4)
    RK4 = x0;
    for j=1:length(t)-1
        k1 = xdot(t(j),RK4(end));
        k2 = xdot(t(j)+0.5*tau(i),RK4(end)+0.5*tau(i)*k1);
        k3 = xdot(t(j)+0.5*tau(i),RK4(end)+0.5*tau(i)*k2);
        k4 = xdot(t(j)+tau(i),RK4(end)+tau(i)*k3);
        x = RK4(end) + (1/6)*tau(i)*(k1+2*k2+2*k3+k4);
        RK4 = [RK4; x]; %#ok
    end
```

Results

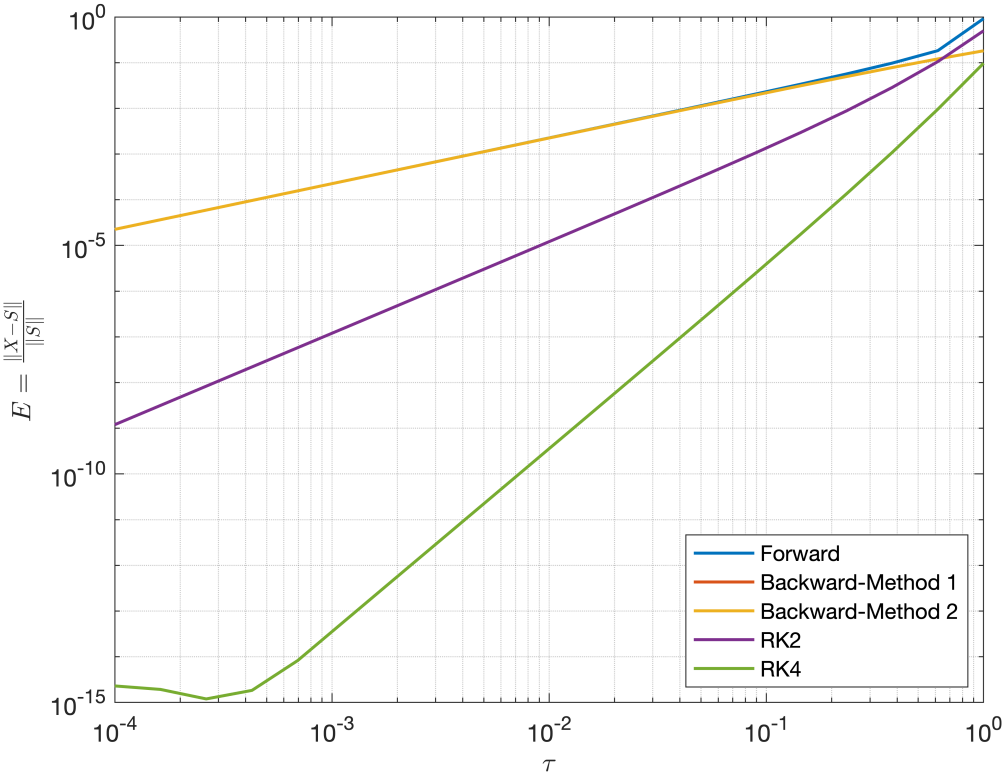
```
    S = xt(t); %The true solution to compare our estimation
```

We find the relative error in the estimated trajectories:

$$E = \frac{\|X - S\|}{\|S\|}$$

where X is the vector of trajectories found using your numerical estimation and S is the actual trajectories found using the analytical solution.

```
    E(i,1) = norm(FD-S)/norm(S);
    E(i,2) = norm(BD1-S)/norm(S);
    E(i,3) = norm(BD2-S)/norm(S);
    E(i,4) = norm(RK2-S)/norm(S);
    E(i,5) = norm(RK4-S)/norm(S);

end %now the loop will go back and compute the error for the next value of tau

loglog(tau,E,'-','LineWidth',1.5)
legend("Forward","Backward-Method 1","Backward-Method 2","RK2","RK4","Location","south
grid minor
xlabel("$\tau$","Interpreter","latex")
ylabel("$E=\frac{\|X-S\|}{\|S\|}$","Interpreter","latex")
set(gca,'FontSize',12)
```



The two methods used for solving the backward difference are identical (overlay each other).

You see that:

- Forward and Backward difference hav the same accuracy order for small step sizes.
- However, the backward difference is more stable. (See at large step size).
- The accuracy of Runge-Kutta increase with increase the order.
- Accuracy of RK4 can be within the machine precision just by using a step size of order -4.

Finite Difference (Inverse Problem)

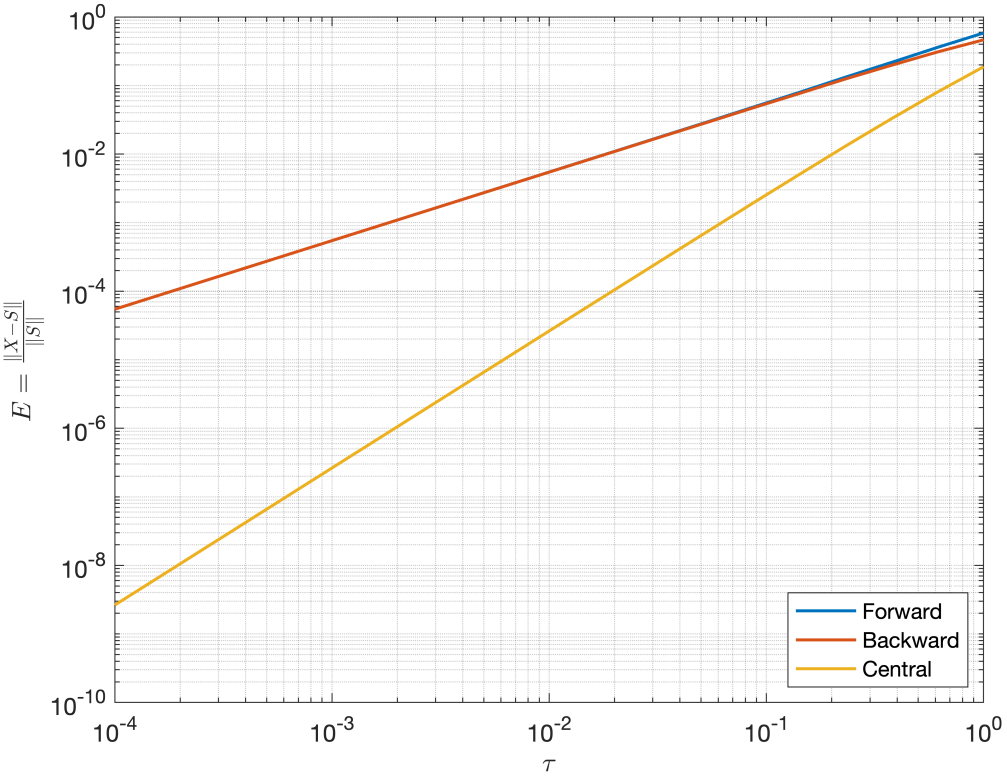
In your HW, the above discussion, we investigated the accuracy of different finite difference methods in solving the forward problem.

Now we will see how some of these methods will perform in the inverse problem. Meaning that:

If you are given $x(t)$ as a data points, how accurate the derivative estimation ($\dot{x}(t, x(t))$) can be?

```
Err = zeros(length(tau),3); %To store the error
```

```
for i=1:length(tau)
    t = (0:tau(i):Tfinal)';     %The time span, meaning that it is the
                                % time steps or the time values that we will
                                % obtain the system state at.


    x = xt(t); %the true trajectories


    xdotTrue = xdot(t,x); %the true derivative.


    %Now we find the derivative using different methods

    %Forward difference
    xdotFD = (x(2:end)-x(1:end-1))./tau(i);

    Err(i,1) = norm(xdotTrue(1:end-1)-xdotFD)/norm(xdotTrue(1:end-1));
    %remember, the forward difference does not have the last point
    %estimation

    %Backward difference
    xdotBD = (x(2:end)-x(1:end-1))./tau(i); %The same as above

    Err(i,2) = norm(xdotTrue(2:end)-xdotBD)/norm(xdotTrue(2:end));
    %remember, the backward difference does not have the first point
    %estimation

    %Central Difference
    xdotCD = (x(3:end)-x(1:end-2))./(2*tau(i));

    Err(i,3) = norm(xdotTrue(2:end-1)-xdotCD)/norm(xdotTrue(2:end-1));
    %remember, the backward difference does not have the first and the last points
    %estimation

end

loglog(tau,Err,'-','LineWidth',1.5)
legend("Forward","Backward","Central","Location","southeast")
grid minor
xlabel("$\tau$","Interpreter","latex")
ylabel("$E=\frac{\|X-S\|}{\|S\|}$","Interpreter","latex")
set(gca,'FontSize',12)
```



You can see that, for the inverse problem (Estimating the derivative from data), the forward and backward difference are very similar, and the central difference is more accurate than both of them.

Now, in the inverse problem, it is VERY important that you think of the following:

1. How the methods above will perform with the presence of noise?.
2. If we consider the noise level as a constant multiple of the time step, $..., 0.01\tau, 0.1\tau, \tau, 10\tau, 100\tau, ..., \rho\tau$ then try to plot the curve above with this constant multiple ρ in the x-axis and the error on the y-axis. (log scale plot), you will have several curves for each method, each curve of them associated with specific time step. How you can arrange such comparison? Is there better way to perform such analysis?
3. What other methods can perform better in the presence of noise?