General Second-Order Runge-Kutta — Theory

January 31, 2025

Second-Order Runge-Kutta — Recap

This is the version of second-order Runge-Kutta that we derived and used on January 29:

$t_{i+1} = t_i + \Delta t$

$x^*(t_{i+1}) = x(t_i) + v(t_i)\,\Delta t$ (using $v(t_i)$ to extrapolate all the way to t_{i+1})

$v(t_{i+1}) = v(t_i) + (a(x(t_i)) + a(x^*(t_{i+1}))) \cdot \frac{\Delta t}{2}$

$x(t_{i+1}) = x(t_i) + (v(t_i) + v(t_{i+1})) \cdot \frac{\Delta t}{2}$

The intermediate quantity x^* is computed and used, but it is not recorded as part of the results of the current time step or handed off to the next time step. It is computed, used, and then forgotten.

Is this *Ad Hoc?*

We noted that the choice of x^* felt *ad hoc.* Why extrapolate to the endpoint? Why use trapezoid for $a_{i \to i+1,\,\text{avg}}$? For example, why not this procedure instead:

$t_{i+1} = t_i + \Delta t$

$x^* = x(t_i) + v(t_i) \cdot \frac{\Delta t}{2}$ (using $v(t_i)$ to extrapolate to $t_i + \frac{\Delta t}{2}$)

$v(t_{i+1}) = v(t_i) + a(x^*) \cdot \Delta t$

$x(t_{i+1}) = x(t_i) + (v(t_i) + v(t_{i+1})) \cdot \frac{\Delta t}{2}$

You see that this is midpoint approximation for $a_{i \to i+1,\,\text{avg}}$ instead of trapezoid, with the midpoint value for x being the crude approximation using the left estimate velocity you see in the formula for x^*.

To summarize, in our first version of Runge-Kutta we let $x^*(t_{i+1})$ be an estimate of the final position by using $v(t_i)$ to extrapolate all the way to t_{i+1}.

Just above, I suggested an alternative version: let x^* be an estimate of the midpoint position by using $v(t_i)$ to extrapolate to $t_i + \frac{\Delta t}{2}$.

This hints at a way that we can generalize. Pick a number pick a number α between 0 and 1, and let

$x^* = x(t_i) + v(t_i) \cdot \alpha \Delta t$ (using $v(t_i)$ to extrapolate to $t_i + \alpha \Delta t$)

Before I write down the rest of the generalized procedure, let's generalize even further.

General Second-Order Runge-Kutta

If we are going to do general second-order Runge-Kutta, let's pull out all of the stops. In addition to introducing the new parameter α, let's also let the force depend (i) directly on the time, (ii) indirectly on the time via the position, and (iii) indirectly on the time via the velocity.

Dependency of type (i) will let us do externally-applied time-dependent forces. Dependency of type (iii) will let us do problems with friction, drag, and magnetism.

To emphasize that the force (or the acceleration) can depend on all three of (i), (ii), and (iii), we will now write:

$a(t, x(t), v(t)) = \frac{F(t,x(t),v(t))}{m}$

Now that we have overcome our previous simplifications, here is general Second-Order Runge-Kutta:

$t^* = t + \alpha \Delta t$

$x^* = x(t_i) + v(t_i) \cdot \alpha \Delta t$

$v^* = v(t_i) + a(t_i, x(t_i), v(t_i)) \cdot \alpha \Delta t$

$t_{i+1} = t_i + \Delta t$

$v(t_{i+1}) = v(t_i) + \left(\left(1 - \frac{1}{2\alpha}\right) a(t_i, x(t_i), v(t_i)) + \frac{1}{2\alpha} a(t^*, x^*, v^*)\right) \cdot \Delta t$

$x(t_{i+1}) = x(t_i) + \left(\left(1 - \frac{1}{2\alpha}\right) v(t_i) + \frac{1}{2\alpha} v^*\right) \cdot \Delta t$

Sanity Checks with $\alpha = 1$ and $\alpha = \frac{1}{2}$

I adapted the preceding from Eqs. 231a and 231b on p. 87 of Butcher, *Numerical Methods for Ordinary Differential Equations,* Wiley, 2003.

Before we go much further with it, as sanity checks, let's see if we can recover the trapezoid and mid-point versions of Second-Order Runge-Kutta that we previously advocated.

$\alpha = 1$

Let's just put in $\alpha = 1$. We get:

$t^* = t + \Delta t$

$x^* = x(t_i) + v(t_i) \cdot \Delta t$

$v^* = v(t_i) + a(t_i, x(t_i), v(t_i)) \cdot \Delta t$

$t_{i+1} = t_i + \Delta t$

$v(t_{i+1}) = v(t_i) + (a(t_i, x(t_i), v(t_i)) + a(t^*, x^*, v^*)) \cdot \frac{\Delta t}{2}$

$x(t_{i+1}) = x(t_i) + (v(t_i) + v^*) \cdot \frac{\Delta t}{2}$

That's pretty darned close to what we previously advocated as trapezoid.

The subtle difference is in the formula for $x(t_{i+1})$ which is arguably not as good because we now have $(v(t_i) + v^*)/2$ in the trapezoid estimate of $v_{i \to i+1,\text{avg}}$ instead of the presumably better $(v(t_i) + v(t_{i+1}))/2$.

$\alpha = \frac{1}{2}$

Now put in $\alpha = \frac{1}{2}$. We get:

$t^* = t + \frac{\Delta t}{2}$

$x^* = x(t_i) + v(t_i) \cdot \frac{\Delta t}{2}$

$v^* = v(t_i) + a(t_i, x(t_i), v(t_i)) \cdot \frac{\Delta t}{2}$

$t_{i+1} = t_i + \Delta t$

$v(t_{i+1}) = v(t_i) + a(t^*, x^*, v^*) \cdot \Delta t$

$x(t_{i+1}) = x(t_i) + v^* \cdot \Delta t$

That's pretty darned close to what we previously advocated as a midpoint/trapezoid hybrid.

Again, the subtle difference is in the formula for $x(t_{i+1})$ which is arguably not as good because we now have v^* for the midpoint in the estimate of $v_{i \to i+1,\text{avg}}$ instead of the presumably better $(v(t_i) + v(t_{i+1}))/2$.

General Second-Order Runge-Kutta — Conclusion

$$t^* = t + \alpha \Delta t$$

$$x^* = x(t_i) + v(t_i) \cdot \alpha \Delta t$$

$$v^* = v(t_i) + a(t_i, x(t_i), v(t_i)) \cdot \alpha \Delta t$$

$$t_{i+1} = t_i + \Delta t$$

$$v(t_{i+1}) = v(t_i) + \left(\left(1 - \frac{1}{2\,\alpha}\right) a(t_i, x(t_i), v(t_i)) + \frac{1}{2\,\alpha}\, a(t^*, x^*, v^*)\right) \cdot \Delta t$$

$$x(t_{i+1}) = x(t_i) + \left(v(t_i) + v(t_{i+1})\right) \frac{\Delta t}{2}$$

After doing the sanity checks and comparing the new and previously-contemplated formulas, I have opted to restore the trapezoid for $x(t_{i+1})$, even though it is not quite what is specified in Eq. 231b of Butcher, *Numerical Methods for Ordinary Differential Equations,* Wiley, 2003.

Epilog — Euler's Method

We've done a lot of things, and I have paid very little attention to the simplest method, which is Euler's Method.

Before I get to Euler's Method, I should explain why what we are doing above is called "second order." Usually the reason given for the nomenclature is that $a(t, x(t), v(t))$ must be evaluated twice per time step, at $a(t_i, x(t_i), v(t_i))$ and $a(t^*, x^*, v^*)$. Looking ahead, we will eventually get to "fourth order," and then, as you might expect, to get $x(t_{i+1})$ and $v(t_{i+1})$ from $x(t_i)$ and $v(t_i)$, we will have to evaluate $a(t, x(t), v(t))$ four times per time step.

In this sense, the first and simplest method one might try is "first order," where you only evaluate $a(t, x(t), v(t))$ once per time step, at $a(t_i, x(t_i), v(t_i))$:

$$t_{i+1} = t_i + \Delta t$$

$$v(t_{i+1}) = v(t_i) + a(t_i, x(t_i), v(t_i))\, \Delta t$$

$$x(t_{i+1}) = x(t_i) + v(t_i)\, \Delta t$$

That is called "Euler's Method" and you can see that Euler's method is what we used to get x^* and v^* above, except that we used Euler's Method to advance by an amount $\alpha \Delta t$ rather than Δt.