CS205 – Class 17

Covered in class: All
Reading: 9.3.9

1. Backward (Implicit) Euler $\frac{y_{k+1} - y_k}{h} = f(t_{k+1}, y_{k+1})$
   a. 1^{st} order accurate
   b. Backward Euler applied to the model equation $y' = \lambda y$ is $y_{k+1} = y_k + h\lambda y_{k+1}$
      i. So $y_{k+1} = (1 - h\lambda)^{-1} y_k$ and $y_k = (1 - h\lambda)^{-k} y_o$
      ii. The error shrinks when $|(1 - h\lambda)^{-1}| < 1$
      iii. Thus, $-\infty < h\lambda < 0$ is needed for stability
      iv. i.e. stable for all h or unconditionally stable
   c. Generally need to solve a nonlinear equation to find y_{k+1}
      i. Can use Newton iteration, i.e. linearize, solve, linearize, solve, etc.
      ii. For some applications, only one linearize and solve cycle is used
   d. One can take very large time steps since it is stable
      i. However it is not very accurate
      ii. As $h \to \infty$, $y_{k+1} = y_k + h\lambda y_{k+1} \to 0 = 0 + h\lambda y_{k+1}$ or $y_{k+1} = 0$
      iii. This is the long run solution for $y' = \lambda y$ with $\lambda < 0$, but we get there too fast
      iv. Everything damps out too quick, i.e. not accurate
2. **<u>Trapezoidal rule</u>** $\frac{y_{k+1} - y_k}{h} = \frac{f(t_k, y_k) + f(t_{k+1}, y_{k+1})}{2}$
   a. 2^{nd} order accurate
   b. Trapezoidal rule applied to the model equation is $y_{k+1} = y_k + \frac{h\lambda}{2}\left(y_k + y_{k+1}\right)$
      i. So $y_{k+1} = (1 + h\lambda/2)/(1 - h\lambda/2) y_k$ and $y_k = (1 + h\lambda/2)^k/(1 - h\lambda/2)^k y_o$
      ii. The error shrinks when $|(1 + h\lambda/2)/(1 - h\lambda/2)| < 1$
      iii. Thus, $-\infty < h\lambda < 0$ is needed for stability
      iv. i.e. unconditionally stable
   c. Generally need to solve a nonlinear equation to find y_{k+1}
      i. One can take very large time steps since it is stable
      ii. However it is not very accurate
      iii. As $h \to \infty$, $y_{k+1} = y_k + \frac{h\lambda}{2}\left(y_k + y_{k+1}\right) \to 0 = 0 + \frac{h\lambda}{2}\left(y_k + y_{k+1}\right)$ or $y_{k+1} = -y_k$
      iv. This is NOT the long time solution for $y' = \lambda y$
      v. Bad oscillatory behavior
3. **<u>1^{st} order Runge-Kutta</u>** is Euler's method $\frac{y_{k+1} - y_k}{h} = f(t_k, y_k)$

4. **2nd order Runge-Kutta** $\frac{y_{k+1}-y_k}{h}=\frac{k_1+k_2}{2}$
    a. $k_1 = f(t_k, y_k)$ and $k_2 = f(t_k+h, y_k+hk_1)$
5. **4th order Runge-Kutta** $\frac{y_{k+1}-y_k}{h}=\frac{k_1+2k_2+2k_3+k_4}{6}$
    a. $k_1 = f(t_k, y_k)$, $k_2 = f(t_k+h/2, y_k+hk_1/2)$, $k_3 = f(t_k+h/2, y_k+hk_2/2)$ and $k_4 = f(t_k+h, y_k+hk_3)$
6. **TVD Runge Kutta**
    a. 1st order accurate TVD RK is Euler's method
    b. 2nd order accurate TVD RK is the standard second order accurate RK scheme
        i. Also known as the midpoint rule, the modified Euler method, and Heun's predictor-corrector method
        ii. Take two successive forward Euler steps
            1. $\frac{y_{k+1}-y_k}{h}=f(t_k, y_k)$ and $\frac{y_{k+2}-y_{k+1}}{h}=f(t_{k+1}, y_{k+1})$
        iii. Average the initial and final state
            1. $y_{k+1}=\frac{1}{2}y_k+\frac{1}{2}y_{k+2}$
        iv. Same as above, but here one can see the averaging at work
        v. If the solution is well behaved for each Euler step, then since linear interpolation is well behaved, the result is well behaved
    c. 3rd order accurate TVD RK
        i. Take two successive forward Euler steps
            1. $\frac{y_{k+1}-y_k}{h}=f(t_k, y_k)$ and $\frac{y_{k+2}-y_{k+1}}{h}=f(t_{k+1}, y_{k+1})$
        ii. Average the initial and final state
            1. $y_{k+1/2}=\frac{3}{4}y_k+\frac{1}{4}y_{k+2}$
        iii. Take another Euler step
            1. $\frac{y_{k+3/2}-y_{k+1/2}}{h}=f(t_{k+1/2}, y_{k+1/2})$
        iv. Then average yet again
            1. $y_{k+1}=\frac{1}{3}y_k+\frac{2}{3}y_{k+3/2}$
7. **Multivalue methods** – efficiently use lower accuracy on higher derivatives
    a. Consider the Taylor expansion $x^{n+1}=x^n+\Delta t x_t^n+\frac{\Delta t^2}{2}x_{tt}^n+O(\Delta t^3)$ i.e. consider case where we have $x_t=v,\ x_{tt}=a\ or\ v_t=a$.
        i. If x^n has $O(\Delta t^r)$ errors than x_t^n can have $O(\Delta t^{r-1})$ errors without ruining the accuracy, similarly x_{tt}^n can have $O(\Delta t^{r-2})$ errors
        ii. e.g., 3rd order accurate $\vec{x}$ can be obtained with a 2nd order accurate $\vec{v}$ and 1st order accurate $\vec{a}$

iii. Solving $\begin{pmatrix}\vec{x}\\ \vec{v}\end{pmatrix}$ as a standard system is overkill on $\vec{v}$

b. Standard **constant acceleration** equations

i. $\vec{x}^{n+1} = \vec{x}^n + \Delta t\vec{v}^n + \frac{\Delta t^2}{2}\vec{a}^n$ *quadratic* position

ii. $\vec{v}^{n+1} = \vec{v}^n + \Delta t\vec{a}^n$ *linear* velocity

iii. $\vec{a}^{n+1} = \vec{a}^n$ *constant* acceleration (that is constant from time n to just before time n+1)

8. **Newmark Method –** most famous multivalue method in *computational mechanics*

a. Actually a lot of methods in disguise

b. $\vec{x}^{n+1} = \vec{x}^n + \Delta t\vec{v}^n + \frac{\Delta t^2}{2}\left[(1-2\beta)\vec{a}^n + 2\beta\vec{a}^{n+1}\right]$

c. $\vec{v}^{n+1} = \vec{v}^n + \Delta t\left[(1-\gamma)\vec{a}^n + \gamma\vec{a}^{n+1}\right]$

d. Choice of β, γ parameters makes a specific method.

i. $\beta = \gamma = 0$ - standard *constant acceleration* case (above)

ii. $\beta = 1/2$, $\gamma = 1$ - constant, implicit acceleration

1. $\vec{x}^{n+1} = \vec{x}^n + \Delta t\vec{v}^n + \frac{\Delta t^2}{2}\vec{a}^{n+1}$
2. $\vec{v}^{n+1} = \vec{v}^n + \Delta t\vec{a}^{n+1}$
3. Second equation is the same as 1st order accurate backward Euler
4. First equation is $\vec{x}^{n+1} = \vec{x}^n + \Delta t\left(\frac{\vec{v}^n + \vec{v}^{n+1}}{2}\right)$ which is the 2nd order accurate midpoint rule
5. Overall still 1st order accurate

iii. Exists a theorem states: 2nd order accuracy is obtained *if and only if* $\gamma = 1/2$

iv. $\beta = 1/4$, $\gamma = 1/2$ - *Trapezoidal rule* – 2nd order accurate

1. Again, constant acceleration, but this time using the midpoint acceleration
2. $\vec{x}^{n+1} = \vec{x}^n + \Delta t\vec{v}^n + \frac{\Delta t^2}{2}\left(\frac{\vec{a}^n + \vec{a}^{n+1}}{2}\right)$
3. $\vec{v}^{n+1} = \vec{v}^n + \Delta t\left(\frac{\vec{a}^n + \vec{a}^{n+1}}{2}\right)$
4. first equation is equivaluent to $\vec{x}^{n+1} = \vec{x}^n + \Delta t\left(\frac{\vec{v}^n + \vec{v}^{n+1}}{2}\right)$

v. $\beta = 0$, $\gamma = 1/2$ *central differencing*

1. $\vec{x}^{n+1} = \vec{x}^n + \Delta t\vec{v}^n + \frac{\Delta t^2}{2}\vec{a}^n$

2. $\vec{v}^{n+1} = \vec{v}^{n} + \Delta t\left(\frac{\vec{a}^{n} + \vec{a}^{n+1}}{2}\right)$
3. Called central differencing because both the acceleration and the velocity can be expressed as centered finite differences
4. $\vec{v}^{n+1} = \frac{\vec{x}^{n+2} - \vec{x}^{n}}{2\Delta t}$ can be derived by adding $\vec{x}^{n+1} = \vec{x}^{n} + \Delta t\vec{v}^{n} + \frac{\Delta t^2}{2}\vec{a}^{n}$ to $\vec{x}^{n+2} = \vec{x}^{n+1} + \Delta t\vec{v}^{n+1} + \frac{\Delta t^2}{2}\vec{a}^{n+1}$ and rearranging to obtain $\frac{\vec{x}^{n+2} - \vec{x}^{n}}{2\Delta t} = \frac{\vec{v}^{n+1}}{2} + \frac{1}{2}\left(\vec{v}^{n} + \Delta t\left(\frac{\vec{a}^{n} + \vec{a}^{n+1}}{2}\right)\right)$ and then realizing that the last term is identical to $\vec{v}^{n+1}$
5. $\vec{a}^{n+1} = \frac{\vec{x}^{n+2} - 2\vec{x}^{n+1} + \vec{x}^{n}}{\Delta t^2}$ can be derived by subtracting $\vec{x}^{n+1} = \vec{x}^{n} + \Delta t\vec{v}^{n} + \frac{\Delta t^2}{2}\vec{a}^{n}$ from $\vec{x}^{n+2} = \vec{x}^{n+1} + \Delta t\vec{v}^{n+1} + \frac{\Delta t^2}{2}\vec{a}^{n+1}$ and reorganizing to obtain $\frac{\vec{x}^{n+2} - 2\vec{x}^{n+1} + \vec{x}^{n}}{\Delta t^2} = \left(\frac{\vec{v}^{n+1} - \vec{v}^{n}}{\Delta t}\right) + \left(\frac{\vec{a}^{n+1} - \vec{a}^{n}}{2}\right)$ and then noting that $\frac{\vec{v}^{n+1} - \vec{v}^{n}}{\Delta t} = \frac{\vec{a}^{n} + \vec{a}^{n+1}}{2}$

9. **<u>Staggering the position and velocity</u>**
   a. Define the velocity at the half grid points so that $\vec{v}^{n+1/2} = \frac{\vec{x}^{n+1} - \vec{x}^{n}}{\Delta t}$ is second order accurae
      i. Note that $\vec{x}^{n}$ is still at the grid points
      ii. If we define $\vec{v}^{n+1} = \frac{\vec{v}^{n+1/2} + \vec{v}^{n+3/2}}{2} = \frac{\vec{x}^{n+2} - \vec{x}^{n}}{2\Delta t}$, then this is exactly central differencing for velocity
      iii. We can rewrite this as an update formula for position $\vec{x}^{n+1} = \vec{x}^{n} + \Delta t\vec{v}^{n+1/2}$
   b. The acceleration is also at the grid points, and $\frac{\vec{v}^{n+3/2} - \vec{v}^{n+1/2}}{\Delta t} = \vec{a}^{n+1}$ is second order accurate
      i. Note that this is also equivalent to $\frac{\left(\frac{\vec{x}^{n+2} - \vec{x}^{n+1}}{\Delta t}\right) - \left(\frac{\vec{x}^{n+1} - \vec{x}^{n}}{\Delta t}\right)}{\Delta t} = \vec{a}^{n+1}$ or $\frac{\vec{x}^{n+2} - 2\vec{x}^{n+1} + \vec{x}^{n}}{\Delta t^2} = \vec{a}^{n+1}$ which is second order accurate for a second derivative

ii. Again, as in the velocity case, this is just central differencing
iii. We can rewrite this as an update formula for acceleration $\vec{v}^{n+3/2} = \vec{v}^{n+1/2} + \Delta t \vec{a}^{n+1}$

c. The acceleration is evaluated at the grid points using
$$\vec{a}^{n+1} = \vec{a}(\vec{x}^{n+1}, \vec{v}^{n+1}) = \vec{a}\left(\vec{x}^{n+1}, \frac{\vec{v}^{n+1/2} + \vec{v}^{n+3/2}}{2}\right)$$

d. Summary, given $\vec{x}^n$ and $\vec{v}^n$:
   i. By definition $\vec{v}^n = \dfrac{\vec{v}^{n-1/2} + \vec{v}^{n+1/2}}{2}$
      1. So $\vec{v}^{n+1/2} = 2\vec{v}^n - \vec{v}^{n-1/2} = 2\vec{v}^n - (\vec{v}^{n+1/2} - \Delta t \vec{a}^n)$ using $\vec{v}^{n+1/2} = \vec{v}^{n-1/2} + \Delta t \vec{a}^n$
      2. This can be rearranged to $\vec{v}^{n+1/2} = \vec{v}^n + \dfrac{\Delta t}{2}\vec{a}^n$ to get the half step velocity $\vec{x}^{n+1} = \vec{x}^n + \Delta t \vec{v}^{n+1/2}$
      3. This is identical to $\vec{x}^{n+1} = \vec{x}^n + \Delta t \vec{v}^n + \dfrac{\Delta t^2}{2}\vec{a}^n$
   ii. Then $\vec{v}^{n+3/2} = \vec{v}^{n+1/2} + \Delta t \vec{a}\left(\vec{x}^{n+1}, \dfrac{\vec{v}^{n+1/2} + \vec{v}^{n+3/2}}{2}\right)$
      1. Using $\vec{v}^{n+1} = \dfrac{\vec{v}^{n+1/2} + \vec{v}^{n+3/2}}{2}$ leads to
         $2\vec{v}^{n+1} - \vec{v}^{n+1/2} = \vec{v}^{n+1/2} + \Delta t \vec{a}\left(\vec{x}^{n+1}, \vec{v}^{n+1}\right)$ or
         $\vec{v}^{n+1} = \vec{v}^{n+1/2} + \dfrac{\Delta t}{2}\vec{a}(\vec{x}^{n+1}, \vec{v}^{n+1})$

e. This last equation is implicit in the velocity
   i. Fully explicit if acceleration doesn't depend on velocity
   ii. Otherwise iterate $\vec{u}^{k+1} = \vec{v}^{n+1/2} + \dfrac{\Delta t}{2}\vec{a}(\vec{x}^{n+1}, \vec{u}^k)$
      1. Starting with $\vec{u}^0 = \vec{v}^{n+1/2}$
      2. $\vec{u}^1 = \vec{v}^{n+1/2} + \dfrac{\Delta t}{2}\vec{a}(\vec{x}^{n+1}, \vec{v}^{n+1/2})$ which is an explicit time step
   iii. Often the dependence of acceleration on velocity is symmetric (e.g. damping forces), so one can use a fast Ax=b solver, e.g. PCG
   iv. The overall velocity update looks like
   $$\vec{v}^{n+1} = \vec{v}^{n+1/2} + \frac{\Delta t}{2}\vec{a}(\vec{x}^{n+1}, \vec{v}^{n+1}) = \vec{v}^n + \frac{\Delta t}{2}\vec{a}(\vec{x}^n, \vec{v}^n) + \frac{\Delta t}{2}\vec{a}(\vec{x}^{n+1}, \vec{v}^{n+1})$$
   which is the trapezoidal rule.
   v. There is no stability restriction on the time step from the accelerations dependence on velocity
      1. The only time step stability restriction comes from the accelerations dependence on position
      2. Can take a slightly bigger time step

3. Of course, bigger time steps are bad for the trapezoidal rule, so one could switch to backward Euler for the velocity
   a. $\vec{v}^{n+1} = \vec{v}^{n} + \Delta t \vec{a}(\vec{x}^{n+1}, \vec{v}^{n+1})$
   b. Still use $\vec{x}^{n+1} = \vec{x}^{n} + \Delta t \vec{v}^{n} + \frac{\Delta t^2}{2} \vec{a}^{n}$ for position
   c. Only 1st order accurate overall