DC2020/02/23

Monday, March 23, 2020 1:13 PM

$$\mathbf{M\dot{U}} + \mathbf{KU} = \mathbf{R} \quad \text{Parabolic} \quad \text{Example: Heat equation } \mathbf{M} = \text{ capacity, } \mathbf{K} = \text{ conductivity matrices} \tag{226b}$$

Note that any second order (or n^{th} order) ODE can be written in the form (226b) and the form itself does not directly imply if the underlying PDE is hyperbolic or parabolic. For example, we can express (226a) as,

$$\dot{\mathbf{U}} - \mathbf{V} = 0 \tag{227a}$$

$$\mathbf{M\dot{V}} + (\mathbf{KU} + \mathbf{CV}) = \mathbf{R} \tag{227b}$$

where $\mathbf{V}$ represents the temporal derivative of $\mathbf{U}$, *i.e.*, velocity when $\mathbf{U}$ is displacement. We can express this in the form (226b):

$$\tilde{\mathbf{M}}\dot{\tilde{\mathbf{U}}} + \tilde{\mathbf{K}}\tilde{\mathbf{U}} = \tilde{\mathbf{R}}, \text{where} \tag{228a}$$

$$\tilde{\mathbf{U}} = \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix}, \quad \tilde{\mathbf{M}} = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{M} \end{bmatrix} \quad \tilde{\mathbf{K}} = \begin{bmatrix} 0 & -1 \\ \mathbf{K} & \mathbf{C} \end{bmatrix} \quad \tilde{\mathbf{R}} = \begin{bmatrix} 0 \\ \mathbf{R} \end{bmatrix} \tag{228b}$$

thus

$$\ddot{x} + 2\xi\omega\dot{x} + \omega^2 x = f(t) \quad \text{corresponding to} \quad \mathbf{M\ddot{U}} + \mathbf{C\dot{U}} + \mathbf{KU} = \mathbf{R} \tag{229a}$$

$$\dot{x} + \lambda x = f(t) \quad \text{corresponding to} \quad \mathbf{M\dot{U}} + \mathbf{KU} = \mathbf{R} \tag{229b}$$

- Consistency
  - Consistency refers to the concept that the update from step t_n (and previous ones in multi-step methods) to t_{n+1} is "consistent" with an underlying update of the exact solution from t_n to t_{n+1} and the local truncation error, *i.e.*, error of computational versus exact values for t_n, is $\mathcal{O}(\Delta t^p)$, $p > 1$.

- Consistency
  - Consistency refers to the concept that the update from step t_n (and previous ones in multi-step methods) to t_{n+1} is "consistent" with an underlying update of the exact solution from t_n to t_{n+1} and the local truncation error, *i.e.*, error of computational versus exact values for t_n, is $\mathcal{O}(\Delta t^p)$, $p > 1$.

if only use t_n & t_{n+1} values → single step

Unknown s/n t_{n+1} ○

- Consistency
  - Consistency refers to the concept that the update from step t_n (and previous ones in multi-step methods) to t_{n+1} is "consistent" with an underlying update of the exact solution from t_n to t_{n+1} and the local truncation error, *i.e.*, error of computational versus exact values for t_n, is $\mathcal{O}(\Delta t^p)$, $p > 1$.

t_n ○

t_{n-1} ○

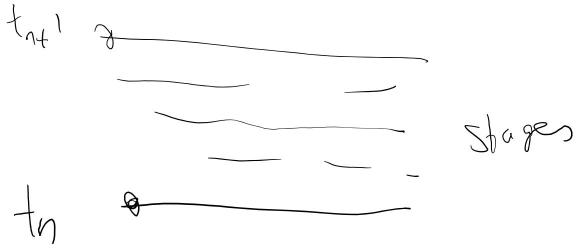
- A variety of time marching schemes can be directly applied to MDOF (226) or to their individual SDOF (188).
- In the next section, §5, we will use the SDOF form (188) for stability analysis of MDOF (226).

$$\dot{u} + a u = 0 \qquad \frac{u_{n+1} - u_{n-1}}{2\Delta t} + a u_n = 0$$

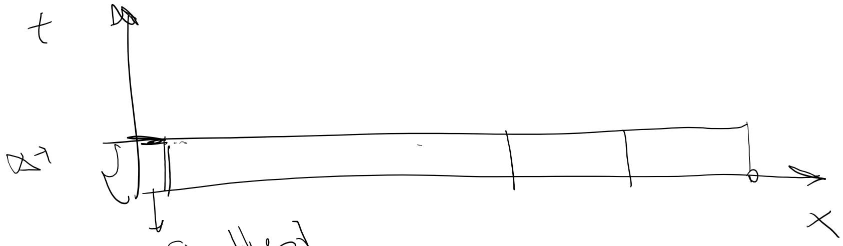
3. Single-step vs. Multi-step:
   - Single-step: only values from t_n are needed to obtain the solution for t_{n+1}.
   - Multi-step: values of t_n, t_{n-1}, and possibly more, are required to obtain the solution for t_{n+1}.

Multistep formula

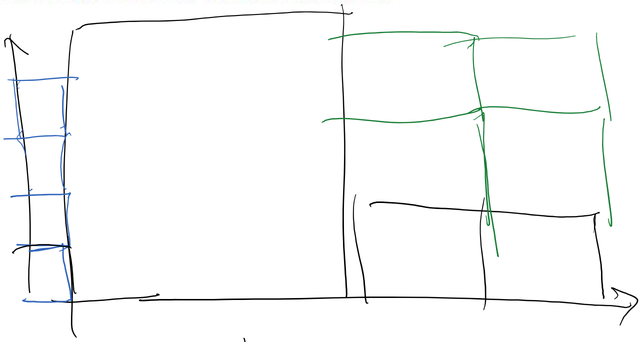
4. Single-stage vs. Multi-stage
   - Single-stage: Single We directly compute t_{n+1} from t_n (and possibly prior values) in one stage.
   - Multi-stage: To obtain t_{n+1} from previous solutions several intermediate values are computed from t_n to t_{n+1}. Runge-Kutta (RK) methods are Multi-stage methods.



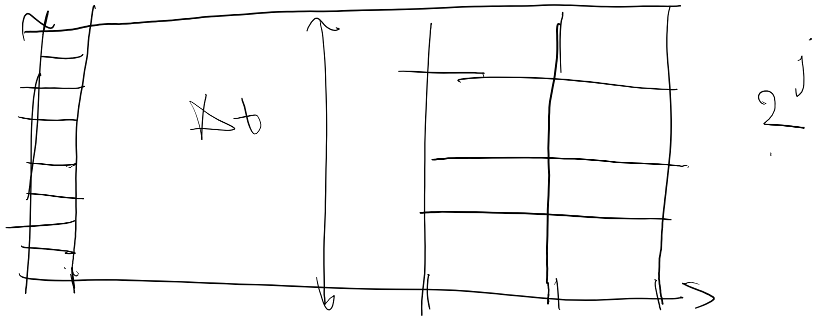
5. Global versus local time step size: In explicit methods (below) smallest elements pose serious challenges in the time step of the entire domain. It is desirable to use smaller time step for smaller elements than larger ones. The same concept, but only from accuracy perspective and not for stability reasons, becomes relevant in implicit methods. The flexibility of a time marching scheme in this respect increases in the following order:



smallest elements → smallest time advance



local time stepping (LTS) each element ideally takes it's max time advance



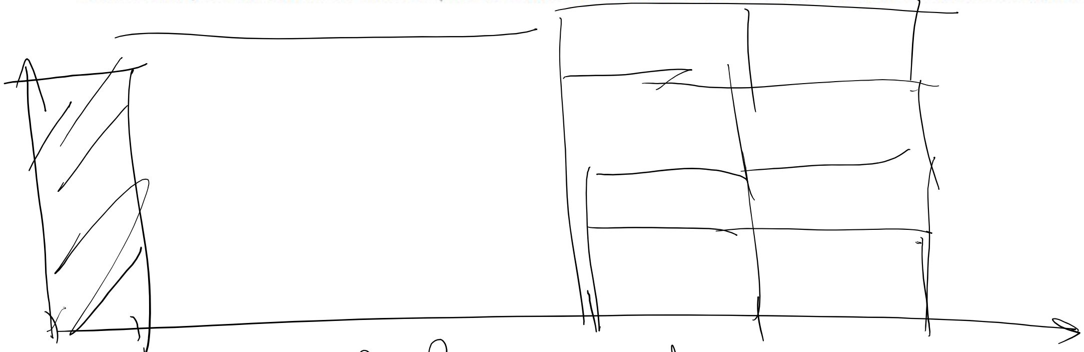
Subcycling   Δt of elements are $\frac{\Delta t}{2^j}$

$\Delta t / n^j$ $n = 2, 3, \ldots$

Large local errors (-> refinement) or high geometry complexity can result in highly refined region in a mesh

- Global time step: All elements, small or large, share the same time step $\Delta t = t_{n+1} - t_n$.
- Subcycling: Smaller elements take smaller time step, often in factors of 2^s $(\Delta t/2^s)$ time smaller than the global time step $\Delta t = t_{n+1} - t_n$. Although elements have different time steps, at the end of the time step all have the same time value t_{n+1}. That, is from global time step to time step the method is synchronous.
- Local Time Stepping (LTS) (asynchronous subcycling): Although LTS is also used for subcycling approaches, in general it is used for asynchronous time stepping schemes that each element takes a local time step (based on its stability limits in explicit methods) and element final times do not need to be synchronized.

LTS/ subcycling approaches, along with IMEX schemes, will later be discussed as approaches to solve highly graded FE meshes and/or still problems where some operators of the PDE requires very small time steps with explicit methods.



explicit solve for large elements

Implicit solve for fine elements

6. Stability:

- Unconditional unstable: The time marching scheme is unstable for any Δt. Clearly, such time integrators will not be used at all!
- Conditional stable: Δt must satisfy $\Delta t_{\max} \le \Delta t_{\max}$ (or more generally satisfy special conditions) for the solution to be stable.
- Unconditional stable: For an underlying physical PDE, the numerical scheme is stable for any Δt.

7. Order of convergence:

- For a method that is convergent we need to know how fast the numerical solution converges to the exact solution of the underlying problem. For example, the error between the two at a given time can temporally converge as Δt^q. We also have a spatial order of convergence h^p based on the order of of elements used and the type of error considered. The concept of order of convergence is closely related to "consistency", "local truncation error", and "convergence" discussed below.
- In general, achieving high temporal convergence rates is much more difficult than spatial ones, as FE method can easily accommodate any spatial order of accuracy, but often FD type temporal update makes it challenging to achieve high temporal orders of accuracy.

Stability

8. Explicit vs. Implicit

- Explicit: Refers to being able to solve solution for t_{n+1} "explicitly". Some aspects are:
  - No global matrix equation or nonlinear solution needs to be solved. That is,
  - Even if the underlying PDE is nonlinear, update equations will typically be linear.
  - Are often only conditionally stable.

- In time stepping methods, explicit methods are often expressed for t_n, have only $\mathbf{M}$ (and $\mathbf{C}$) on the LHS of the update equation. If $\mathbf{C} = 0$ and mass matrix $\mathbf{M}$ is diagonal (or capacity diagonal in heat equation) is used the update equation is trivial and no global equations should be solved. In fact, if the solution cannot be reduced to a local and small matrix update equation, many still do not call the scheme explicit!. In these course notes, we label a larger group of problems explicit, basically by referring to schemes that write the equations at t_n rather than at a later time.
- To enable local and no global matrix solution strategy and other reasons discussed later explicit methods are often matched with diagonal "mass" matrices.

- Implicit: Refers to being to solve solution for t_{n+1} "implicitly". Some aspects are:
  - Involve global matrix equations. Also,
  - If the underlying PDE is nonlinear, update equations will be nonlinear.
  - Are often unconditionally stable. The rare ones that are conditionally stable are not used. There are different levels and types of stability which we do not cover all in this course. But in general, the implicit methods in practice are stable for much wider ranges and stages of time step and PDE coefficient.
  - In time stepping methods, explicit methods are often expressed for t_{n+1}, have all $\mathbf{M}$, $\mathbf{K}$ (and $\mathbf{C}$) on the LHS of the update equation. Even for linear PDEs they require the solution of a full matrix equation for the solution of $\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R}$.

  - For considerations that will be discussed later implicit methods are often matched with consistent "mass" matrices.
- Implicit-Explicit (IMEX): These are the schemes that use both explicit and implicit integration schemes, in one of both of the following modes:

shared solver

  - Domain IMEX: Some parts of the domain are solved with explicit solver some parts with implicit solver. For example regions with small elements use implicit solver and large elements explicit solver. Another example, is using implicit ~~solver for solid~~ part and explicit method for a fluid part.
  - Operator IMEX: Example is using implicit integrators for more stringent operators such as parabolic ones and explicit integrators otherwise, *e.g.*, hyperbolic operator. Same concept can be applied to stiff PDEs where the stiff operators are integrated implicitly.

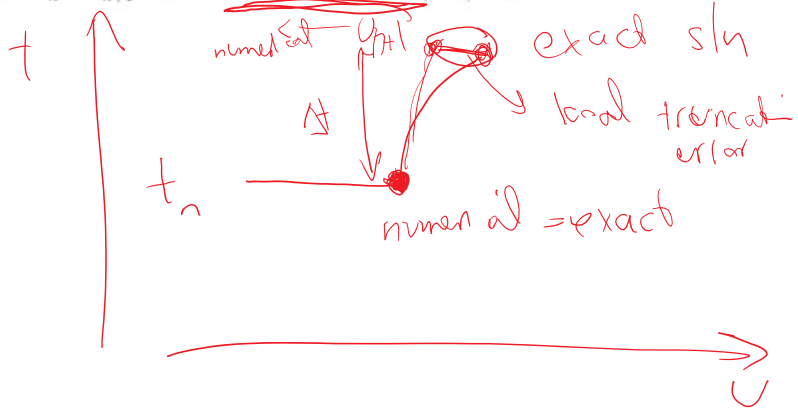
$$\dot{u} + a u_{,x} - \nu u_{,xx} = 0$$

(a) $\dot{u} + a u_{,x}$ hyperbolic $\Delta t \propto \frac{\Delta x}{a}$

(b) $\dot{u} - \nu u_{,xx} = 0$ parabolic $\boxed{\Delta t \propto \frac{\Delta x^2}{2\nu}}$ very stiff

Operator splitting is used. (a) is integrated explicitly (b) implicitly → Operator splitting IMEX method.

- Consistency
  - Consistency refers to the concept that the update from step t_n (and previous ones in multi-step methods) to t_{n+1} is "consistent" with an underlying update of the exact solution from t_n to t_{n+1} and the local truncation error, *i.e.*, error of computational versus exact values for t_n, is $\mathcal{O}(\Delta t^p)$, $p > 1$.
- Convergence and rate of convergence



  - Convergence: The numerical method convergence to the exact solution as $\Delta t \to 0$, *i.e.*, $\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R}$ is exactly integrated. This does not mean that we converge to the underlying PDE solution. For that element size h must also approach zero.
  - Convergence rate: The rate in which the numerical solution convergences to the exact solution in terms of Δt^p.

In this section, we review various time marching schemes and in the following section we present their stability and convergence analysis.

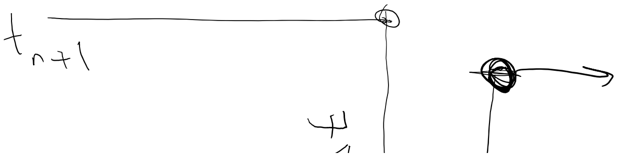
4.2 A One-step single-field time stepping method: Generalized trapezoidal rule

- We consider the solution of a first order ODE of the form:

$$\mathbf{M}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{F} \quad \text{Temporal ODE (after FEM spatial discretization)} \quad (230a)$$
$$\mathbf{d}(t = 0) = \mathbf{d}_0 \quad \text{Initial Condition (IC)} \quad (230b)$$

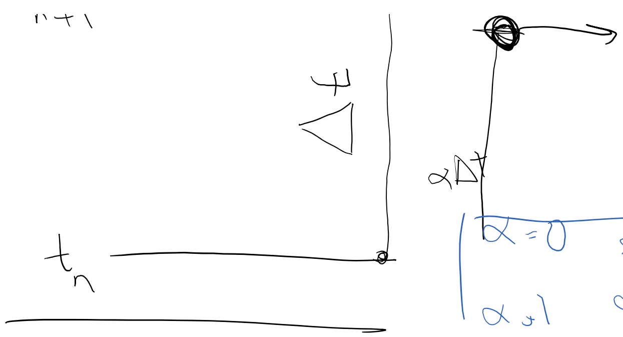
this can have been derived from the discretization of a parabolic equation (226b) or two-field representation of a second order hyperbolic equation (228).

- The update equation for the time $t = t_n + \alpha\Delta t$ is written as,



$\dot{d} = \frac{d_{n+1} - d_n}{\Delta t}$

$\dot{d}$, d



$$d = \frac{d_{n+1} - c_n}{\Delta t}$$

$$d = (1-\alpha) d_n + \alpha d_{n+1} \quad \begin{pmatrix} \alpha = 1 \rightarrow d_{n+1} \\ \alpha = 0 \rightarrow d_n \end{pmatrix}$$

$\alpha = 0$ $\quad \frac{d_{n+1} - d_n}{\Delta t} + \lambda d_n = 0$ Forward Euler $\dot{d} + \lambda d = 0$

$\alpha = 1$ $\quad \frac{d_{n+1} - d_n}{\Delta t} + \lambda d_{n+1} = 0$ Backward Euler

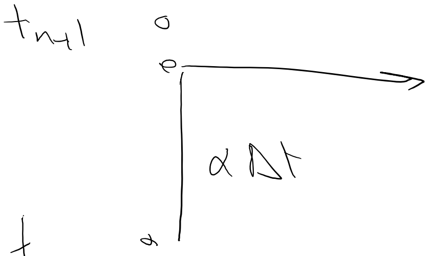
Solve a SDOF $\quad M\dot{d} + Kd = F \longrightarrow$

Modal analysis $\quad M\phi = \lambda K \phi$

SDOF's are $\quad \dot{d}_i + \lambda_i d = f_i$

1 SDOF $\quad \boxed{\dot{d} + \lambda d = f(t)} \ \star$

use Generalized trapezoidal rule (α Method)



$$\left.\begin{aligned} \dot{d} &= \frac{d_{n+1} - d_n}{\Delta t} \\ d &= (1-\alpha) d_n + \alpha d_{n+1} \end{aligned}\right] \rightarrow \text{plug in } \star$$

$$f = (1-\alpha) f_n + \alpha f_{n+1}$$

Δt X $\quad \frac{d_{n+1} - d_n}{\Delta t} + \lambda \left((1-\alpha) d_n + \alpha d_{n+1} \right) = (1-\alpha) f_n + \alpha f_{n+1}$

$$\underline{d_{n+1}} \ \underline{\underline{- d_n}} + \Delta t \lambda \underline{(1-\alpha) d_n} + \underline{\Delta t \lambda \alpha d_{n+1}} = \Delta t (1-\alpha) f_n + \Delta t \alpha f_{n+1}$$

$$(1 + \alpha \Delta t \lambda) d_{n+1} = (1 - (1-\alpha) \Delta t \lambda) d_n + \ \downarrow \ //$$

$$\boxed{\begin{aligned} d_{n+1} &= \left[\frac{1 - (1-\alpha)\Delta t \lambda}{1 + \alpha \Delta t \lambda} \right] d_n + \Delta t \frac{(1-\alpha) f_n + \alpha f_{n+1}}{1 + \alpha \Delta t \lambda} \\ d_{n+1} &= A d_n + L_n \end{aligned}}$$

$$d_{n+1} = A\, d_n + L_n$$

$$\boxed{A = \frac{1-(1-\alpha)\Delta t\,\lambda}{1+\alpha\,\Delta t\,\lambda}}$$

if $f=0$

$$d_{n+1} = A d_n \qquad A = \frac{1-(1-\alpha)\Delta t\,\lambda}{1+\alpha\,\Delta t\,\lambda}$$

$d_1 = A d_0$

$d_2 = A d_1$

$\vdots$

$$d_{n+1} = A^{n+1} d_0$$

if $|A|>1$ $d_n \to \infty$ unstable

$$|A| = \left|\frac{1-(1-\alpha)\Delta t\,\lambda}{1+\alpha\,\Delta t\,\lambda}\right| \le 1$$

$$-1 \le \frac{1-(1-\alpha)\Delta t\,\lambda}{1+\underbrace{\alpha}_{\ge 0}\underbrace{\Delta t}_{>0}\underbrace{\lambda}_{\ge 0}} \le 1$$

Exact sln

$\dot d + \lambda d = 0$

$d(t) = d_0 e^{-\lambda t}$

$\lambda < 0$ physically OK is not well-posed

$$-1 - \alpha\Delta t\lambda \le 1-(1-\alpha)\Delta t\,\lambda \le 1 + \alpha\Delta t\lambda$$

$$\boxed{(1-2\alpha)\Delta t\,\lambda \le 2} \quad (\square)$$

Stability condition ✓

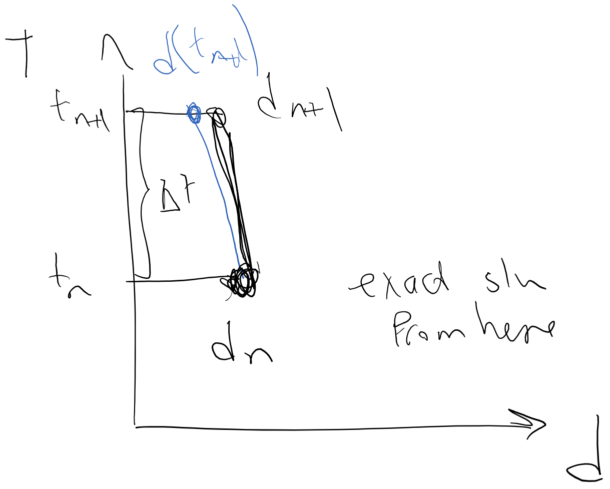
$\alpha \ge \frac{1}{2}$ $(\square)$ is satisfied for all Δt → unconditionally stable

$\alpha < \frac{1}{2}$ $\boxed{\Delta t \le \frac{2}{(1-2\alpha)\lambda}}$

$\alpha > 0$ implicit (because it involves d & $\dot{d}$ of t_{n+1})

Local truncation error

$$\boxed{d_{n+1} = d_n A}$$



$\dot{d} + \lambda d = 0$ & $d(t_n) = d_n$

$d(t) = d_n e^{-\lambda(t-t_n)}$ $\quad t > t_n$

$$\boxed{d(t_{n+1}) = d_n e^{-\lambda \Delta t}}$$

truncation error $= d_{n+1} - d(t_{n+1}) = d_n\left(A - e^{-\lambda \Delta t}\right)$

$= O(\Delta t^p)$ $\quad p > 1$

then order of accuracy of time marching is $\underline{p-1}$

α	Method	Explicit/Implicit	Temporal Order
0	Forward differences; forward Euler	Explicit	1
$\frac{1}{2}$	Trapezoidal rule; midpoint rule; Crank-Nicolson	Implicit	2
1	Backward differences; backward Euler	Implicit	1

- It is clear for (306) not to blow up we need to have,

$$|A| \leq 1 \qquad \Rightarrow \qquad -1 \leq \frac{1-(1-\alpha)\Delta t\lambda}{1+\alpha\Delta t\lambda} \leq 1 \tag{238}$$

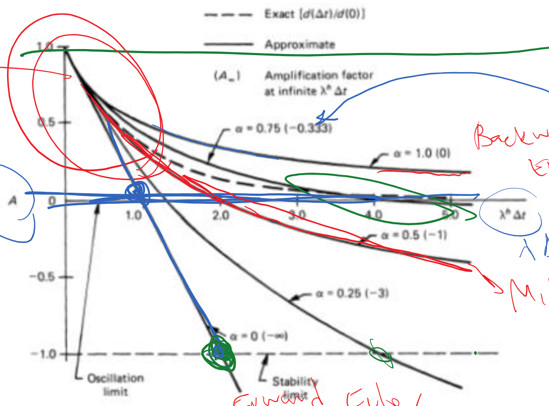
which results in the following conditions:

$$\begin{cases} \alpha < \frac{1}{2} & \text{Conditionally stable} \quad \boxed{\Delta t\lambda < \frac{2}{1-2\alpha}} \\ \alpha \geq \frac{1}{2} & \text{Unconditionally stable} \end{cases} \qquad \Delta t < \frac{2}{(1-\alpha)\lambda} \tag{239}$$

- Also figure below demonstrate how for unstable methods amplification factor A becomes less than one, resulting in oscillating and gowning solution.

Midpoint is most accurate as $\Delta t \to 0$

positivity limit



Amplification factor for typical one-step methods.

$$A = \frac{1-(1-\alpha)\Delta t\lambda}{1+\alpha\Delta t\lambda}$$

$$\lim_{\Delta t \to \infty} A = \frac{-(1-\alpha)}{\alpha}$$

for $\Delta t = 1.2$

Generalized trapezoidal rule (α Method)

$$\begin{tikzcd}
\boxed{\begin{array}{l} M\dot{d} + Kd = F \\ d(0) = d_0 \end{array}} \arrow[r, "\text{temporal discretization}"] \arrow[d, "\text{modal decomposition}"'] & \boxed{\begin{array}{l} (M + \alpha\Delta t K)d_{n+1} \\ = (M - (1-\alpha)\Delta t K)d_n \\ + \Delta t F_{n+\alpha} \\ d_0 \text{ given} \end{array}} \arrow[d, "\text{modal decomposition}"] \\
\boxed{\begin{array}{l} \dot{d} + \lambda^h d = F \\ d(0) = d_0 \end{array}} \arrow[r, "\text{temporal discretization}"] & \boxed{\begin{array}{l} (1 + \alpha\Delta t\lambda^h)d_{n+1} \\ = (1 - (1-\alpha)\Delta t\lambda^h)d_n \\ + \Delta t F_{n+\alpha} \\ d_0 \text{ given} \end{array}}
\end{tikzcd}$$

4.3 Linear multi-step (LMS) methods

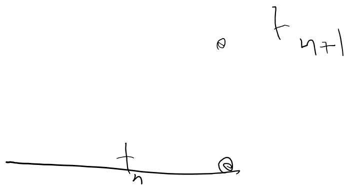
- Consider the first order ODE,

$$\boxed{\dot{\mathbf{y}} = f(\mathbf{y}, t)} \qquad \text{General nonlinear first order ODE} \tag{240a}$$

$$= \mathbf{G}\mathbf{y} + \mathbf{H}(t) \qquad \xleftarrow{\text{if linear}} \text{Linear first order ODE} \tag{240b}$$

Forward Euler

$$\longrightarrow \dot{y}_n = f(y_n, t_n)$$



$$\frac{y_{n+1} - y_n}{\Delta t} = f(y_n, t_n)$$

$$\boxed{\underset{\alpha_0}{(1)} y_{n+1} + \Delta t \cdot \underset{\beta_0}{0} \cdot f(y_{n+1}, t_{n+1}) + \underset{\alpha_1}{(-1)} y_n + \Delta t \underset{\beta_1}{(-1)} f(y_n, t_n) = 0}$$

Forward Elue

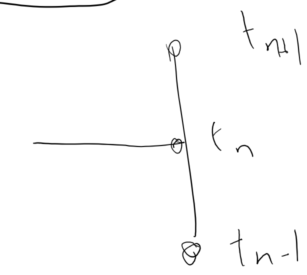
Backward Euler $\quad \dfrac{y_{n+1} - y_n}{\Delta t} = \dot{y}_{n+1} = f(y_{n+1}, t_{n+1})$

$$\boxed{\underset{\alpha_0}{(1)} y_{n+1} + \Delta t \underset{\beta_0}{(-1)} f(y_{n+1}, t_{n+1}) + \underset{\alpha_1}{(-1)} y_n + \Delta t \underset{\beta_1}{(0)} f(y_n, t_n) = 0}$$

Backward Euler

Central Difference

$$\frac{y_{n+1} - y_{n-1}}{2\Delta t} = f(y_n, t_n)$$

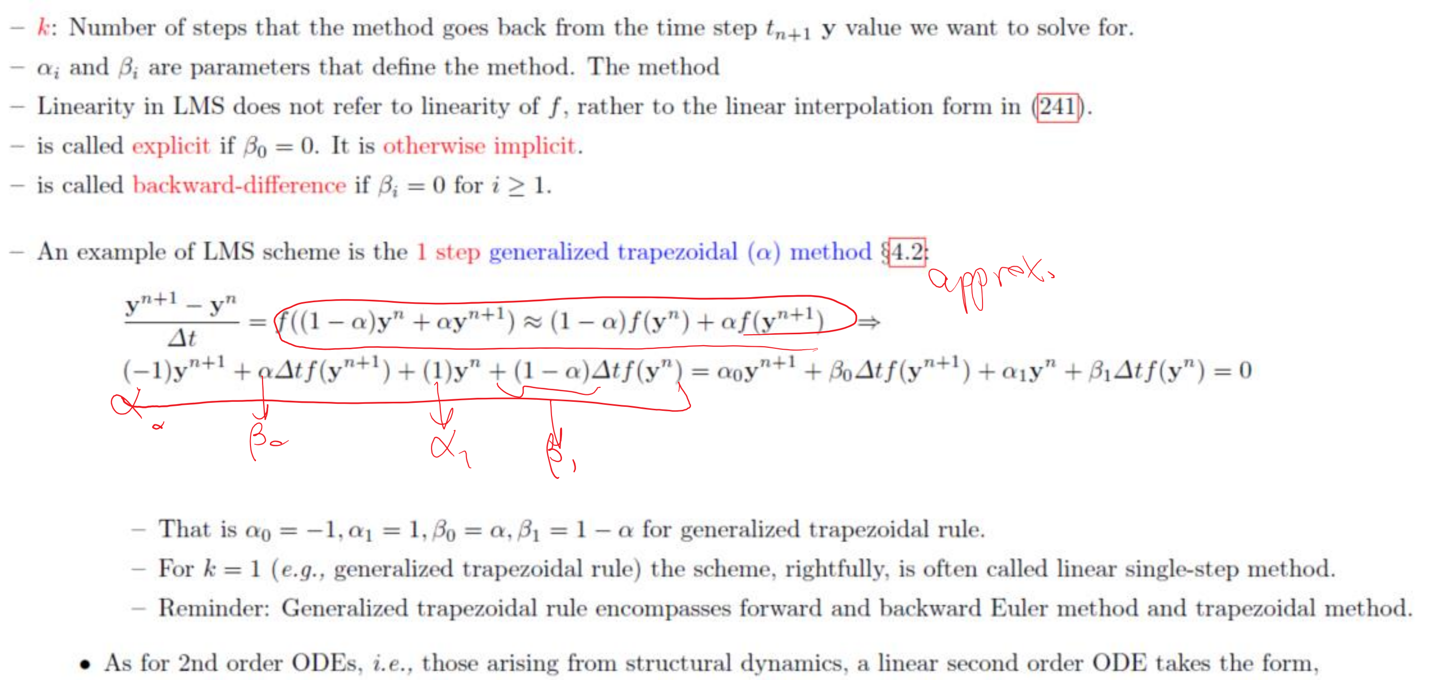


$$\boxed{\begin{aligned} &\underset{\alpha_0}{(1)} y_{n+1} + \Delta t \underset{\beta_0}{(0)} f(y_{n+1}, t_{n+1}) \\ &+ \underset{\alpha_1}{0} y_n + \Delta t \underset{\beta_1}{(-2)} f(y_n, t_n) \\ &\underset{\alpha_2}{(-1)} y_{n-1} + \Delta t \underset{\beta_2}{(0)} f(y_{n-1}, t_{n-1}) = 0 \end{aligned}}$$

- A k-step linear multi-step method is defined as,

$$\sum_{i=0}^{k} \{\underline{\alpha_i \mathbf{y}^{n+1-i}} + \Delta t \underline{\beta_i} f(\mathbf{y}^{n+1-i}, t_{n+1-i})\} = 0 \tag{241}$$

$$\beta_0 = 0 \longrightarrow \text{explicit}$$



- k: Number of steps that the method goes back from the time step t_{n+1} $\mathbf{y}$ value we want to solve for.
- α_i and β_i are parameters that define the method. The method
- Linearity in LMS does not refer to linearity of f, rather to the linear interpolation form in (241).
- is called explicit if $\beta_0 = 0$. It is otherwise implicit.
- is called backward-difference if $\beta_i = 0$ for $i \geq 1$.

- An example of LMS scheme is the 1 step generalized trapezoidal (α) method §4.2:

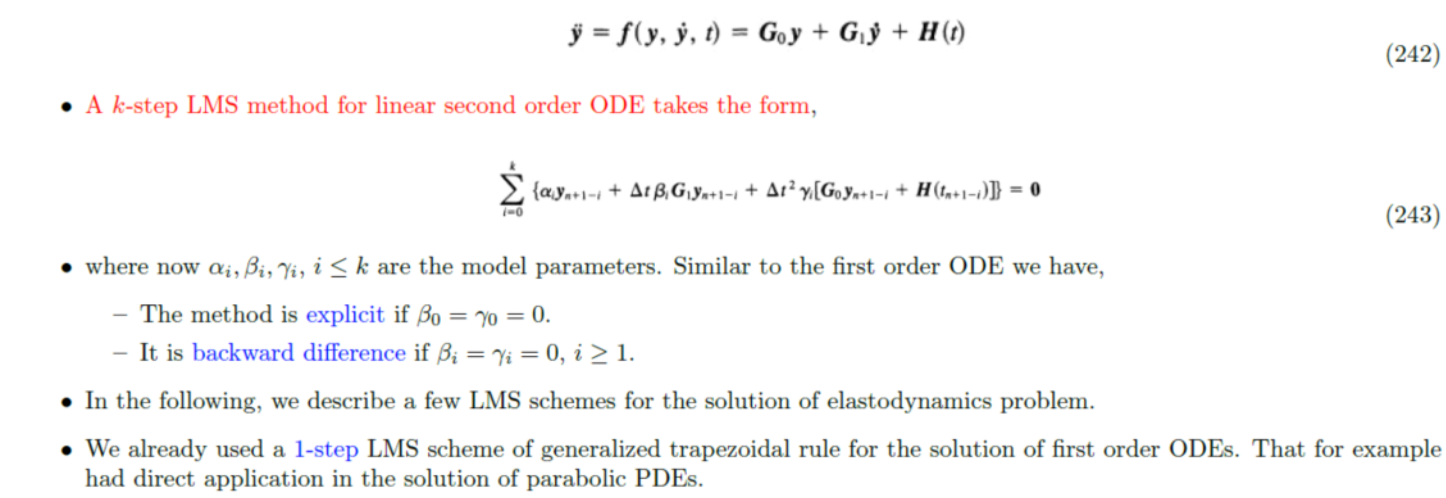
approx.

$$\frac{\mathbf{y}^{n+1}-\mathbf{y}^n}{\Delta t} = f((1-\alpha)\mathbf{y}^n + \alpha\mathbf{y}^{n+1}) \approx (1-\alpha)f(\mathbf{y}^n) + \alpha f(\mathbf{y}^{n+1}) \Rightarrow$$
$$\underset{\alpha_0}{(-1)}\mathbf{y}^{n+1} + \underset{\beta_0}{\alpha}\Delta t f(\mathbf{y}^{n+1}) + \underset{\alpha_1}{(1)}\mathbf{y}^n + \underset{\beta_1}{(1-\alpha)}\Delta t f(\mathbf{y}^n) = \alpha_0\mathbf{y}^{n+1} + \beta_0\Delta t f(\mathbf{y}^{n+1}) + \alpha_1\mathbf{y}^n + \beta_1\Delta t f(\mathbf{y}^n) = 0$$

  - That is $\alpha_0 = -1, \alpha_1 = 1, \beta_0 = \alpha, \beta_1 = 1-\alpha$ for generalized trapezoidal rule.
  - For $k = 1$ (*e.g.*, generalized trapezoidal rule) the scheme, rightfully, is often called linear single-step method.
  - Reminder: Generalized trapezoidal rule encompasses forward and backward Euler method and trapezoidal method.

- As for 2nd order ODEs, *i.e.*, those arising from structural dynamics, a linear second order ODE takes the form,

For linear 2nd order ODEs we have the following form of LMS



$$\ddot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y}, \dot{\boldsymbol{y}}, t) = \boldsymbol{G}_0\boldsymbol{y} + \boldsymbol{G}_1\dot{\boldsymbol{y}} + \boldsymbol{H}(t) \tag{242}$$

- A k-step LMS method for linear second order ODE takes the form,

$$\sum_{i=0}^{k} \{\alpha_i \boldsymbol{y}_{n+1-i} + \Delta t\,\beta_i \boldsymbol{G}_1 \boldsymbol{y}_{n+1-i} + \Delta t^2 \gamma_i [\boldsymbol{G}_0 \boldsymbol{y}_{n+1-i} + \boldsymbol{H}(t_{n+1-i})]\} = \boldsymbol{0} \tag{243}$$

- where now $\alpha_i, \beta_i, \gamma_i$, $i \leq k$ are the model parameters. Similar to the first order ODE we have,
  - The method is explicit if $\beta_0 = \gamma_0 = 0$.
  - It is backward difference if $\beta_i = \gamma_i = 0$, $i \geq 1$.
- In the following, we describe a few LMS schemes for the solution of elastodynamics problem.
- We already used a 1-step LMS scheme of generalized trapezoidal rule for the solution of first order ODEs. That for example had direct application in the solution of parabolic PDEs.