Numerical Solution of Ordinary Differential Equations (ODEs)

- It is often the case when modeling some phenomena that we know something about the rate of change of the quantity of interest, that is, its derivative.

- For example, in Calculus I you probably looked at exponential growth and decay laws. One application of this is to model the decay of a sample of a radioactive isotope by saying that the rate of decay is proportional to the amount present at any time. For example, if $Q(t)$ denotes the amount present at time t then

$$\frac{dQ}{dt} = kQ$$

- Recall that the general solution of this problem is $Q(t) = Ce^{kt}$ for some constant C since $Q'(t) = Cke^{kt} = CQ(t)$

- In order to uniquely determine the solution then we must be given an initial condition such as $Q(0) = Q_0$ which gives us the unique solution $Q(t) = Q_0e^{kt}$.

- We call the following ODE a first order initial value problem.

Initial Value Problem (IVP)

$$\frac{dy}{dt} = f(t, y) \quad t_0 < t \le T$$

$$y(t_0) = y_0$$

Here $f(t, y)$ is a given function of t, y and y_0 is the given initial data.

- We will also use the shorthand notation $y'(t) = f(t, y)$.
- We call this IVP a time dependent problem and our goal will be to
  - determine an accurate solution at some time T
  - and/or determine the time evolution of the solution.
- The information we have to determine the solution is the

  - the initial value of y

  - the slope of y given by $f(t, y)$, i.e., how y changes with time.

- For example, if $\frac{dy}{dt} = 1$ and $y(0) = 2$ then this says that the slope is a constant value one and y is initially two so we know that $y(t) = t + 2$.

- Our model IVP is a first order ODE, that is, the highest derivative in the equation is first order.

Uniqueness of the Solution of an Initial Value Problem

- Before we approximate the solution of a differential equation in general, we should ask ourselves if it has a unique solution.
- For example, if you were asked to write a code to approximate the solution of $y'(t) = \sin t$ you would not be able to. The reason is that this problem does not have a unique solution but rather its solution is given by $y(t) = -\cos t + C$ for some arbitrary constant C. The problem is, of course, that we have not provided an initial condition.
- We might ask ourselves, however, if every IVP has a solution; that is for any $f(t, y)$.
- The answer to this is no. We have to require a certain amount of smoothness of the function $f(t, y)$.
- Standard texts on ODEs discuss conditions which guarantee existence and uniqueness of a solution to our IVP.

- As an example, consider the IVP

$$y'(t) = \sqrt{y}, \qquad y(0) = 0$$

This problem does not have a unique solution. In fact both

$$y = 0 \quad \text{and} \quad y = \frac{1}{4}t^2$$

are solutions.

- Recall that to verify a given function is a solution to the IVP we simply show that it satisfies the DE and the initial condition. In our example $y = \frac{1}{4}t^2$ is a solution to our IVP since

$$y' = \frac{1}{2}t = \sqrt{\frac{1}{4}t^2} = \sqrt{y}$$

Clearly it satisfies the initial condition $y(0) = 0$

- In the sequel, we will assume that our IVP has a solution which is unique.

Approximating Derivatives by Difference Quotients

- One standard approach is approximating the solution of a DE is to replace the derivatives with difference quotients.
- You have already used difference quotients in calculus. For example, you have approximated the derivative of y with respect to t by the change in y over the change in t. This is a difference quotient because you have the difference in y divided by a difference in t; i.e.,

$$\frac{y(t+\Delta t)-y(t)}{\Delta t}$$

- An easy way to derive difference quotients is through the use of Taylor's series.
- Recall that a Taylor's series for $y(t)$ in the neighborhood of t is given by

$$y(t+\Delta t)=y(t)+\frac{\Delta t}{1!}\frac{dy}{dt}+\frac{\Delta t^2}{2!}\frac{d^2y}{dt^2}+\frac{\Delta t^3}{3!}\frac{d^3y}{dt^3}+\cdots+\frac{\Delta t^n}{n!}\frac{d^ny}{dt^n}+\cdots$$

- We assume this expansion is valid near t; i.e., when Δt is small. Note that we expect the terms to decrease in size as n increases because each term is a factor of Δt to a higher power.

- If we keep two terms on the right hand side of this expansion then we have

  $$y(t+\Delta t) \approx y(t) + \frac{\Delta t}{1!}\frac{dy}{dt}$$

  which implies

  $$y'(t) \approx \frac{y(t+\Delta t) - y(t)}{\Delta t}$$

- This is called a forward difference because we are sitting at the point t and differencing ahead to $t + \Delta t$.

- If we consider the Taylor series

  $$y(t-\Delta t) = y(t) - \frac{\Delta t}{1!}\frac{dy}{dt} + \frac{\Delta t^2}{2!}\frac{d^2y}{dt^2} - \frac{\Delta t^3}{3!}\frac{d^3y}{dt^3} + \cdots + (-1)^n\frac{\Delta t^n}{n!}\frac{d^ny}{dt^n} + \cdots$$

  then keeping the first two terms on the right gives

  $$y(t-\Delta t) \approx y(t) - \frac{\Delta t}{1!}\frac{dy}{dt} \Rightarrow y'(t) \approx \frac{y(t) - y(t-\Delta t)}{\Delta t}$$

This is called a backward difference because we are sitting at the point t and differencing backwards in time to $t - \Delta t$.

- We can also obtain another approximation to $y'(t)$ by keeping the first three terms on the right side of each expansion and then combining them. We have the two approximations

$$y(t + \Delta t) \approx y(t) + \frac{\Delta t}{1!}\frac{dy}{dt} + \frac{\Delta t^2}{2!}\frac{d^2y}{dt^2}$$

$$y(t - \Delta t) \approx y(t) - \frac{\Delta t}{1!}\frac{dy}{dt} + \frac{\Delta t^2}{2!}\frac{d^2y}{dt^2}$$

and subtracting gives

$$y(t + \Delta t) - y(t - \Delta t) \approx 2\frac{\Delta t}{1!}\frac{dy}{dt} \Rightarrow \frac{dy}{dt} \approx \frac{y(t + \Delta t) - y(t - \Delta t)}{2\Delta t}$$

This is called a centered difference approximation to $y'(t)$.

- We can also obtain approximations to higher order derivatives. For example, to approximate $y''(t)$ we add the expansions (so that the terms for $y'(t)$ disappear) for $y(t + \Delta t)$ and $y(t - \Delta t)$ to get

$$y(t+\Delta t) + y(t-\Delta t) = 2y(t) + \Delta t^2 y''(t) + \mathcal{O}(\Delta t^4)$$

so that

$$y''(t) \approx \frac{y(t+\Delta t) - 2y(t) + y(t-\Delta t)}{\Delta t^2}$$

This is called a second centered difference. We will return to this difference quotient when we look at a second order equation.

- How do we know which approximation to $y'(t)$ to use?
- All are useful in particular problems so the type of the problem is important. For example, we will see there is a difference in choice of differences for an IVP and a BVP.
- Another way to choose between two difference quotients which work for your particular problem is the accuracy of the approximation.

Forward Difference Approximation to y' at t

$$y'(t) \approx \frac{y(t+\Delta t) - y(t)}{\Delta t}$$

Backward Difference Approximation to y' at t

$$y'(t) \approx \frac{y(t) - y(t-\Delta t)}{\Delta t}$$

Centered Difference Approximation to y' at t

$$y'(t) \approx \frac{y(t+\Delta t) - y(t-\Delta t)}{2\Delta t}$$

- We call the forward and backward approximations first order and the centered difference a second order approximation. We will make this more precise shortly.

Errors in difference approximations

- We would expect the centered difference to be more accurate than either the forward or backward difference. Why?
- When we obtained our forward or backward difference we kept two terms on the right hand side. The next term (which dominates the others on the right) is order $(\Delta t)^2$. Let's keep this term and look again at the derivation

  $$y(t+\Delta t) \approx y(t) + \frac{\Delta t}{1!}\frac{dy}{dt} + \frac{\Delta t^2}{2!}\frac{d^2y}{dt^2}$$

  which implies

  $$y'(t) \approx \frac{y(t+\Delta t) - y(t)}{\Delta t} - \frac{\Delta t}{2}\frac{d^2y}{dt^2}$$

- So we say that the error is order Δt and denote as $\mathcal{O}(\Delta t)$ which means a constant times Δt.
- Clearly a backward difference has the same accuracy, i.e., $\mathcal{O}(\Delta t)$.
- Let's look at the centered difference. Recall that we kept three terms on the right side to derive the difference approximation. To derive the error we keep the next term.

$$y(t+\Delta t) \approx y(t) + \frac{\Delta t}{1!}\frac{dy}{dt} + \frac{\Delta t^2}{2!}\frac{d^2y}{dt^2} + \frac{\Delta t^3}{3!}\frac{d^3y}{dt^3}$$

$$y(t-\Delta t) \approx y(t) - \frac{\Delta t}{1!}\frac{dy}{dt} + \frac{\Delta t^2}{2!}\frac{d^2y}{dt^2} - \frac{\Delta t^3}{3!}\frac{d^3y}{dt^3}$$

Recall that we subtracted these two expansions to get our approximation so we have

$$y(t+\Delta t) - y(t-\Delta t) \approx 2\Delta t\frac{dy}{dt} + 2\frac{\Delta t^3}{3!}\frac{d^3y}{dt^3}$$

which implies

$$y'(t) \approx \frac{y(t+\Delta t) - y(t-\Delta t)}{2\Delta t} - \frac{\Delta t^2}{6}\frac{d^3y}{dt^3}$$

and thus

$$y'(t) = \frac{y(t+\Delta t) - y(t-\Delta t)}{2\Delta t} + \mathcal{O}(\Delta t)^2$$

- We see that this approximation to $y'(t)$ has a smaller error than either the forward or backward difference.
- These three differences are the most commonly used for approximations to the first derivative. We summarize them here.

Approximating the Solution to our IVP

- Recall that we are given the value of y at some initial t; for simplicity we take $t = 0$ and the value of y at zero to be y_0. We want to find y at later times. This was represented by our general IVP

$$\frac{dy}{dt} = f(y,t) \qquad y(0) = y_0$$

  where f is a given function of t and y.

- The strategy to approximating an IVP is to use the initial value y_0 at $t = 0$ and the slope (i.e., f) to predict the solution at time $t = \Delta t$. Then to use the solution at $t = \Delta t$ and the slope (or perhaps both the solution at $t = \Delta t$ and $t = 0$) to predict the solution at $t = 2\Delta t$, etc.

- If we do everything correctly, then we expect that as $\Delta t \to 0$ our discrete solution at $\Delta t, 2\Delta t, 3\Delta t, \cdots$ will approach the actual solution of the IVP at these times.

- This strategy can be generalized to spatial approximations too.

- So the question is, how do we use the solution at $t = 0$ and the slope to get an approximate solution at $t = \Delta t$?
- The way we obtain an approximate solution clearly has to be related to the DE.
- In the DE we replace the derivative with a difference quotient and evaluate the right hand side at the appropriate time level.
- Recall that the forward difference operator is an approximation to $y'(t)$ using the time values at t and $t + \Delta t$. Let's substitute this into our DE

  $$\frac{y(t + \Delta t) - y(t)}{\Delta t} \approx f(y(t), t)$$

  This equation could be solved for $y(t + \Delta t)$ in terms of y at t; i.e.,

  $$y(t + \Delta t) \approx y(t) + \Delta t f(y(t), t)$$

  Now everything on the right hand side is known when $t = 0$ so we could get an approximation to $y(\Delta t)$.

- Notation We will use Y to denote our discrete (approximate) solution. We will add a superscript to denote the time it corresponds to. Consequently

$$Y^n \approx y(t^n)$$

where $y(t)$ is the exact solution to our IVP. We take $Y^0 = y_0$ the initial value at $t = 0$.
- Then our difference approximation to our IVP becomes

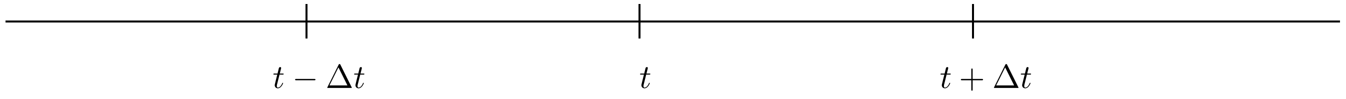
Forward Euler Method for IVP

$$y'(t) = f(y(t), t), \qquad y(0) = y_0$$

$$Y^{n+1} = Y^n + \Delta t f(Y(t^n), t^n) \qquad n = 0, 1, 2, \ldots$$

$$Y^0 = y_0$$

- The name Euler is often used for these first order difference equations.
- The way to remember that it is a forward difference scheme is that we are sitting at the time t and differencing forward in time to $t + \Delta t$ whereas in a backward Euler we are sitting at the point t and differencing backward in time to $t = t - \Delta t$.



Example

Consider the IVP

$$\frac{dy}{dt} = y + t \qquad y(0) = 2$$

whose exact solution is $y = 3e^t - t - 1$ since $y' = 3e^t - 1 = (3e^t - t - 1) + t = y(t) + t$ and $y(0) = 3e^0 - 1 = 2$. Approximate the solution at $t = 1$ using a Forward Euler approximation with $\Delta t = .2$ and calculate the error.

Note that to get the approximate solution at $t = 1$ we need to get the solution at $t = .2, .4, .6, .8$ first. In our example $f(y,t) = t + y$. We will denote Y^1 as our approximation to $y(.2)$, Y^2 as our approximation to $y(.4)$, etc.

$$y(.2) \approx Y^1 = Y^0 + \Delta t(0 + Y^0) \implies Y^1 = 2 + .2(2) = 2.4$$

$$y(.4) \approx Y^2 = Y^1 + \Delta t(.2 + Y^1) \implies Y^2 = 2.4 + .2(.2 + 2.4) = 2.922.88$$

$$y(.6) \approx Y^3 = Y^2 + \Delta t(.4 + Y^2) \implies Y^3 = 3.584$$

$$y(0.8) \approx Y^4 = Y^3 + \Delta t(.6 + Y^3) \implies Y^4 = 4.4208$$

$$y(1.0) \approx Y^5 = Y^4 + \Delta t(.8 + Y^4) \implies Y^5 = 5.46496$$

The exact solution at $t = 1$ is $e^1 = 6.15485$ so our error is 0.68989 which is quite large.

If we repeat the calculation reducing Δt then we get the following results

Δt	n, the number of steps	Y^n	error
0.2	5	5.46496	0.689885
0.1	10	5.78123	0.373618
0.05	20	5.95989	0.194952
0.025	40	6.05519	0.099654
0.0125	80	6.10445	0.0503907

- So as Δt becomes smaller, our error becomes smaller.
- One thing to note about this example (which is true in general) is that as time increases our error grows. For example, in the above calculations the error at $t = .6$ is much smaller than the error at $t = 1$.
- This is because we are actually make two types of errors.
- We are making one type of error because we are replacing a derivative with a difference quotient. We know that this error is $\mathcal{O}(\Delta t)$.
- However, after calculating Y^1 we are making another error. When we calculate Y^1 we are using the exact value of $Y^0 = y_0$ whereas when we calculate Y^2 we are using our approximate value for $y(\Delta t)$ given by Y^1. This is repeated in subsequent steps and our error grows.
- We will return to looking at this error after we implement the method and consider both a local error and a global error.

Implementing Forward Euler

- The implementation should be clear from our previous example.
- We need to know
  - y_0
  - the final time
  - the number of steps (from which we can determine Δt)
  - a function routine for determining the right hand side $f(y,t)$
- As far as storage goes we could either store our approximation at every time or we could overwrite.
- Typically we will overwrite and write our solution to a file for graphing.
- So basically we just have to loop over the number of steps and implement our algorithm; in our loop we have

```
t = t + deltat
ynew = yold + deltat * rhs(yold, t)
```

where `rhs` is our function for the right hand side and `t` has been initially set to zero and `deltat` computed from the final time and the number of steps.

- We would then write off the time and `ynew` and to get ready for the next step we set `yold = ynew` since we are overwriting.
- For classwork you will write a function to get `ynew` from the values `yold, t, dt` and test it on our example we did where the rhs was $y+t$. In addition to calculating an error, you can plot the exact solution and its approximation too.

Higher Order Taylor Series Methods

- We can derive more accurate methods based on the Taylor Series by keeping more terms.
- For example, we used the Taylor Series expansion

$$y(t+\Delta t) = y(t) + y'(t)\Delta t + \mathcal{O}(\Delta t^2)$$

  to derive Euler's method.
- If we keep another term in the series we have

$$y(t+\Delta t) = y(t) + y'(t)\Delta t + y''(t)\frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3)$$

- In order to solve this for $y'(t)$ we need an expression for $y''(t)$.
- In some cases this is easy to obtain since

$$y'(t) = f(t,y) \implies y''(t) = \frac{d}{dt}(f(t,y) = \frac{df}{dt}\frac{dt}{dt} + \frac{df}{dy}\frac{dy}{dt} = f_t + f_y y'$$

  where we have used the chain rule.

Runge Kutta (RK) Methods for IVPs

- Recall that our goal is to now find methods (other than those obtained by keeping more terms in the Taylor series) which give more accurate results than forward Euler for our IVP

Initial Value Problem (IVP)

$$\frac{dy}{dt} = f(t, y) \quad t_0 < t < T$$

$$y(t_0) = y_0$$

- Recall that to derive higher order Taylor series method we had to repeatedly differentiate $f(t, y)$.

- The Runge Kutta (RK) methods are a family of methods that do not require differentiating $f(t, y)$.
- Recall that in Euler's Method we simply use the solution at t^n and f evaluated here to estimate the solution at t^{n+1}.
- The basic idea in RK is that we will sample f at several judiciously chosen points in $[t^n, t^{n+1})$ and use this information to more accurately estimate the solution at t^{n+1}.

Midpoint Method

- The simplest RK is the midpoint method where we estimate the slope (i.e., f) at the midpoint of $[t^n, t^{n+1}]$ and then use this to estimate the solution at t^{n+1} using Euler's method.
- To estimate the slope at the midpoint $t^{n+\frac{1}{2}} = t^n + \frac{\Delta t}{2}$ we take f evaluated at $t^{n+\frac{1}{2}}$ and an estimate to the solution at $t^{n+\frac{1}{2}}$, i.e.,

$$f(t^n + \frac{\Delta t}{2}, Y^{n+\frac{1}{2}}).$$

  We don't have $Y^{n+\frac{1}{2}}$ but as an estimate we take an Euler step of length $\frac{\Delta t}{2}$.
- Recall that the solution at t^{n+1} predicted by Euler's method is

$$Y^{n+1} = Y^n + \Delta t f(t^n, Y^n)$$

  so that we approximate $Y^{n+\frac{1}{2}}$ by

$$Y^n + \frac{\Delta t}{2} f(t^n, Y^n)$$

- Now we estimate the solution at t^{n+1} by Euler's method where we use the approximation to the slope at $t^{n+\frac{1}{2}}$. We obtain

$$Y^{n+1} = Y^n + \Delta t f\left(t^{n+\frac{1}{2}}, Y^n + \frac{\Delta t}{2} f(t^n, Y^n)\right)$$

- The standard way that this method is written is

$$k_1 = \Delta t f(t^n, Y^n)$$

$$k_2 = \Delta t f(t^n + \frac{\Delta t}{2}, Y^n + \frac{1}{2} k_1)$$

$$Y^{n+1} = Y^n + k_2$$

Example

Approximate the solution to the IVP

$$\frac{dy}{dt} = t + y \qquad y(0) = 2$$

at $T = 1$ using 5 equal timesteps in the Midpoint RK method. Here $f(t, y) = t + y$.

If we follow the steps of the algorithm above we have as an approximation to $y(\Delta t) = y(.2)$

$$k_1 = \Delta t f(t^0, Y^0) = .2f(0, 2) = .2(0 + 2) = .4$$

$$k_2 = \Delta t f(\frac{\Delta t}{2}, Y^0 + \frac{1}{2}k_1) = .2(f(.1, 2 + .5(.4)) = .2f(.1, 2.2) = .2(.1 + 2.2) = .46$$

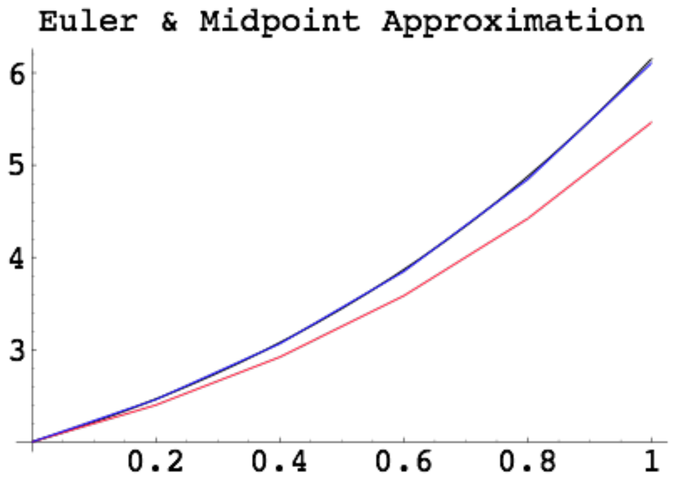
$$Y^1 = Y^0 + k_2 = 2 + .46 = 2.36$$

The actual error here is 0.00420827.

Continuing in this manner we get the following table of results:

n	t^n	Y^n	$y(t^n)$	error
1	0.2	2.46	2.4621	0.004208
2	0.4	3.0652	3.07547	0.010274
3	0.6	3.84754	3.863	0.01881
4	0.8	4.846	4.87662	0.030619
5	1.0	6.10812	6.1548	0.046721

If we compare this error with the results from Euler's Method we can see that it is much better. In the plot below the RK and the exact solution lie almost on top of each other whereas Euler's method starts to deviate quickly.



Deriving a RK Method to Get a Second Order Scheme

- In the Midpoint Method we chose an "easy" point in the interval $[t^n, t^{n+1}]$, i.e., the midpoint $t^{n+\frac{1}{2}}$. We could derive the error estimate for this method but we are going to do something more general.
- We are going to take the approach that instead of using the midpoint $t^{n+\frac{1}{2}}$ we are going to find the point in $[t^n, t^{n+1}]$ which gives us the most accurate method.
- Here we are deriving a two-stage RK method since we are using information at t^n and at another point in our interval $[t^n, t^{n+1}]$.
- This is the way that most useful RK schemes are derived.
- Recall that the midpoint rule was written as

$$k_1 = \Delta t f(t^n, Y^n)$$

$$k_2 = \Delta t f(t^n + \frac{\Delta t}{2}, Y^n + \frac{1}{2} k_1)$$

$$Y^{n+1} = Y^n + k_2$$

Instead of using the midpoint we will use an arbitrary point $t^n + \alpha\Delta t$, for $0 \leq \alpha \leq 1$. Likewise instead of evaluating f in the second step at $y = Y^n + \frac{1}{2}k_1$ we will use $Y^n + \beta k_1$. Moreover instead of setting $Y^{n+1} = Y^n + 0 \cdot k_1 + 1 \cdot k_2$ we will use the general expression $Y^{n+1} = Y^n + ak_1 + bk_2$. We have

$$k_1 = \Delta t f(t^n, Y^n)$$

$$k_2 = \Delta t f(t^n + \alpha\Delta t, Y^n + \beta k_1)$$

$$Y^{n+1} = Y^n + ak_1 + bk_2$$

- Our goal is the find α, β, a, b so that the method is as accurate as we can achieve; in this case it is second order.

- To do this we return to our Taylor series expansion with remainder for $y(t^n + \Delta t)$

$$y(t^{n+1}) = y(t^n) + y'(t^n)\Delta t + y''(t^n)\frac{\Delta t^2}{2} + y'''(\xi_n)\frac{\Delta t^3}{6}$$

- Now we want to take this expansion and subtract our expansion for Y^{n+1} to get the highest power of Δt, i.e., the highest order method possible with only using information from two points.

- To do this we would like to relate the derivatives of $y(t)$ to $f(t,y)$. Clearly $y'(t) = f(t,y)$ but what about $y''(t)$? We know that
$$y''(t) = \frac{d}{dt}y'(t) = \frac{d}{dt}f(t,y)$$
and since f is a function of both t and $y(t)$ we use the chain rule to get
$$y''(t) = \frac{\partial f}{\partial t}\frac{\partial t}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt} = f_t + f_y f$$
Thus we can write our Taylor series expansion as
$$y(t^{n+1}) = y(t^n) + f\Delta t + (f_t + f_y f)\frac{\Delta t^2}{2} + \mathcal{O}\Delta t^3$$
where we have left off the fact that f and $f_t + f_y f$ are explicitly evaluated at $t^n, y(t^n)$ for brevity.

- To make our truncation error second order we have to have
$$y(t^{n+1}) - Y^{n+1} = \mathcal{O}(\Delta t^3)$$
where Y^{n+1} is computed using the exact solution $y(t^n)$. Plugging $y(t^n)$ into our RK for Y^n we have
$$Y^{n+1} = y(t^n) + ak_1 + bk_2$$

where now $k_1 = \Delta t f(t^n, y(t^n))$ and

$$k_2 = \Delta t f\Big(t^n + \alpha \Delta t, y(t^n) + \beta k_1\Big)$$

Now k_1 is in a form we can subtract from our Taylor series but k_2 is not. So we expand $f\Big(t^n + \alpha \Delta t, y(t^n) + \beta k_1\Big)$ in a Taylor series in each component. We can either do this twice, the first time holding the second component first and then holding the first component fixed or we can use a Taylor series expansion for two independent variables. We get

$$f(t^n + \alpha \Delta t, z) = f(t^n, z) + \alpha \Delta t f_t(t^n, z) + \mathcal{O}(\Delta t^2)$$

Here we have set $z = y(t^n) + \beta k_1$ for shorthand notation. Note that this term $f(t^n + \alpha \Delta t, z)$ is multiplied by Δt in the definition of k_2 so we only need to keep terms through Δt^2. Now we expand each of these terms in the second argument z. We have

$$f(t^n, y(t^n) + \beta k_1) = f(t^n, y(t^n)) + \beta k_1 f_y(t^n, y(t^n)) + \mathcal{O}(\Delta t^2)$$

and

$$\alpha \Delta t f_t(t^n, y(t^n) + \beta k_1) = \alpha \Delta t f_t(t^n, y(t^n)) + \mathcal{O}(\Delta t^2)$$

Again we only kept terms through $\mathcal{O}(\Delta t^2)$ since the whole expansion is multiplied by Δt.

- Combining these we get the following expression for k_2

$$k_2 = \Delta t\Big[f\big(t^n, y(t^n)\big) + \beta k_1 f_y(t^n, y(t^n)) + \alpha \Delta t f_t(t^n, y(t^n))\Big] + \mathcal{O}(\Delta t^3)$$

- We are now ready to subtract our expansions for Y^{n+1} and $y(t^{n+1})$. Recapping, we have the following where f and its derivatives are all evaluated at $(t^n, y(t^n))$ (I have left this off for clarity of exposition)

$$\begin{aligned} Y^{n+1} &= y(t^n) + ak_1 + b\Delta t\big[f + \beta k_1 f_y + \alpha \Delta t f_t\big] + \mathcal{O}(\Delta t)^3 \\ &= y(t^n) + a\Delta t f + b\Delta t\big[f + \beta \Delta t f f_y + \alpha \Delta t f_t\big] + \mathcal{O}(\Delta t)^3 \end{aligned}$$

and

$$y(t^{n+1}) = y(t^n) + f\Delta t + (f_t + f_y f)\frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3)$$

Subtracting yields

$$y(t^{n+1}) - Y^{n+1} = f\Delta t(1-a-b) + \Delta t^2 f_t\Big(\frac{1}{2} - b\alpha\Big) + f f_y \Delta t^2\Big(\frac{1}{2} - b\beta\Big) + \mathcal{O}(\Delta t^3)$$

So to make $y(t^{n+1}) - Y^{n+1} = \mathcal{O}(\Delta t^3)$ we need all the terms involving lower powers of Δt to disappear, i.e., we need

$$a + b = 1, \qquad 2\alpha b = 1 \qquad 2\beta b = 1$$

- These equations are under-determined and have an infinite number of solutions.
- For the midpoint rule we have $a = 0$, $b = 1$, $\alpha = \frac{1}{2}$, $\beta = \frac{1}{2}$ which satisfy the equations. Thus the midpoint rule is a second order method.
- However, the usual choice is $a = b = \frac{1}{2}$ and $\alpha = \beta = 1$.
- We take our second order scheme as follows.

Classical Second Order Runge Kutta Method

$$k_1 = \Delta t f(t^n, Y^n)$$

$$k_2 = \Delta t f(t^{n+1}, Y^n + k_1)$$

$$Y^{n+1} = Y^n + \frac{1}{2}(k_1 + k_2)$$

Runge Kutta Methods for Solving Systems of IVPs

- The concept of generalizing RK methods for solving systems of IVPs is similar to what we did for Euler's method.
- Consider the system of two IVPs

  $$u'(t) = f(t, u, v) \qquad v'(t) = g(t, u, v)$$

  where

  $$u(t^0) = u_0, \quad v(t^0) = v_0 \,.$$

- Since each equation has a different right hand side, we have to compute the k_i for each equation.
- Suppose that we are using the midpoint rule which is a second order scheme. Recall that here we have for a single IVP

$$k_1 = \Delta t f(t^n, Y^n) \qquad k_2 = \Delta t f(t^n + \frac{1}{2}, Y^n + \frac{1}{2} k_1), \qquad Y^{n+1} = Y^n + k_2$$

- To distinguish the two sets of coefficients lets call the terms for u, k_i and those for v, m_i.

- Lets look at how we compute U^1, V^1. We first compute k_1 and m_1

$$k_1 = \Delta t f(t^0, U^0, V^0), \qquad m_1 = \Delta t g(t^0, U^0, V^0)$$

- Now we compute k_2, m_2

$$k_2 = \Delta t f(t^0 + \frac{\Delta t}{2}, U^0 + \frac{1}{2}k_1, V^0 + \frac{1}{2}m_1), \quad m_2 = \Delta t g(t^0 + \frac{\Delta t}{2}, U^0 + \frac{1}{2}k_1, V^0 + \frac{1}{2}m_1),$$

- Note that we can not compute all the k_i's and then all the m_i's. Why?

- Finally we compute the solution

$$U^1 = U^0 + k_2 \qquad V^1 = V^0 + m_2$$

- Example. Consider the system

$$u' = v \quad v' = -\frac{2}{t}v$$

$$u(1) = 10 \qquad v(1) = 1$$

Lets do one step of length 0.2 using the midpoint RK scheme.

For U^1 we have

$$k_1 = .2f(1, U^0, V^0) = .2 * V^0 = .2 \quad \text{since } f = v$$

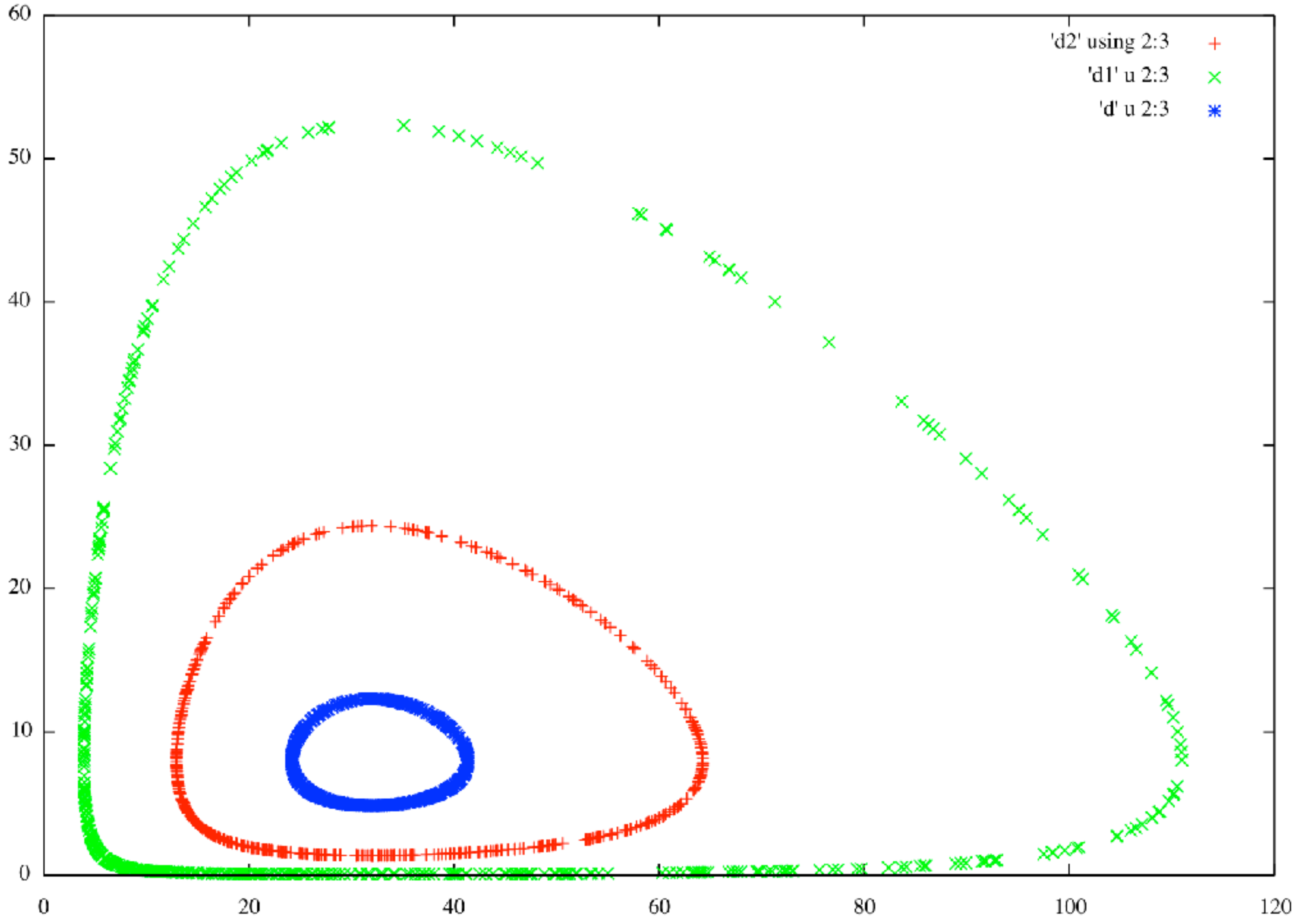
$$m_1 = .2g(1, U^0, V^0) = .2 * \left(\frac{-2}{1}\right) * V^0 = -0.4 \quad \text{since } g = -\tfrac{2}{t}v$$

$$k_2 = .2f(1.1, U^0 + \frac{k_1}{2}, V^0 + \frac{m_1}{2}) = .2(1 + (-.4)/2) = 0.16$$

$$m_2 = .2f(1.1, U^0 + \frac{k_1}{2}, V^0 + \frac{m_1}{2}) = .2(\frac{-2}{1.1}).8 = -0.2909$$

$$U^1 = U^0 + k_2 = 10 + 0.16 = 10.16$$

$$V^1 = V^0 + m_2 = 1 - 0.2909 = 0.709091$$


'd2' using 2:3 +
'd1' u 2:3 ×
'd' u 2:3 ✱
0
10
20
30
40
50
60
0
20
40
60
80
100
120

Local and Global Errors for Forward Euler

Local truncation error

- The local truncation error is the error that we would make in one step if we start with the exact solution.
- Lets take $Y^n = y(t^n)$ (i.e., the exact solution at time t^n) and perform one step of our method. We take our approximation for Y^{n+1} and combine it with our Taylor's Series. We have our difference equation with $Y^n = y(t^n)$

$$Y^{n+1} = y(t^n) + \Delta t f(t, y(t^n)) = y(t^n) + \Delta t y'(t^n)$$

and the Taylor series

$$y(t^n + \Delta t) = y(t^n) + \Delta t y'(t^n) + \frac{\Delta t^2}{2} y''(\xi_n) \quad t^n < \xi_n < t^n + \Delta t$$

where we have used the remainder form of the Taylor's series. Subtracting gives the local truncation error

$$y(t^n + \Delta t) - Y^{n+1} = \frac{\Delta t^2}{2} y''(\xi_n) = \mathcal{O}(\Delta t^2)$$

- This error is due to the fact that we are approximating the derivative by a difference quotient.
- A method is called order k if the local truncation error is $\mathcal{O}(\Delta t^{k+1})$.
- Consequently Euler's method for our IVP is called first order.

Global error

- Separate from the local truncation error is the global error which is the actual difference in our exact solution at t^n and our approximate solution at this time.
- Of course this is the error we are most interested in.
- This error is found by accumulating the errors that we make at each of the steps before t^n.
- It turns out that under certain conditions we can control the local truncation error (by adjusting Δt) to get a desired global error.

- In fact, under certain conditions one can prove that

  $$\text{global error at } t^n \leq \sum_{k=1}^{n} LTE_k$$

  where LTE_k represents the local truncation error at step k.

- So in our case the global error at time t^n is bounded by

  $$\sum_{k=1}^{n} C_i \frac{\Delta t^2}{2}$$

  where we let $C_i = |y''(\xi_i)|$. Now if C is the maximum of C_i then

  $$\sum_{k=1}^{n} C_i \frac{\Delta t^2}{2} \leq \frac{1}{2}\Big[C\Delta t^2 + C\Delta t^2 + \cdots + C\Delta t^2\Big] = \frac{C}{2}(n\Delta t)\Delta t = \frac{C}{2}t^n\Delta t = \widetilde{C}\Delta t$$

  Thus we say our global error is $\mathcal{O}(\Delta t)$ since $\frac{C}{2}t^n$ is a constant.

- The condition that guarantees that we can control the global error by controlling the local truncation error is tied to the concept of stability which basically means that small changes in the initial data produce small changes in the IVP solution. We will assume we can control the global error by controlling the LTE.

How can we control the local truncation error to guarantee a global error less than some tolerance?

- Suppose we want the global error at $T = n\Delta t$ to be less than tol.
- Assume we have a bound M for the second derivative of our solution, i.e.,

  $$|y''(t)| \leq M \qquad 0 \leq t \leq n\Delta t = T$$

  and our local truncation error at each step satisfies

  $$|LTE_k| \leq M\frac{\Delta t^2}{2}$$

- If we can bound the global error at T by the accumulated LTE then

  $$|y(T) - Y^n| \leq \sum_{k=1}^{n} \frac{M}{2}\Delta t^2 = \frac{M}{2} T \Delta t$$

- If we want the global error at time T to be less than some tolerance, i.e.,

  $$|y(T) - Y^n| \leq tol$$

  then we can determine the number of steps (and thus Δt) which guarantees this.

- To make the global error less than our tolerance we simply require

  $$\frac{M}{2} T \Delta t \leq tol$$

  then we are guaranteed that

  $$|y(t^n) - Y^n| \leq tol$$

- Consequently, we choose the number of steps n (and thus Δt) to be the smallest integer that satisfies

  $$\frac{M}{2} \frac{T^2}{n} \leq tol\,,$$

  where we have written $\Delta t = \frac{T}{n}$.

- Consequently

  $$n = \frac{MT^2}{2 \cdot tol}$$

- Of course the only way we can use this is if we have a bound on y''.

- Example

Consider the IVP

$$\frac{dy}{dt} = \sin t \qquad y(0) = -1$$

where we want to approximate the solution at $T = 2$ with an error tolerance of 0.01.

Now $y' = \sin t$ so $y'' = \cos t$ and thus $|y''(t)| \leq 1$ for all t. Using the expression

$$\frac{MT^2}{2n} \leq tol$$

yields

$$\frac{1 \cdot 2^2}{2n} \leq 0.01 \implies \frac{2}{n} \leq 0.01 \implies n \geq \frac{2}{0.01} = 200$$

Below is a summary of the errors at $t = 2$ using a range of steps. Note that our estimate does NOT guarantee that this is the smallest number of steps you can take to get the desired error. Rather it says that if you choose this number of steps, you are guaranteed that the error will be less than the prescribed tolerance. Note that in our case choosing $n = 100$ actually has the global error less than our tolerance.

n	error
25	0.037127
50	0.01837
100	0.00914
150	0.006083
200	0.00456

- How can you incorporate this estimate into a code?
- You need to input your final time, your tolerance and a bound for $y''(t)$. Then you can use the intrinsic function

  ```
  ceiling(a)
  ```

  which gives the smallest integer $\geq a$.
- For example

  ```
  n = ceiling ( M * T * T / ( two * tol ) )
  ```