Simple numerical method for integrating EOM

For those of you without much experience solving equations numerically, I'll write some details of a simple method for solving EOMs. This method can be implemented within a spreadsheet or with a simple program (C, fortran, etc). There are many other ways to find such solutions using canned routines or packages such as Mathematica, of course, but there is a lot to learn by implementing a simple scheme yourself. If you'd like to try, then read on...

The most basic method for integrating a first order differential equation is Euler's method. It's not the most efficient and can sometimes fail, but it is transparent. In general, a differential equation can be written

$$\frac{dy}{dx} = f(x,y).$$

Our goal is to find $y(x)$ for $x \in [x_a, x_b]$, and

we know $y(x_a)$ (initial value). To do this on the computer, we need to discretize x:

$$x = x_1, x_2, \dots, x_N$$

such that

$$x_{i+1} - x_i = \Delta x,$$

$$x_1 = x_a \quad \& \quad x_N = x_b.$$

Then, we find $y(x_i)$. We thus take the diff. eq'n and write

$$\left.\frac{dy}{dx}\right|_{x=x_i} = f(x,y)\Big|_{x=x_i}$$

There are a few ways to evaluate the LHS, but the Euler prescription uses

$$y(x_i+\Delta x) = y(x_i) + \left.\frac{dy}{dx}\right|_{x_i}\Delta x + \frac{1}{2}\left.\frac{d^2y}{dx^2}\right|_{x_i}\Delta x^2 + \dots$$

$$\Rightarrow \left.\frac{dy}{dx}\right|_{x_i} = \frac{y(x_i+\Delta x) - y(x_i)}{\Delta x} - \frac{1}{2}\left.\frac{d^2y}{dx^2}\right|_{x_i}\Delta x + \dots$$

The differential equation becomes the difference equation

$$\frac{y(x_i+\Delta x) - y(x_i)}{\Delta x} + \underbrace{\mathcal{O}(\Delta x)}_{\text{local error term!}} = f(x_i, y(x_i))$$

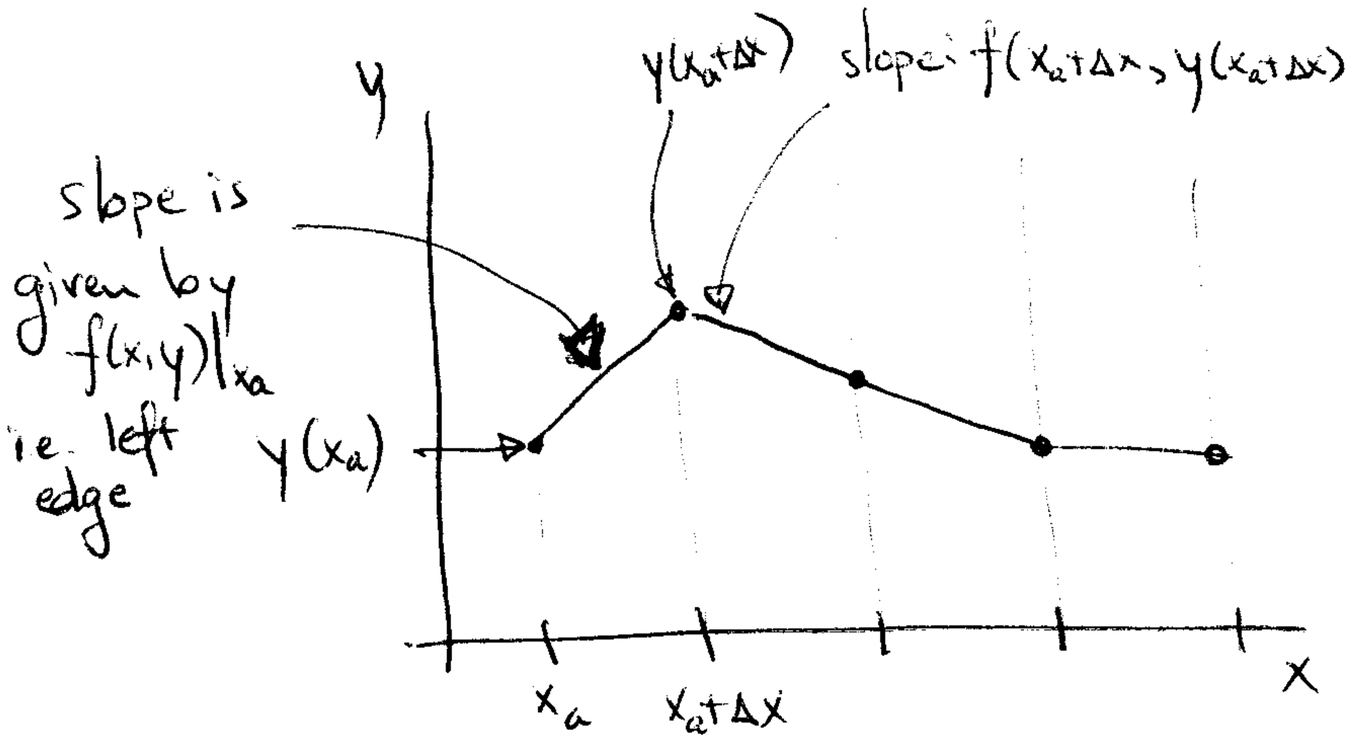
or

$$y(x_i+\Delta x) = y(x_i) + f(x_i, y(x_i))\Delta x.$$

Since we know $y(x_a)$, we can calculate $y(x_a+\Delta x)$, then

$y(x_a+2\Delta x)$, and so on recursively. In the limit $\Delta x \to 0$, this approach should be exact.

It's good to have a picture to go along with the method, so here it is:



Maybe you're asking yourself how this helps with EOMs which are 2nd order. Here's how:

$$m\ddot{x} = F \longrightarrow \begin{cases} \dot{x} = v \\ m\dot{v} = F \end{cases}$$

The 2nd order eq'n becomes two 1st order — which you know how to solve. In particular,

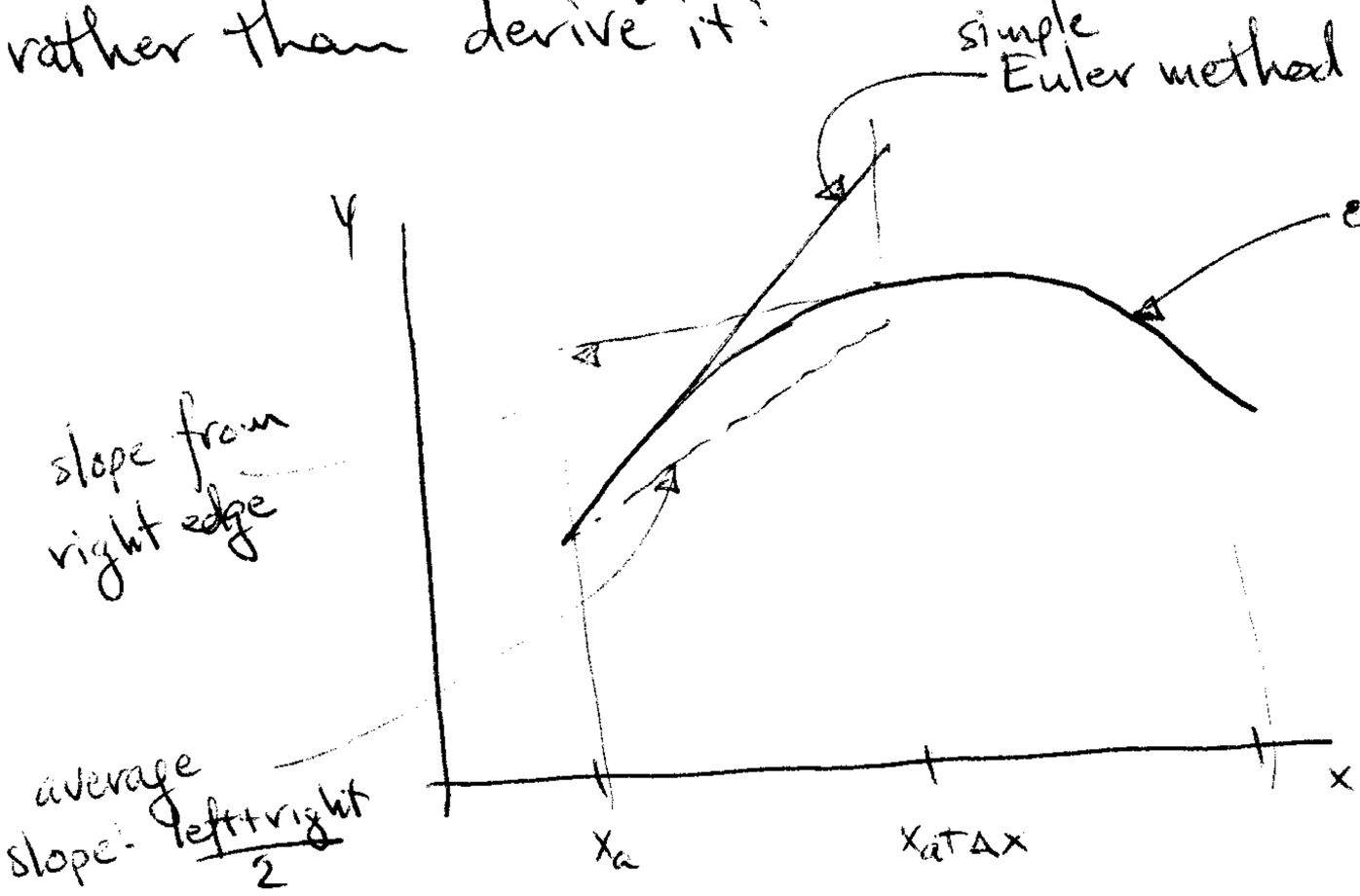
$$v_{i+1} = \frac{F_i}{m}\Delta t + v_i$$

$$x_{i+1} = v_i \Delta t + x_i$$

shorthand for $v(t_i+\Delta t)$

you know everything on this side!

It turns out for the Kepler problem (including our three-body HW problem), the simple Euler method is not quite good enough. Fortunately, a slight modification makes it work quite well. I'll explain it with a picture rather than derive it:



In other words, it's just like Euler, except the slope used to calculate $y(x_a+\Delta x)$ from $y(x_a)$ is not evaluated at the left edge (i.e. x_a) but rather in the middle of the interval ($x_a+\frac{1}{2}\Delta x$). This is essentially the trapezoidal rule compared to the endpoint rule of the simple Euler method.

As an algorithm, this idea is implemented as follows:

$$y'_{i+1} = y_i + f_i \Delta x$$

prime does not mean derivative here

← Euler step to get approx. value of y_{i+1}

$$f_{i+1} = f(x_{i+1}, y'_{i+1})$$

← gives estimate of slope from right edge

$$y_{i+1} = y_i + \frac{1}{2}(f_i + f_{i+1})\Delta x$$

This method can be adapted to the EOM in almost exactly the same way as for the simple Euler (see p. 3). It is useful to write the final expressions (i.e. last line of algorithm ~~of~~ on p. 3a) for propagating the EOM:

$$v_{i+1} = v_i + \frac{1}{2}\left(\frac{F_i}{m} + \frac{F_{i+1}}{m}\right)\Delta t$$ ← acceleration $(\frac{F}{m})$ not constant, but approx. by its average

$$x_{i+1} = x_i + \frac{1}{2}\left(v_i + v_{i+1}\right)\Delta t$$ ← exact if acceleration is constant over Δt

A couple of pointers:

1) Adjust the timestep as you go along. When the force or velocity gets large, ~~the~~ Euler's method will fail. This is because the error term in the derivatives - which we are neglecting - tends to get large. A good rule of thumb is that you don't want either v or x to change too much in one time step:

$$\Delta v_i = \frac{F_i \Delta t}{m} \leq \Delta v_{max} \Rightarrow \Delta t = \frac{m \Delta v_{max}}{F_i}$$

$$\Delta x_i = v_i \Delta t \leq \Delta x_{max} \Rightarrow \Delta t = \frac{\Delta x_{max}}{v_i}$$

This gives a prescription for choosing the new Δt at every time step. Keep in mind that you can have only <u>one</u> Δt, so you should choose the smaller of the two possibilities above. You'll have to adjust Δx_{max} & Δv_{max} until you get the accuracy you want without too many steps.
Per the discussion on p. 2, the limits $\Delta x_{max} \to 0$ and/or $\Delta v_{max} \to 0$ imply $\Delta t \to 0$ which should give the exact answer.

2) If you have a 2D problem, then the (simple Euler) scheme generalizes as

$$v_{x,i+1} = v_{x,i} + \frac{F_i}{m}\Delta t \qquad v_{y,i+1} = v_{y,i} + \frac{F_i}{m}\Delta t$$

$$x_{i+1} = x_i + v_{x,i}\Delta t \qquad y_{i+1} = y_i + v_{y,i}\Delta t$$

3) Anytime you use an algorithm like these to solve a new problem, you should test it. The best way to test is to solve a problem that you know the answer to. For the three-body HW problem, for instance, making sure you can solve the Earth-Sun system reasonably well is a good idea. Also, making sure that quantities that should be conserved are actually conserved by your calculation. Make sure, too, that something like the Earth-Sun system is stable for long times (however you define that for your problem).