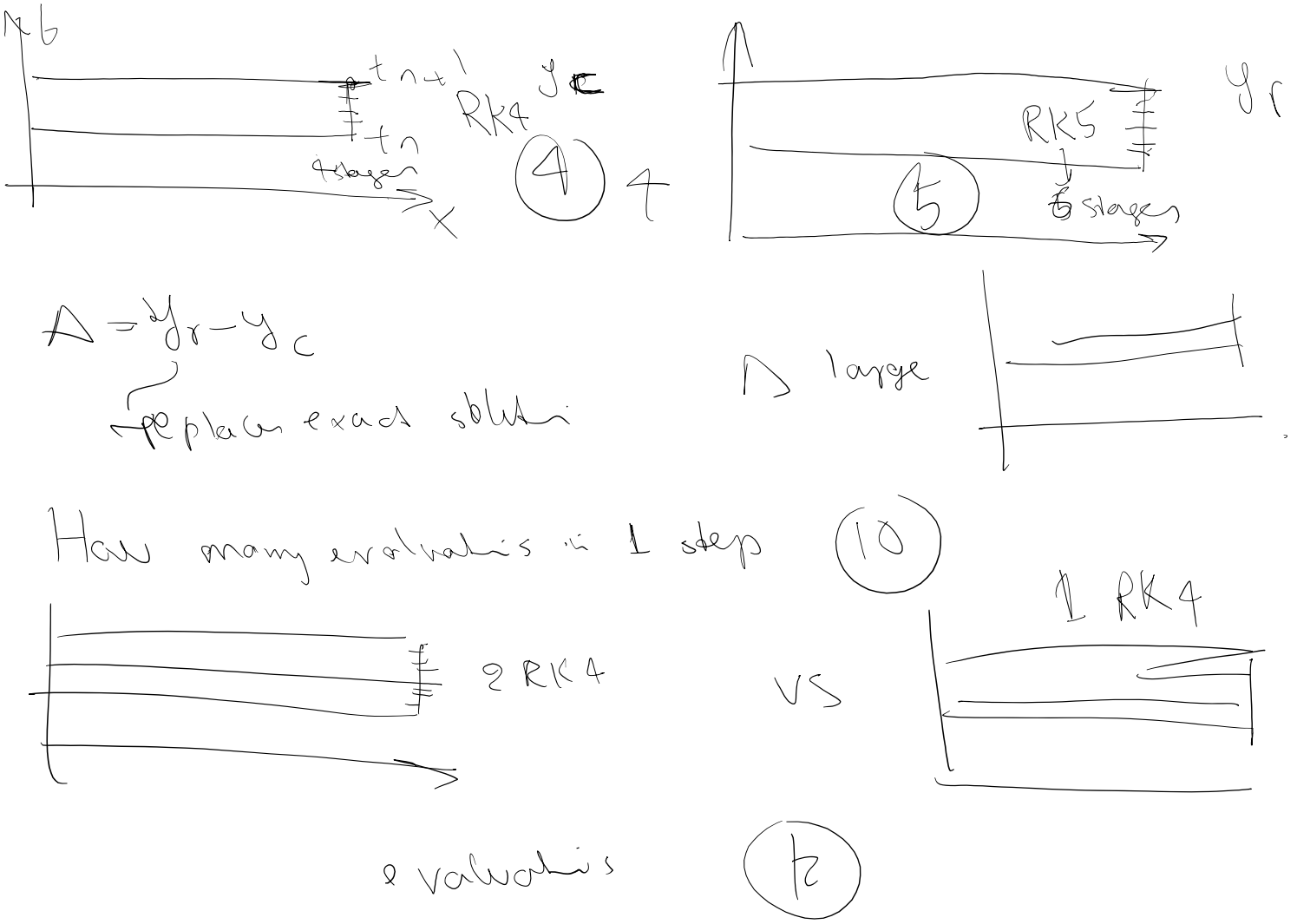
DC2020/04/01

Wednesday, April 1, 2020 1:10 PM

The same idea can be used with two successive RK methods



Aside from step halving as a strategy to adjust step size, an alternative approach for obtaining an error estimate involves computing two RK predictions of different order. The results can then be subtracted to obtain an estimate of the local truncation error. One shortcoming of this approach is that it greatly increases the computational overhead. For example, a fourth- and fifth-order prediction amount to a total of 10 function evaluations per step. The *Runge-Kutta Fehlberg* or *embedded RK* method cleverly circumvents this problem by using a fifth-order RK method that employs the function evaluations from the accompanying fourth-order RK method. Thus, the approach yields the error estimate on the basis of only six function evaluations!

For the present case, we use the following fourth-order estimate

$$y_{i+1} = y_i + \left(\frac{37}{378}k_1 + \frac{250}{621}k_3 + \frac{125}{594}k_4 + \frac{512}{1771}k_6\right)h \tag{281}$$

along with the fifth-order formula:

4 used for 4th order

6 Stage evaluations $k_1, \ldots, k_6$

$$y_{i+1} = y_i + \left(\frac{2825}{27,648}k_1 + \frac{18,575}{48,384}k_3 + \frac{13,525}{55,296}k_4 + \frac{277}{14,336}k_5 + \frac{1}{4}k_6\right)h \qquad \text{where} \tag{282}$$

$$\begin{aligned}
k_1 &= f(x_i, y_i)\\
k_2 &= f\left(x_i + \frac{1}{5}h, y_i + \frac{1}{5}k_1 h\right)\\
k_3 &= f\left(x_i + \frac{3}{10}h, y_i + \frac{3}{40}k_1 h + \frac{9}{40}k_2 h\right)\\
k_4 &= f\left(x_i + \frac{3}{5}h, y_i + \frac{3}{10}k_1 h - \frac{9}{10}k_2 h + \frac{6}{5}k_3 h\right)\\
k_5 &= f\left(x_i + h, y_i - \frac{11}{54}k_1 h + \frac{5}{2}k_2 h - \frac{70}{27}k_3 h + \frac{35}{27}k_4 h\right)\\
k_6 &= f\left(x_i + \frac{7}{8}h, y_i + \frac{1631}{55,296}k_1 h + \frac{175}{512}k_2 h + \frac{575}{13,824}k_3 h + \frac{44,275}{110,592}k_4 h + \frac{253}{4096}k_5 h\right)
\end{aligned} \tag{283}$$

- The ODE is solved by using the fifth order scheme (282).
- *a posteriori* error estimate is obtained by computing the difference between 4^{th} and 5^{th} order solutions at each time step.

6 (embedded RK4/5) instead of 10 (separate RK4 and RK5) evaluations

4.5.6 Implicit RK methods

- The stability of explicit RK methods can be studies very similar to LMS methods.
- Similar to that case, explicit RK methods are only conditionally stable.
- Explicit Runge-Kutta methods are unsuitable for stiff systems or problems were mainly the first few modes are excited (*e.g.*, structural dynamic applications) because of their small region of absolute stability. That is, stability stipulates time steps that are much smaller than what is needed from accuracy perspectives for these problems.
- Implicit RK methods with very large regions of absolute stability, on the other hand, can be formulated by having a full matrix a matrix as shown in the following butcher tableau:

c_1	a_{11}	$\dots$	a_{1s}
$\dots$	$\dots$	$\dots$	$\dots$
c_s	a_{s1}	$\dots$	a_{ss}
	b_1	$\dots$	b_s

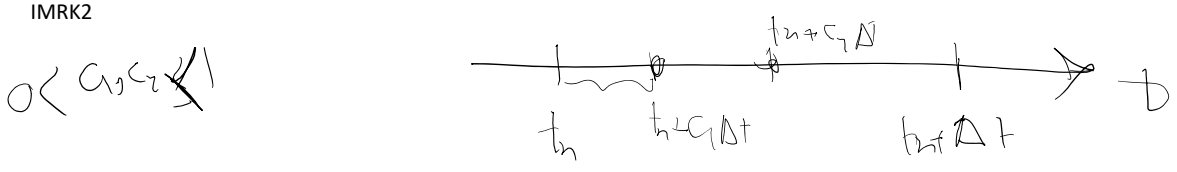
With implicit RKs all the k's are coupled, potentially through nonlinear equations for f (in terms of y)

- The update can be written as,

$$y_{n+1} = y_h + \Delta t \sum_{i=1}^{s} b_i k_i \qquad \text{where} \tag{285a}$$

$$k_i = f(t_n + \Delta t c_i, y_n + \Delta t \sum_{j=1}^{s} a_{ij} k_j), \qquad 1 \le i \le s \tag{285b}$$

IMRK2



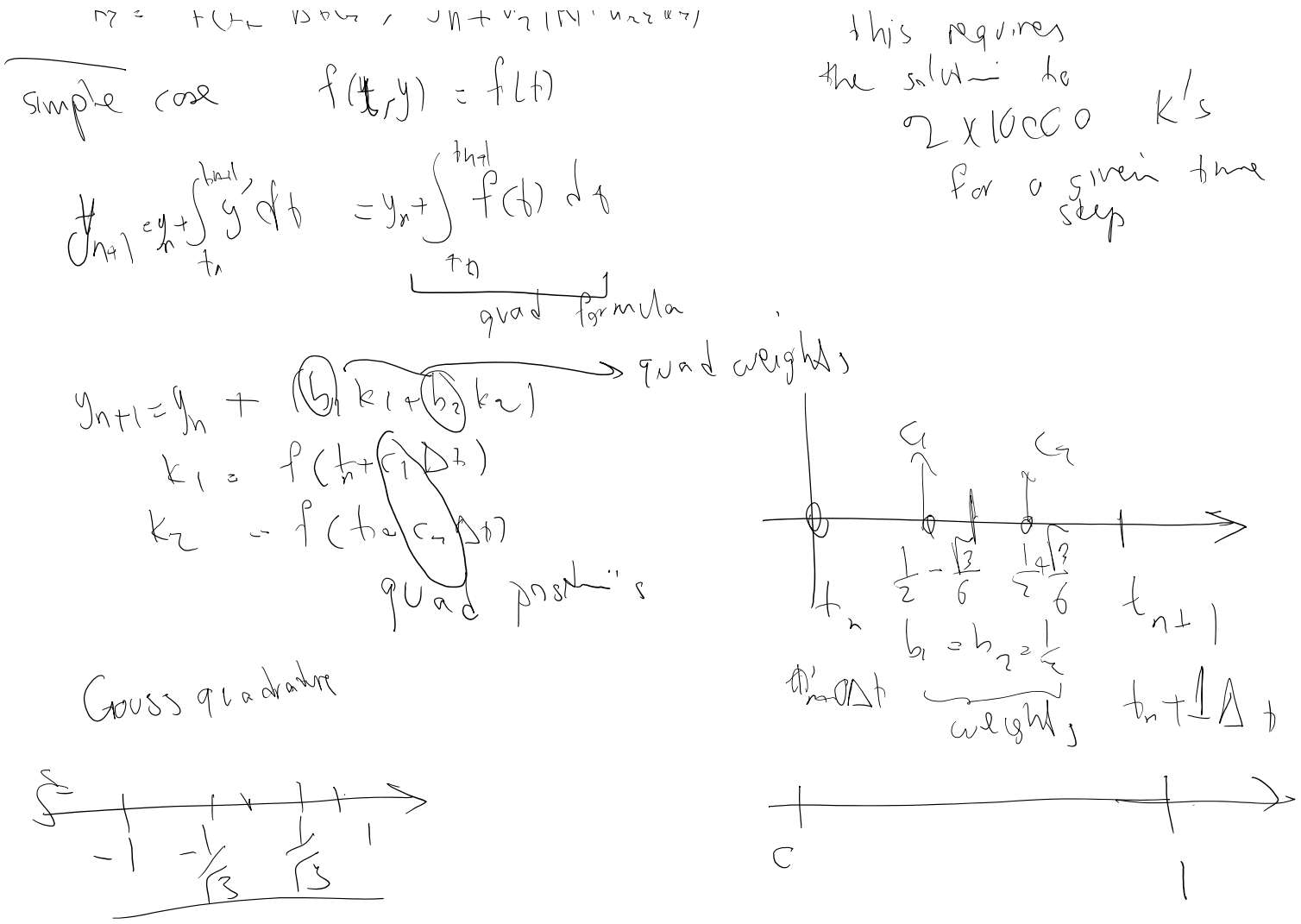
$y_{n+1} = y_n + \Delta t(b_1 k_1 + b_2 k_2)$

$k_1 = f(t_n + \Delta t c_1, y_n + a_{11}k_1 + a_{12}k_2)$

$k_2 = f(t_n + \Delta t c_2, y_n + a_{21}k_1 + a_{22}k_2)$

10000 y's for each step

this requires the solution to



1. c_i and b_i are set to the quadrature points and weights, respectively, in the Gauss quadrature formula in the evaluation of polynomials on [0 1],

$$\int_0^1 P(x)\,\mathrm{d}x = \sum_{i=1}^{s} b_i P(c_i) \tag{286}$$

for polynomials up to order $2s-1$.

2. The numbers a_{ij} can then be chosen so that the method has order 2s, and is A-stable.

- For example, the butcher tableau,

$$\begin{array}{c|cc} \frac{1}{6}(3-\sqrt{3}) & \frac{1}{4} & \frac{1}{12}(3-2\sqrt{3}) \\ \frac{1}{6}(3+\sqrt{3}) & \frac{1}{12}(3+2\sqrt{3}) & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{287}$$

defines a 2-stage ($s=2$) A-stable method of order 4.

- Implicit RK methods are rarely used due to the following reasons,
  - Unlike explicit RK methods were k_i could be solved in succession ($k_i = 1,\ldots,s$), for implicit RK methods k_i must be solved simultaneously.
  - That is, if we solved an m dof MDOF system with an s-stage implicit RK scheme, we need to solved a coupled system of size $m \times s$ for each time step!
  - This can be a huge drawback both from computational costs and memory perspectives.
  - If $f(t,y)$ is nonlinear in y the solution can become prohibitive as we need to solve now a $m \times s$ coupled system of nonlinear equations for each step update!

- For these reasons implicit Runge-Kutta methods cannot compete in efficiency with the Backward Differentiation methods (which are a group of LMS methods with very large absolute stability region), and their use is almost exclusively limited to stiff systems of ODEs.

Introduction 366

5 Mathematical analysis of time marching schemes

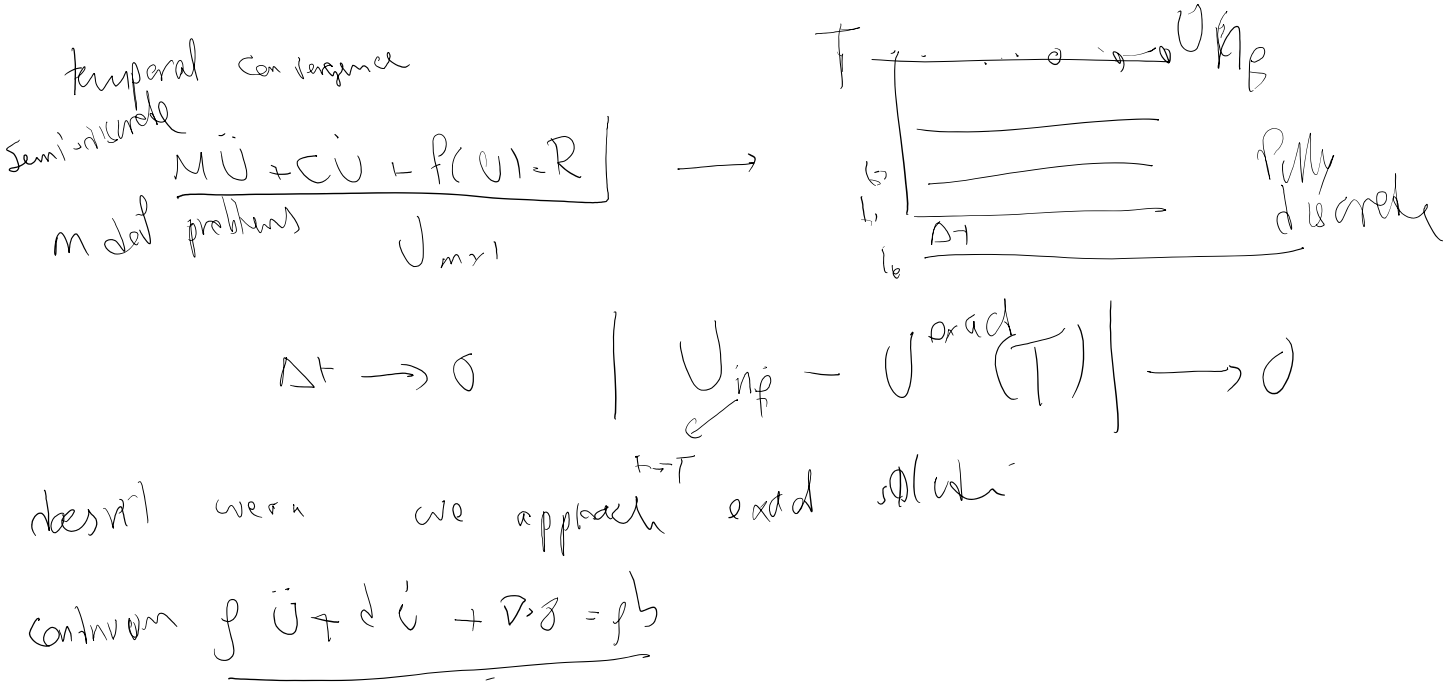
Introduction: Convergence, Consistency, and Stability 367

5.1 Introduction

An informal overview of these three important topics (before discussing them for different methods and time integration schemes):

- Convergence: The numerical method convergence to the exact solution of the underlying problem as the relevant grid sizes decrease and/or interpolation degree increases. This is an analysis limit type argument, meaning that we can make the numerical solution as close as we want to the exact solution of the underlying equation by choosing small enough grid size(s) and/or interpolation order.

The concept of the underlying equation is very important. For example for a time integration scheme that solves an FEM discretized equation $\mathbf{M\ddot{U}} + \mathbf{C\dot{U}} + \mathbf{KU} = \mathbf{R}$ consistency refers to capturing the analytical solution of the underlying ODE $\mathbf{M\ddot{U}} + \mathbf{C\dot{U}} + \mathbf{KU} = \mathbf{R}$ not the PDE that the FEM derived ODE $\mathbf{M\ddot{U}} + \mathbf{C\dot{U}} + \mathbf{KU} = \mathbf{R}$ is based on. To converge to the exact solution of the underlying PDE: *e.g.*, $\rho A\frac{\mathrm{d}^2u}{\mathrm{d}t^2} - EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = q$ for 1D elastodynamics (E is constant), we need to let spatial grid size $h \to 0$ so that the exact solution to the ODE $\mathbf{M\ddot{U}} + \mathbf{C\dot{U}} + \mathbf{KU} = \mathbf{R}$ is close enough to $\rho A\frac{\mathrm{d}^2u}{\mathrm{d}t^2} - EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = q$ then use small enough time step Δt so that the time-marching based numerical solution of $\mathbf{M\ddot{U}} + \mathbf{C\dot{U}} + \mathbf{KU} = \mathbf{R}$ is closed to its exact value. Finally, by using triangular inequality, we can argue that for small enough $h, \Delta t$ the numerical solution to the ODE $\mathbf{M\ddot{U}} + \mathbf{C\dot{U}} + \mathbf{KU} = \mathbf{R}$ is close enough to the exact solution of the PDE $\rho A\frac{\mathrm{d}^2u}{\mathrm{d}t^2} - EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = q$.



$\mathbf{KU} = \mathbf{R}$ (where only $\Delta t \to 0$ is needed) or the underlying PDE $\rho A\frac{\mathrm{d}^2u}{\mathrm{d}t^2} - EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = q$ (where both $\Delta t, h \to 0$ are needed).

Convergence rate: Is the rate in which the error between numerical and analytical solution goes to zero. For example for a method that the error is $\mathcal{O}(h^p) + \mathcal{O}(\Delta t^s)$ we call th convergence rate in space (h) is p and in time (Δt) is s.

---- Consistency

which noting $E/Z = Z/\rho = \sqrt{E/\rho} = c$ we have,

$$S_m^{n+1} = S_m^n + \frac{\bar{k}}{2}\left\{Z\left(V_{m+1}^n - V_{m-1}^n\right) + \left(S_{m+1}^n + S_{m-1}^n - 2S_m^n\right)\right\} \qquad \text{Stress update using Riemann fluxes} \qquad (113a)$$

$$V_m^{n+1} = V_m^n + \frac{\bar{k}}{2}\left\{\frac{1}{Z}\left(S_{m+1}^n - S_{m-1}^n\right) + \left(V_{m+1}^n + V_{m-1}^n - 2V_m^n\right)\right\} \qquad \text{Velocity update using Riemann fluxes} \qquad (113b)$$

- We observe that compared to update equations with average flux option (108), (113) has the additional terms in red.
- To better understand what equation (113) represents, we write in FD form (by multiplying (112) by $\frac{1}{k}$,

$$\frac{S_m^{n+1} - S_m^n}{k} - E\frac{V_{m+1}^n - V_{m-1}^n}{2h} - \frac{hc}{2}\frac{S_{m+1}^n + S_{m-1}^n - 2S_m^n}{h^2} = 0 \qquad (114a)$$

$$\frac{V_m^{n+1} - V_m^n}{k} - \frac{1}{\rho}\frac{S_{m+1}^n - S_{m-1}^n}{2h} - \frac{hc}{2}\frac{V_{m+1}^n + V_{m-1}^n - 2V_m^n}{h^2} = 0 \qquad (114b)$$

- (7) FD equations approximate the equations,

$$s_{,t} + (-E)v_{,x} - D_h s_{,xx} = 0 \qquad (115a)$$

$$v,t + (-\frac{1}{\rho})s_{,x} - D_h v_{,xx} = 0 \qquad (115b)$$

- We observe that compared to (100) the diffusion terms with diffusion coefficient,

$$D_h = \frac{hc}{2} \qquad \text{Numerical diffusion coefficient} \qquad (116)$$

  are added to both equations: ($s_{,t} - D_h s_{,xx}$ and $v,t - D_h v_{,xx}$).
- Here are some points about the added diffusion terms:
  - First, compared to average fluxes these fluxed are obtained by solving the exact fluxes on the cell interfaces.
  - The diffusion term tends to zero as grid is refined: $D_h = \frac{hc}{2} \to 0$ as $h \to 0$.
  - When h is large the diffusion terms further stabilize the solution by damping solution oscillations.

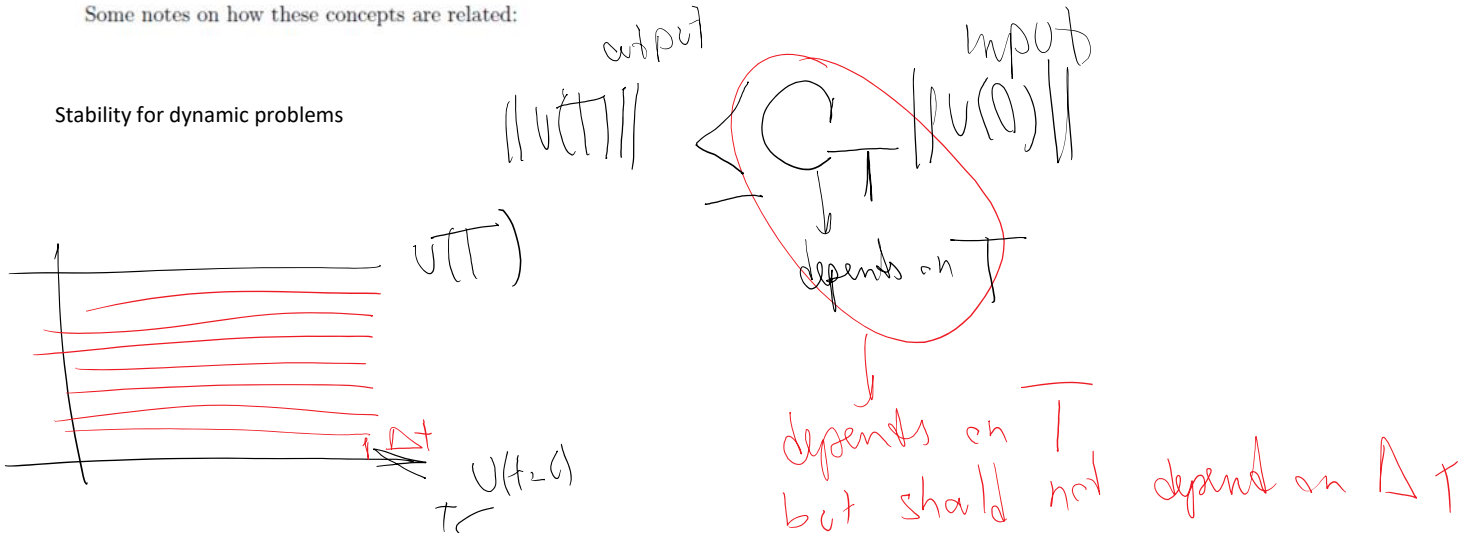
- Consistency: Consistency is a concept that is relevant to step-by-step advancing schemes. This is particularly to any time marching method that advances the solution one time step at a time. Consistency is an easier condition that convergence and only requires that ONE time advance step be "consistent" with the underlying exact solution. It basically requires that for a sufficiently smooth exact solution from time step t_n to t_{n+1} if both exact and numerical solutions start from the same initial condition at t_n the truncation error which is the error at the end of step t_{n+1} between the exact an numerical time integration scheme goes "sufficiently fast" to zero. This "sufficiently fast" will be quantified in the context of different method.

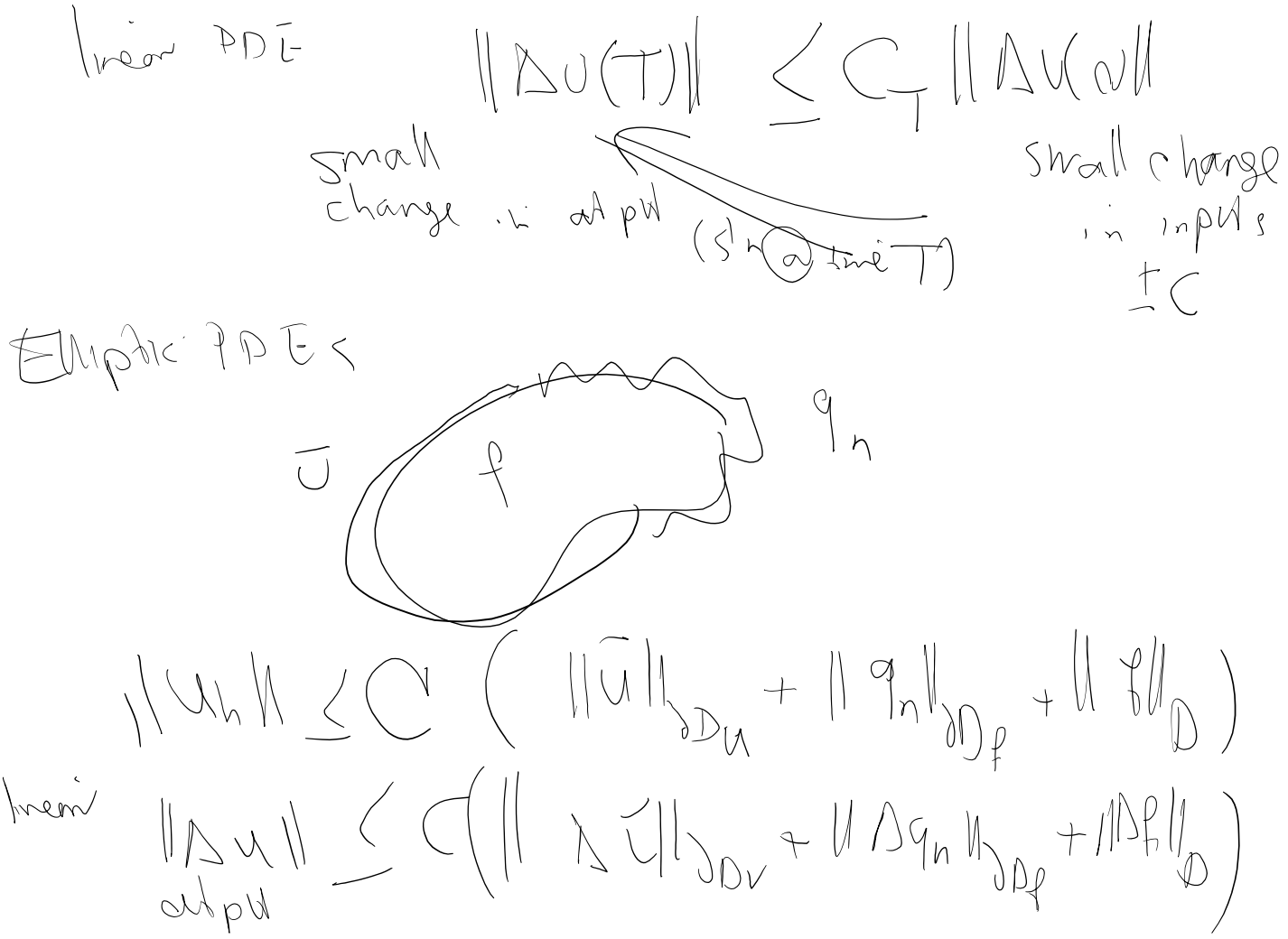
  Consistency condition is a much easier condition to verify than convergence as it includes only algebraic operations. It also deals with once (time) advance step / local truncation error vs. total solution (*e.g.*, final solution time) / and global error which is used in convergence analysis. We will see that consistency is one of the two conditions used to prove convergence.

  - Stability For a (time) advancing scheme stability requires that the solution at a time T is bounded by the solution at the initial time with a factor C_T which only depends on the given time T **not** the time step Δt. For a stable underlying PDE/ODE (where the physical solution does not blow up in time), stability requires the numerical solution too does not blow up in time.

Some notes on how these concepts are related:

Stability for dynamic problems





- Lax-Richtmyer equivalence theorem in FD states that for a consistent FD scheme convergence and stability are equivalent:

$$\text{Consistency} \quad \Rightarrow \quad (\text{ Stability} \quad \Leftrightarrow \quad \text{Convergence }) \tag{288}$$

- The way this theorem is used in practice is as follows:

$$\text{Consistency and Stability} \quad \Rightarrow \quad \text{Convergence} \tag{289}$$

  - because eventually we want to have convergent numerical methods.
  - However, the proof of convergence is very difficult as we need to consider arbitrary initial and boundary condition and using analysis tools show that the limit of numerical solution as the grid resolution goes to zero (and/or interpolation order increases) the numerical solution tends to the exact solution.
  - The Lax-Richtmyer scheme shows that if we can prove the easier conditions of consistency and stability, which are generally more straightforward and require simple algebraic/arithmetic operations, we prove convergence.
  - Various form of similar theorems exist with other numerical methods, *e.g.*, FEM, FV, DG, *etc.*, where solution is discretized differently in space, yet the same conclusion is made for the dynamic solutions in time: consistency + stability $\Rightarrow$ convergence.

  - The proof and discussion of consistency and stability will be the focus of this section.
  - We will observe that the behavior of local truncation error in consistency verification also determines convergence rate!

5.2 Analysis of direct time integration methods (for FEMs): A sample analysis

Consider the hyperbolic $\mathbf{M}\ddot{\mathbf{U}}+\mathbf{C}\dot{\mathbf{U}}+\mathbf{K}\mathbf{U}=\mathbf{R}$ and parabolic (or two/multi-field first temporal order representation of a hyperbolic) $\mathbf{M}\dot{\mathbf{U}}+\mathbf{K}\mathbf{U}=\mathbf{R}$ n dof ODEs from (226). The analysis of time integration of FEM-based discrete ODEs requires the following steps:

1. Modal reduction to SDOF: We first reduce $\mathbf{M}\ddot{\mathbf{U}}+\mathbf{C}\dot{\mathbf{U}}+\mathbf{K}\mathbf{U}=\mathbf{R}$ or $\mathbf{M}\dot{\mathbf{U}}+\mathbf{K}\mathbf{U}=\mathbf{R}$ to n SDOFs in the form $\ddot{x}+2\xi\omega\dot{x}+\omega^2 x=f(t)$ or $\dot{x}+\lambda x=f(t)$, respectively; *cf.* (229). We show that the analysis of the underling matrix form ODE reduces to the analysis of n SDOFs.

2. Stability of SDOF: For the SDOFs we analyze their stability based on the time step Δt and SDOF parameters ξ,ω (2^{nd}order ODE), and λ for **all** modes 1 to n. If conditionally stable, the maximum time step $\Delta t_{\max}$ is chosen as the **minimum** of all SDOF time steps.

3. Consistency of SDOF: We show that local truncation error $\tau(t_n)$ is $\mathcal{O}(\Delta t^s)$, for $s>0$. This is only based on analyzing the numerical error for **one** time step.

4. Convergence of SDOF: Using consistency **and** stability results, we prove the convergence of the time integration scheme and show that the temporal convergence rate is k.

Some important considerations are:

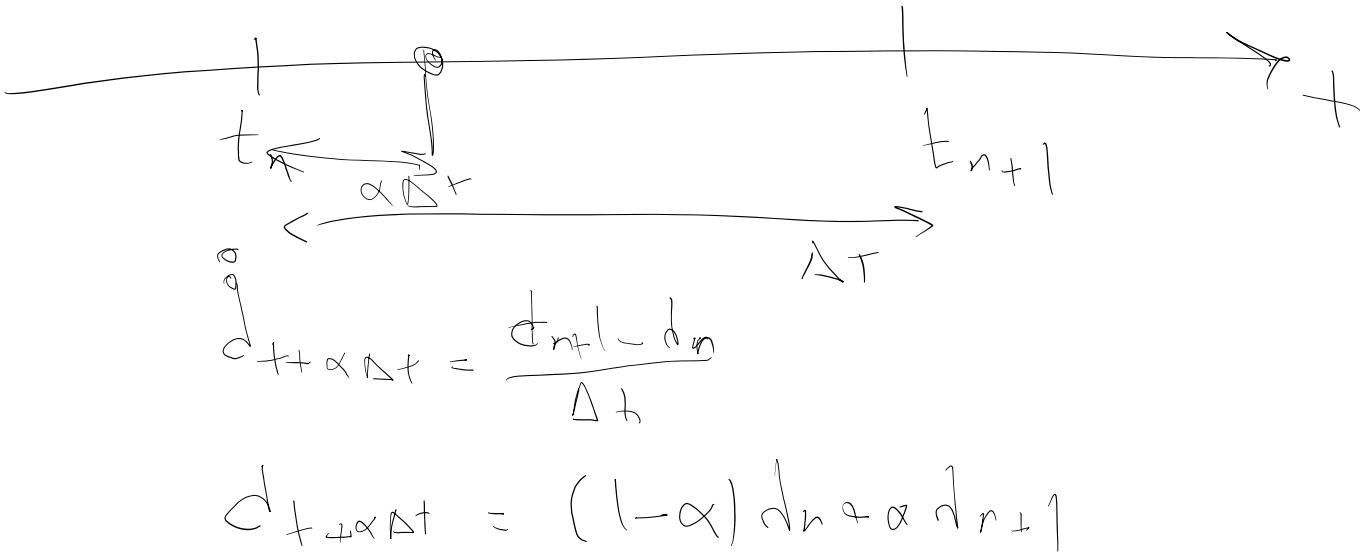
- Worst SDOF system (*i.e.*, highest n natural frequency, *etc.*): Finding the worst SDOF that gives the lowest time step (for conditionally stable methods) and in general for error analysis itself is computationally prohibitive; it requires a complete modal analysis which is expensive!

  Fortunately, a simple analysis shows that for example for a second order temporal PDE, the highest natural frequency is **smaller** that the worst case element which is generally the smallest element in the domain. So, in fact, we do not need to solve for an n dof FEM model's modal parameters! We can use the worst case element parameters as conservative estimates. This is the practice for first or second order ODEs discussed above.

- Dissipation, dispersion, and other errors: One important consideration is how much the amplitude of moving waves decreases or basically energy is dissipated with a stable time integration Scheme. Equally important is how the period (or frequency) of a periodic moving wave is modified by the numerical time integration scheme. The latter error is called dispersion or period error. Both errors play important roles in the overall accuracy of the solution. We also comment on some other aspects of numerical error, *i.e.*, features such as overshoot and undershoot.

Analysis of Generalized trapezoidal rule (α-method)

- We briefly repeat some material from §4.2 and complete the analysis of generalized trapezoidal rule.
- From (230) we consider the solution of an n dof first order ODE obtained by FEM spatial discretization: $\mathbf{M}\dot{\mathbf{d}}+\mathbf{K}\mathbf{d}=\mathbf{F}$ with IC $\mathbf{d}(t=0)=\mathbf{d}_0$.
- The update equation for the time $t=t_n+\alpha\Delta t$ was given by (231). That is, $\dot{\mathbf{d}}^{t_n+\alpha\Delta t}=\frac{\mathbf{d}^{n+1}-\mathbf{d}^n}{\Delta t}$ and $\mathbf{d}^{t_n+\alpha\Delta t}=(1-\alpha)\mathbf{d}^n+\alpha\mathbf{d}^{n+1}$.
- Below we describe how we can analyze the method by reducing it to n SDOF problems.



5.2.1 Generalized trapezoidal rule: Modal reduction to SDOF

- We perform modal analysis for the first order ODE below,

$$\dot{d}_l + \lambda_l d_l = f_l \quad \text{(scalars; SDOFs)} \longleftarrow \begin{cases} M\dot{d} + Kd = F \Longrightarrow \text{Modal eigenproblem} \\ (K - \lambda_i^h M)\psi_i = 0, \quad i \in \{1, 2, \ldots, n_{eq}\} \end{cases}$$

where

$$\begin{cases} 0 \le \lambda_1^h \le \lambda_2^h \le \cdots \le \lambda_{n_{eq}}^h \\ \psi_m^T M \psi_n = \delta_{mn} \quad \text{(orthonormality)} \Longrightarrow \\ \psi_m^T K \psi_n = \lambda_i^h \delta_{in} \quad \text{(no sum)} \end{cases} \tag{290}$$

similar to the second order ODE $\mathbf{M\ddot{U} + C\dot{U} + KU = R}$ we observe modes ψ_i are M-orthonormal and K orthogonal with diagonal values λ_i^h which are natural eigenvalues.

$$\psi_l^T K \psi_m = \lambda_l \delta_{lm} \qquad \text{no summation } l$$

$$\psi_l^T M \psi_m = \delta_{lm}$$

$$d_n = \sum_m d_{n(m)} \psi_m$$

Claim: Time marching of the MDOF with Delta T is equivalent to time marching of all the SDOFs with Delta T

$$d \longrightarrow \text{solution of } M\dot{d} + Kd = F \quad @ \; t_{n+\alpha}\Delta t$$

difference formula @ $t_{n+\alpha}\Delta t$

$$M\left(\frac{d_{n+1} - d_n}{\Delta t}\right) + K\left((1-\alpha)d_n + \alpha d_{n+1}\right) = F_{n+\alpha\Delta t}$$

$\psi_l^T \times$

$$\left(M + \alpha\Delta t K\right) d_{n+1} + \left(-M + (1-\alpha)\Delta t K\right) d_n = \Delta t F_{n+\alpha\Delta t}$$

$$\psi_l^T M \sum \psi_m d_{n+1(m)} + \alpha\Delta t \; \psi_l^T K \sum \psi_m d_{n+1(m)} + \cdots$$

$$\left[d_{n+1(l)} + \alpha\Delta t \lambda_l d_{n+1(l)} \right] + \left[-1 + (1-\alpha)\lambda_l \Delta t \right] d_{n(l)}$$

$$= \Delta t \, F_{n+\alpha\Delta t(l)}$$

$$\longrightarrow d_{n+1(l)} - d_{n(l)} + \lambda_l \left[(1-\alpha) d_{n(l)} + \alpha d_{n+1(l)} \right] = F_{n+\alpha\Delta t(l)}$$

$$\rightarrow \frac{d_{n+1(l)} - d_{n(l)}}{\Delta t} + \lambda_l \left[(1-\alpha)\, d_{n(l)} + \alpha\, d_{n+1(l)} \right] = f_{n+\alpha\Delta t(l)}$$

$$\dot{d}_{t_n+\alpha\Delta t(l)} + \lambda_l \, d_{t_n+\alpha\Delta t(l)} = F_{n+\alpha\Delta t(l)}$$

- Time integration scheme directly applied to $\mathbf{M}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{F}$ is equivalent to integrating SDOFs with the same integration scheme.
- This can be demonstrated for generalized trapezoidal rule:
  - Equation (294)(e) is generalized trapezoidal rule applied to $\mathbf{M}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{F}$ premultiplied ψ_l^{T} where
  - $\mathbf{d}_n$ and $\mathbf{d}_{n+1}$ are expressed in terms of modal components.
  - Now, using M-orthonormal and K orthogonal (with diagonal values λ_i^h) properties from (290) MDOF generalized trapezoidal method for MDOF system in (294)(e) (premultiplied by ψ_l) results SDOF generalized trapezoidal method for SDOFs in (294)(g).

$$\boldsymbol{d}_n = \sum_{m=1}^{n_{eq}} d_{n(m)} \boldsymbol{\psi}_m \tag{a}$$

$$\boldsymbol{d}_{n+1} = \sum_{m=1}^{n_{eq}} d_{n+1(m)} \boldsymbol{\psi}_m \tag{b}$$

$$d_{n(l)} = \boldsymbol{\psi}_l^T \boldsymbol{M} \boldsymbol{d}_n \tag{c}$$

$$d_{n+1(l)} = \boldsymbol{\psi}_l^T \boldsymbol{M} \boldsymbol{d}_{n+1} \tag{d}$$

$$\sum_{m=1}^{n_{eq}} \left[d_{n+1(m)} \boldsymbol{\psi}_l^T (\boldsymbol{M} + \alpha \Delta t\, \boldsymbol{K}) \boldsymbol{\psi}_m - d_{n(m)} \boldsymbol{\psi}_l^T (\boldsymbol{M} - (1-\alpha)\Delta t \boldsymbol{K}) \boldsymbol{\psi}_m \right] = \Delta t\, \boldsymbol{\psi}_l^T \boldsymbol{F}_{n+\alpha} \tag{e}$$

$$\boldsymbol{F}_{n+\alpha} = (1-\alpha)\boldsymbol{F}_n + \alpha \boldsymbol{F}_{n+1} \tag{f}$$

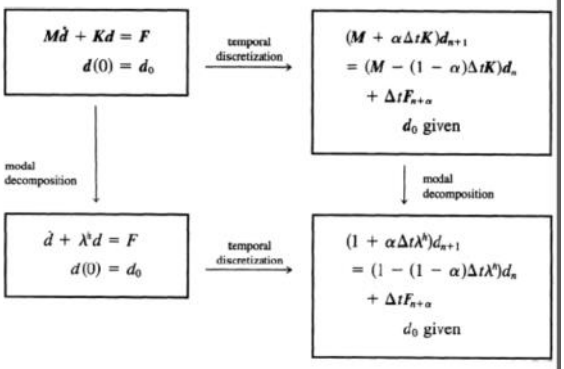
$$(1 + \alpha \Delta t\, \lambda_l^h) d_{n+1(l)} = (1 - (1-\alpha)\Delta t\, \lambda_l^h) d_{n(l)} + \Delta t\, F_{n+\alpha(l)} \tag{g}$$

(294)

- Thus, solution of MDOF $\mathbf{M}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{F}$ with generalized trapezoidal rule reduces to solving the following SDOF equations again with generalized trapezoidal rule

$$\left.\begin{aligned} (1 + \alpha \Delta t\, \lambda^h) d_{n+1} &= (1 - (1-\alpha)\Delta t\, \lambda^h) d_n + \Delta t\, F_{n+\alpha} \\ d_0 &\text{ given} \end{aligned}\right\} \quad \textit{(temporally discretized SDOF model problem)} \tag{295}$$

- The same can be shown for basically any ODE time integration scheme.
- Basically, it does not matter if we first do modal decomposition, then apply generalized trapezoidal integration to SDOFs OR first apply generalized trapezoidal integration then modal decomposition, as shown in the figure:



- Error analysis, stability analysis, *etc.* of MDOF also reduces to the analysis of SDOF.
- For this reason we define the following for MDOF solution

$$\mathbf{d}_n = \text{Numerical vector solution for MDOF at time step } t_n \tag{296a}$$

$$\mathbf{d}(t_n) = \text{Exact vector solution for MDOF at time } t = t_n \text{ (evaluated at same dofs)} \tag{296b}$$

$$\mathbf{e}(t_n) = \mathbf{d}_n - \mathbf{d}(t_n) = \text{Error vector for MDOF numerical ODE solution for relative to exact solution} \tag{296c}$$

- Similarly, we define numerical, exact, and error values for SDOF number l

$$e_{(l)}(t_n) = d_{n(l)} - d_{(l)}(t_n) \tag{297}$$

- We observe that MDOF error norm squared with kernel M is equal to the sum of squares of individual SDOF errors:

$$\|\mathbf{e}(t_n)\|_{\boldsymbol{M}}^2 = \mathbf{e}(t_n)^T \boldsymbol{M} \mathbf{e}(t_n) = \sum (e_{(l)}(t_n)\boldsymbol{\psi}_l)^T \boldsymbol{M} (e_{(m)}(t_n)\boldsymbol{\psi}_m)$$

$$\boldsymbol{e}(t_n)^T\boldsymbol{M}\boldsymbol{e}(t_n) = \sum_{l,m=1}^{n_{eq}} (e_{(l)}(t_n)\boldsymbol{\psi}_l)^T\boldsymbol{M}(e_{(m)}(t_n)\boldsymbol{\psi}_m)$$
$$= \sum_{l,m=2}^{n_{eq}} e_{(l)}(t_n)e_{(m)}(t_n)\boldsymbol{\psi}_l^T\boldsymbol{M}\boldsymbol{\psi}_m$$
$$= \sum_{l,m=1}^{n_{eq}} e_{(l)}(t_n)e_{(m)}(t_n)\delta_{lm} \qquad \text{(orthonormality)}$$
$$= \sum_{l=1}^{n_{eq}} (e_{(l)}(t_n))^2 \tag{298}$$

- Thus, the convergence of MDOF system (with M kernel) which requires $\mathbf{e}(t_n)^{\mathrm{T}}\mathbf{M}\mathbf{e}(t_n)$ is equivalent to individual convergence of SDOFs:

$$\boldsymbol{e}(t_n)^T\boldsymbol{M}\boldsymbol{e}(t_n) \longrightarrow 0 \text{ if and only if } e_{(l)}(t_n) \longrightarrow 0 \text{ for each } l \in \{1, 2, \ldots n_{eq}\} \tag{299}$$

- Finally, given that M is positive definite convergence of norm square with kernel M is equivalent to L2 convergence of $\mathbf{e}(t_n)$:

$$\boldsymbol{e}(t_n)^T\boldsymbol{M}\boldsymbol{e}(t_n) \longrightarrow 0 \text{ if and only if } \boldsymbol{e}(t_n) \longrightarrow \mathbf{0} \tag{300}$$

- **L2 Convergence of MDOF error is equivalent to convergence of scalar SDOF** $\Rightarrow$
- For convergence of MDOF we only need to investigate convergence of all SDOFs.

5.2.2 Generalized trapezoidal rule: Stability of SDOF

- As we observe, solution and even convergence analysis of MDOF $\mathbf{M}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{F}$ reduces to the solution and convergence analysis of SDOFs.
- To analyze the stability of the method, we first study how the exact solution behaves for a given modal value λ^h.

$$\dot{d} + \lambda^h d = 0 \tag{301}$$

- which has the solution:

$$d(t_{n+1}) = \exp(-\lambda^h(t_{n+1} - t_n))d(t_n) \tag{302}$$

- The exact numerical solution is stable basically when $\lambda^h \geq 0$:

$$\left.\begin{aligned} |d(t_{n+1})| < |d(t_n)|, \quad & \lambda^h > 0 \\ d(t_{n+1}) = d(t_n), \quad & \lambda^h = 0 \end{aligned}\right\} \tag{303}$$

- To study the stability of the numerical method, we find the update of (301) based on a SDOF generalized trapezoidal rule; *cf.* (294)(g),

$$(1 + \alpha\Delta t\,\lambda^h)d_{n+1} = (1 - (1 - \alpha)\Delta t\,\lambda^h)d_n \tag{304}$$

- Which is,

$$d_{n+1} = Ad_n, \qquad \text{where } A = \frac{1 - (1-\alpha)\Delta t\lambda}{1 + \alpha\Delta t\lambda} \text{ Amplification factor} \tag{305}$$

- which from (305) we observe,

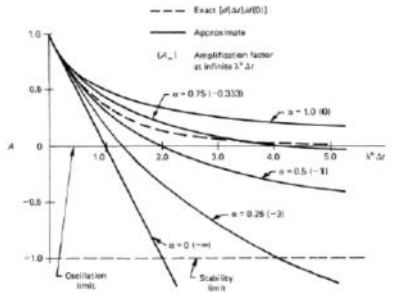
$$d^n = A^n d^n \tag{306}$$

- Clearly, numerical method is stable iff $A \leq 1$.
- now given that the exact solution is only stable for $\lambda^h \geq 0$, we are looking for conditions in which the numerical method is convergence for the same λ^h for which exact solution is stable.
- That is, we consider the condition $\lambda^h \geq 0$ and look for Δt for which $A \leq 1$:

$$|A| \leq 1 \quad \Rightarrow \quad -1 \leq \frac{1 - (1-\alpha)\Delta t\lambda}{1 + \alpha\Delta t\lambda} \leq 1 \tag{307}$$

which results in the following conditions:

$$\begin{cases} \alpha < \frac{1}{2} & \text{Conditionally stable} \quad \Delta t\lambda < \frac{2}{1-2\alpha} \\ \alpha \geq \frac{1}{2} & \text{Unconditionally stable} \end{cases} \tag{308}$$



Amplification factor for typical one-step methods.

Summary: Stability for the generalized trapezoidal methods

Amplification factor: $A = \frac{1-(1-\alpha)\Delta t\lambda^h}{1+\alpha\Delta t\lambda^h}$

Stability requirement: $|A| < 1$ for $\lambda^h = \lambda^h_{n_{eq}}$ (= maximum eigenvalue)

Unconditional stability: $\alpha \geq \frac{1}{2}$

Conditional stability: $\alpha < \frac{1}{2}$, $\Delta t < \frac{2}{(1-2\alpha)\lambda^h_{n_{eq}}}$

- The maximum stable time stable can be chosen as follows

$$\alpha \geq \frac{1}{2} \quad \Rightarrow \quad \text{any } \Delta t \qquad \text{Unconditional stability} \tag{309a}$$

All is left is to prove that SDOF is consistent

Before that a few points about stability

$\alpha < \frac{1}{2}$ $\quad \Delta t < \frac{2}{(1-2\alpha)\lambda_l}$ for all λ

$\Delta t \leq \frac{2}{(1-2\alpha)\lambda_{max}}$ → the max eigen value for $M\psi_l = \lambda_l K\psi_l$

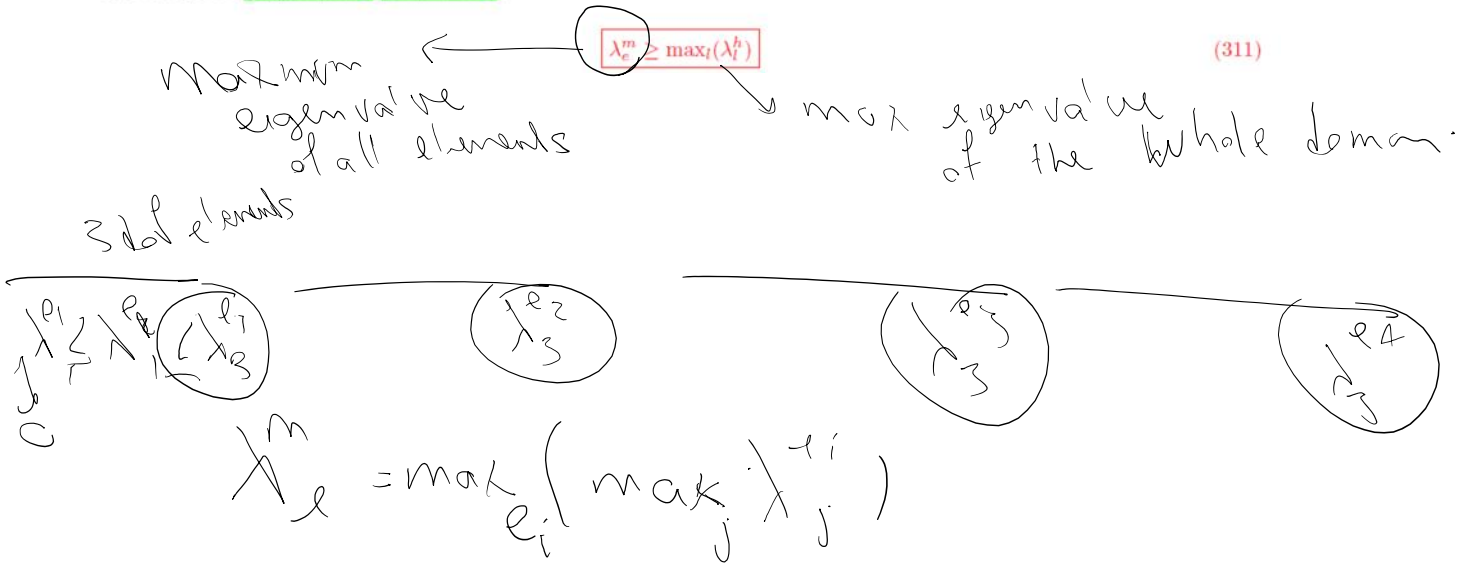
$$\alpha > \frac{1}{2} \quad \Rightarrow \quad \Delta t \leq \Delta t_{\max} = \frac{1}{\max_l(\lambda_l^h)} \frac{2}{1-2\alpha} \qquad \text{Conditional stability} \qquad (309b)$$

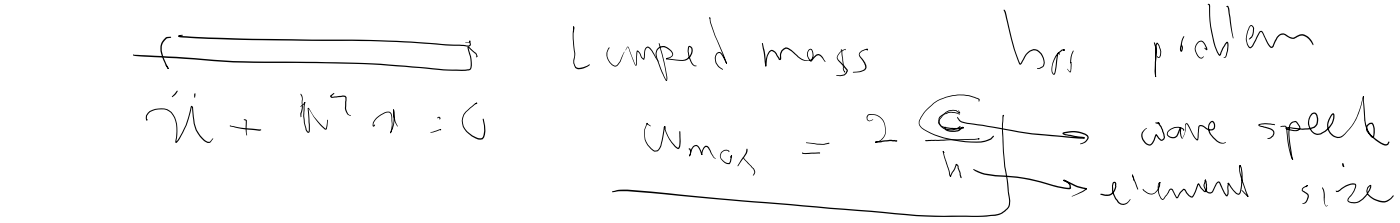
- where $\max_l(\lambda_l^h)$ is the maximum modal eigen value obtained from modal analysis. The maximum value is chosen because smaller model eigenvalues result in more loose time constraint.
- In practice it is difficult/computationally expensive to actually compute $\max_l(\lambda_l^h)$ by a modal analysis.
- Instead, we can use

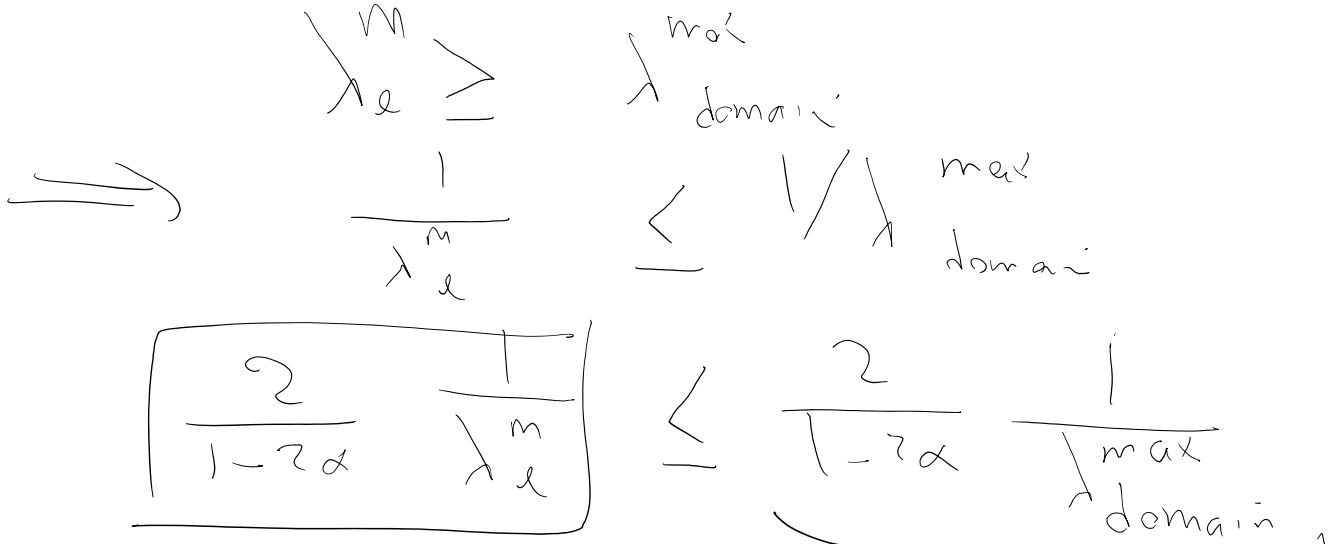
$$\lambda_e^m := \max_{e,i}(\lambda_e^i h) = \text{maximum of all element's } (e) \text{ maximum modal eigenvalue (index (i) in element)} \qquad (310)$$

  - which is the maximum modal eigenvalue that any element can produce.
  - This value is very easy to be computed and values are already obtain for various element types in the literature.
  - λ_e^m It only depends on element size (geometry) and material properties.
- One can prove [Hughes, 2012, Bathe, 2006],

$$\lambda_e^m \geq \max_l(\lambda_l^h) \qquad (311)$$







domain

Conservative max time step for generalized hyperbolic method. Δt_{max}

- The same process can be applied to other time integration schemes.
- Later, we provide analytical formulas for λ_e^m for some types of elements.
- As a note, we observe that,

$$\lambda_e^m \propto \frac{c}{h_{\min}} \qquad \text{Simple hyperbolic PDE, } e.g., \ u_{,tt} - c^2 \nabla.\nabla u = 0,\ c = \text{wave speed} \tag{314a}$$

$$\lambda_e^m \propto \frac{D}{h_{\min}^2} \qquad \text{Simple parabolic problem, } e.g., \ u_{,t} - D \nabla.\nabla u = 0,\ D = \text{damping coefficient} \tag{314b}$$

where $h_{\min}$ is the minimum element size. For simplicity here it is assumed the domain is covered with the same element type. For more general cases, we need to consider the maximum eigen-frequency of all elements which may not necessarily correspond to that of the smallest element.

- From (313) and (314) we reach to the same conclusion we had reach for simple hyperbolic and parabolic PDEs with FD methods:

$$\Delta t \leq C \begin{cases} h_{\min} & \text{Simple hyperbolic PDE, } e.g., \ u_{,tt} - c^2 \nabla.\nabla u = 0 \\ h_{\min}^2 & \text{Simple parabolic PDE, } e.g., \ u_{,t} - D \nabla.\nabla u = 0 \end{cases} \tag{315a}$$

where C depends on material properties, $e.g.$, c, D, and the particular form of time integration scheme.

- The stable time step for more complex dynamic systems, $e.g.$, $\tau u_{,tt} + u_{,t} - D \nabla.\nabla u = 0$ will be discussed later.

truncation error

$\tau = O(\Delta t^q)$

Consistency

Stability $|A| < 1$

$d_n = d(t_n)$

d_{n+1}

$\tau(t_n) = \frac{d_{n+1} - d(t_{n+1})}{\Delta t}$

removed

$|d(T) - d_f|$

exact sln

Corresponding to T

Convergence