Study Guide: Truncation Error Analysis

Hans Petter Langtangen[1,2]

Center for Biomedical Computing, Simula Research Laboratory[1]

Department of Informatics, University of Oslo[2]

Dec 12, 2013

Overview of what truncation errors are

- Definition: The *truncation error* is the discrepancy that arises from performing a finite number of steps to approximate a process with infinitely many steps.
- Widely used: truncation of infinite series, finite precision arithmetic, finite differences, and differential equations.
- Why? The truncation error is an error measure that is easy to compute.

Abstract problem setting

Consider an abstract differential equation

$$\mathcal{L}(u) = 0\,.$$

Example: $\mathcal{L}(u) = u'(t) + a(t)u(t) - b(t)$.
The corresponding discrete equation:

$$\mathcal{L}_\Delta(u) = 0\,.$$

Let now

- u be the numerical solution of the discrete equations, computed at mesh points: u^n, $n = 0, \ldots, N_t$
- u_e the exact solution of the differential equation

$$\begin{aligned}\mathcal{L}(u_e) &= 0,\\ \mathcal{L}_\Delta(u) &= 0\,.\end{aligned}$$

u is computed at mesh points

Various error measures

- Dream: the true error $e = u_e - u$, but usually impossible
- Must find other error measures that are easier to calculate
  - Derive formulas for u in (very) special, simplified cases
  - Compute empirical convergence rates for special choices of u_e (usually non-physical u_e)
- To what extent does u_e fulfill $\mathcal{L}_\Delta(u_e) = 0$?
- It does not fit, but we can measure the error $\mathcal{L}_\Delta(u_e) = R$
- R is the truncation error and it is easy to compute in general, without considering special cases

Truncation errors in finite difference formulas

Example: The backward difference for $u'(t)$

Backward difference approximation to u':

$$[D_t^- u]^n = \frac{u^n - u^{n-1}}{\Delta t} \approx u'(t_n)\,. \tag{1}$$

Define the truncation error of this approximation as

$$R^n = [D_t^- u]^n - u'(t_n)\,. \tag{2}$$

The common way of calculating R^n is to

1. expand $u(t)$ in a Taylor series around the point where the derivative is evaluated, here t_n,
2. insert this Taylor series in (2), and
3. collect terms that cancel and simplify the expression.

Taylor series

General Taylor series expansion from calculus:

$$f(x+h) = \sum_{i=0}^{\infty} \frac{1}{i!} \frac{d^i f}{dx^i}(x) h^i \, .$$

Here: expand u^{n-1} around t_n:

$$\begin{aligned} u(t_{n-1}) = u(t - \Delta t) &= \sum_{i=0}^{\infty} \frac{1}{i!} \frac{d^i u}{dt^i}(t_n)(-\Delta t)^i \\ &= u(t_n) - u'(t_n)\Delta t + \frac{1}{2} u''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^3), \end{aligned}$$

- $\mathcal{O}(\Delta t^3)$: power-series in Δt where the lowest power is Δt^3
- Small Δt: $\Delta t \gg \Delta t^3 \gg \Delta t^4$

Taylor series inserted in the backward difference approximation

$$
\begin{aligned}
[D_t^- u]^n - u'(t_n) &= \frac{u(t_n) - u(t_{n-1})}{\Delta t} - u'(t_n) \\
&= \frac{u(t_n) - (u(t_n) - u'(t_n)\Delta t + \frac{1}{2}u''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^3))}{\Delta t} \\
&\quad - u'(t_n) \\
&= -\frac{1}{2}u''(t_n)\Delta t + \mathcal{O}(\Delta t^2))
\end{aligned}
$$

Result:

$$
R^n = -\frac{1}{2}u''(t_n)\Delta t + \mathcal{O}(\Delta t^2))\,. \tag{3}
$$

The difference approximation is of *first order* in Δt. It is exact for linear $u_{\rm e}$.

The forward difference for $u'(t)$

Now consider a forward difference:

$$u'(t_n) \approx [D_t^+ u]^n = \frac{u^{n+1} - u^n}{\Delta t}\,.$$

Define the truncation error:

$$R^n = [D_t^+ u]^n - u'(t_n)\,.$$

Expand u^{n+1} in a Taylor series around t_n,

$$u(t_{n+1}) = u(t_n) + u'(t_n)\Delta t + \frac{1}{2}u''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^3)\,.$$

We get

$$R = \frac{1}{2}u''(t_n)\Delta t + \mathcal{O}(\Delta t^2)\,.$$

The central difference for $u'(t)$ (1)

For the central difference approximation,

$$u'(t_n) \approx [D_t u]^n, \quad [D_t u]^n = \frac{u^{n+\frac{1}{2}} - u^{n-\frac{1}{2}}}{\Delta t},$$

the truncation error is

$$R^n = [D_t u]^n - u'(t_n)\,.$$

Expand $u(t_{n+\frac{1}{2}})$ and $u(t_{n-1/2})$ in Taylor series around the point t_n where the derivative is evaluated:

$$\begin{aligned}
u(t_{n+\frac{1}{2}}) =& u(t_n) + u'(t_n)\frac{1}{2}\Delta t + \frac{1}{2}u''(t_n)(\frac{1}{2}\Delta t)^2 + \\
& \frac{1}{6}u'''(t_n)(\frac{1}{2}\Delta t)^3 + \frac{1}{24}u''''(t_n)(\frac{1}{2}\Delta t)^4 + \mathcal{O}(\Delta t^5) \\
u(t_{n-1/2}) =& u(t_n) - u'(t_n)\frac{1}{2}\Delta t + \frac{1}{2}u''(t_n)(\frac{1}{2}\Delta t)^2 - \\
& \frac{1}{6}u'''(t_n)(\frac{1}{2}\Delta t)^3 + \frac{1}{24}u''''(t_n)(\frac{1}{2}\Delta t)^4 + \mathcal{O}(\Delta t^5)\,.
\end{aligned}$$

The central difference for $u'(t)$ (1)

$$u(t_{n+\frac{1}{2}}) - u(t_{n-1/2}) = u'(t_n)\Delta t + \frac{1}{24}u'''(t_n)\Delta t^3 + \mathcal{O}(\Delta t^5)\,.$$

By collecting terms in $[D_t u]^n - u(t_n)$ we find R^n to be

$$R^n = \frac{1}{24}u'''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4), \tag{4}$$

Note:

- Second-order accuracy since the leading term is Δt^2
- Only even powers of Δt

Leading-order error terms in finite differences (1)

$$[D_t u]^n = \frac{u^{n+\frac{1}{2}} - u^{n-\frac{1}{2}}}{\Delta t} = u'(t_n) + R^n, \tag{5}$$

$$R^n = \frac{1}{24}u'''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4) \tag{6}$$

$$[D_{2t} u]^n = \frac{u^{n+1} - u^{n-1}}{2\Delta t} = u'(t_n) + R^n, \tag{7}$$

$$R^n = \frac{1}{6}u'''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4) \tag{8}$$

$$[D_t^- u]^n = \frac{u^n - u^{n-1}}{\Delta t} = u'(t_n) + R^n, \tag{9}$$

$$R^n = -\frac{1}{2}u''(t_n)\Delta t + \mathcal{O}(\Delta t^2) \tag{10}$$

$$[D_t^+ u]^n = \frac{u^{n+1} - u^n}{\Delta t} = u'(t_n) + R^n, \tag{11}$$

$$R^n = \frac{1}{2}u''(t_n)\Delta t + \mathcal{O}(\Delta t^2) \tag{12}$$

Leading-order error terms in finite differences (2)

$$[\bar{D}_t u]^{n+\theta} = \frac{u^{n+1} - u^n}{\Delta t} = u'(t_{n+\theta}) + R^{n+\theta}, \tag{13}$$

$$R^{n+\theta} = \frac{1}{2}(1 - 2\theta)u''(t_{n+\theta})\Delta t - \frac{1}{6}((1-\theta)^3 - \theta^3)u'''(t_{n+\theta})\Delta t^2 + \mathcal{O}(\tag{14}$$

$$[D_t^{2-} u]^n = \frac{3u^n - 4u^{n-1} + u^{n-2}}{2\Delta t} = u'(t_n) + R^n, \tag{15}$$

$$R^n = -\frac{1}{3}u'''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^3) \tag{16}$$

$$[D_t D_t u]^n = \frac{u^{n+1} - 2u^n + u^{n-1}}{\Delta t^2} = u''(t_n) + R^n, \tag{17}$$

$$R^n = \frac{1}{12}u''''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4) \tag{18}$$

Leading-order error terms in mean values (1)

Weighted arithmetic mean:

$$[\overline{u}^{t,\theta}]^{n+\theta} = \theta u^{n+1} + (1-\theta)u^n = u(t_{n+\theta}) + R^{n+\theta}, \tag{19}$$

$$R^{n+\theta} = \frac{1}{2}u''(t_{n+\theta})\Delta t^2\theta(1-\theta) + \mathcal{O}(\Delta t^3)\,. \tag{20}$$

Standard arithmetic mean:

$$[\overline{u}^{t}]^{n} = \frac{1}{2}(u^{n-\frac{1}{2}} + u^{n+\frac{1}{2}}) = u(t_n) + R^n, \tag{21}$$

$$R^n = \frac{1}{8}u''(t_n)\Delta t^2 + \frac{1}{384}u''''(t_n)\Delta t^4 + \mathcal{O}(\Delta t^6)\,. \tag{22}$$

Leading-order error terms in mean values (2)

Geometric mean:

$$[\overline{u^2}^{t,g}]^n = u^{n-\frac{1}{2}}u^{n+\frac{1}{2}} = (u^n)^2 + R^n, \tag{23}$$

$$R^n = -\frac{1}{4}u'(t_n)^2\Delta t^2 + \frac{1}{4}u(t_n)u''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4)\,. \tag{24}$$

Harmonic mean:

$$[\overline{u}^{t,h}]^n = u^n = \frac{2}{\frac{1}{u^{n-\frac{1}{2}}} + \frac{1}{u^{n+\frac{1}{2}}}} + R^{n+\frac{1}{2}}, \tag{25}$$

$$R^n = -\frac{u'(t_n)^2}{4u(t_n)}\Delta t^2 + \frac{1}{8}u''(t_n)\Delta t^2\,. \tag{26}$$

Software for computing truncation errors

- Can use sympy to automate calculations with Taylor series.
- Tool: course module `truncation_errors`

```
>>> from truncation_errors import TaylorSeries
>>> from sympy import *
>>> u, dt = symbols('u dt')
>>> u_Taylor = TaylorSeries(u, 4)
>>> u_Taylor(dt)
D1u*dt + D2u*dt**2/2 + D3u*dt**3/6 + D4u*dt**4/24 + u
>>> FE = (u_Taylor(dt) - u)/dt
>>> FE
(D1u*dt + D2u*dt**2/2 + D3u*dt**3/6 + D4u*dt**4/24)/dt
>>> simplify(FE)
D1u + D2u*dt/2 + D3u*dt**2/6 + D4u*dt**3/24
```

Notation: `D1u` for u', `D2u` for u'', etc.

See `trunc/truncation_errors.py`.

Symbolic computing with difference operators

A class `DiffOp` represents many common difference operators:

```
>>> from truncation_errors import DiffOp
>>> from sympy import *
>>> u = Symbol('u')
>>> diffop = DiffOp(u, independent_variable='t')
>>> diffop['geometric_mean']
-D1u**2*dt**2/4 - D1u*D3u*dt**4/48 + D2u**2*dt**4/64 + ...
>>> diffop['Dtm']
D1u + D2u*dt/2 + D3u*dt**2/6 + D4u*dt**3/24
>>> diffop.operator_names()
['geometric_mean', 'harmonic_mean', 'Dtm', 'D2t', 'DtDt',
 'weighted_arithmetic_mean', 'Dtp', 'Dt']
```

Names in `diffop`: `Dtp` for D_t^+, `Dtm` for D_t^-, `Dt` for D_t, `D2t` for D_{2t}, `DtDt` for D_tD_t.

Truncation errors in exponential decay ODE

$$u'(t) = -au(t)$$

Truncation error of the Forward Euler scheme

The Forward Euler scheme:

$$[D_t^+ u = -au]^n \,. \tag{27}$$

Definition of the truncation error R^n:

$$[D_t^+ u_{\rm e} + au_{\rm e} = R]^n \,. \tag{28}$$

From (11)-(12):

$$[D_t^+ u_{\rm e}]^n = u_{\rm e}'(t_n) + \frac{1}{2}u_{\rm e}''(t_n)\Delta t + \mathcal{O}(\Delta t^2)\,.$$

Inserted in (28):

$$u_{\rm e}'(t_n) + \frac{1}{2}u_{\rm e}''(t_n)\Delta t + \mathcal{O}(\Delta t^2) + au_{\rm e}(t_n) = R^n \,.$$

Note: $u_{\rm e}'(t_n) + au_{\rm e}^n = 0$ since $u_{\rm e}$ solves the ODE. Then

$$R^n = \frac{1}{2}u_{\rm e}''(t_n)\Delta t + \mathcal{O}(\Delta t^2)\,. \tag{29}$$

Truncation error of the Crank-Nicolson scheme

Crank-Nicolson:

$$[D_t u = -au]^{n+\frac{1}{2}}, \tag{30}$$

Truncation error:

$$[D_t u_{\rm e} + a\overline{u_{\rm e}}^t = R]^{n+\frac{1}{2}} . \tag{31}$$

From (5)-(6) and (21)-(22):

$$
\begin{aligned}
{[D_t u_{\rm e}]}^{n+\frac{1}{2}} &= u'(t_{n+\frac{1}{2}}) + \frac{1}{24}u_{\rm e}'''(t_{n+\frac{1}{2}})\Delta t^2 + \mathcal{O}(\Delta t^4),\\
{[a\overline{u_{\rm e}}^t]}^{n+\frac{1}{2}} &= u(t_{n+\frac{1}{2}}) + \frac{1}{8}u''(t_n)\Delta t^2 + +\mathcal{O}(\Delta t^4)
\end{aligned}
$$

Inserted in the scheme we get

$$R^{n+\frac{1}{2}} = \left(\frac{1}{24}u_{\rm e}'''(t_{n+\frac{1}{2}}) + \frac{1}{8}u''(t_n)\right)\Delta t^2 + \mathcal{O}(\Delta t^4) \tag{32}$$

$R^n = \mathcal{O}(\Delta t^2)$ (second-order scheme)

Test the understanding!

Analyze the the truncation error of the Backward Euler scheme and show that it is $\mathcal{O}(\Delta t)$ (first order scheme).

Truncation error of the θ-rule

The θ-rule:

$$[\bar{D}_t u = -a\overline{u}^{t,\theta}]^{n+\theta}\,.$$

Truncation error:

$$[\bar{D}_t u_{\rm e} + a\overline{u_{\rm e}}^{t,\theta} = R]^{n+\theta}\,.$$

Use (13)-(14) and (19)-(20) along with $u_{\rm e}'(t_{n+\theta}) + au_{\rm e}(t_{n+\theta}) = 0$ to show

$$\begin{aligned}R^{n+\theta} =& (\frac{1}{2} - \theta)u_{\rm e}''(t_{n+\theta})\Delta t + \frac{1}{2}\theta(1-\theta)u_{\rm e}''(t_{n+\theta})\Delta t^2 + \\ & \frac{1}{2}(\theta^2 - \theta + 3)u_{\rm e}'''(t_{n+\theta})\Delta t^2 + \mathcal{O}(\Delta t^3)\end{aligned} \tag{33}$$

Note: 2nd-order scheme if and only if $\theta = 1/2$.

Using symbolic software

Can use sympy and the tools in `truncation_errors.py`:

```
def decay():
    u, a = sm.symbols('u a')
    diffop = DiffOp(u, independent_variable='t',
                    num_terms_Taylor_series=3)
    D1u = diffop.D(1)   # symbol for du/dt
    ODE = D1u + a*u     # define ODE

    # Define schemes
    FE = diffop['Dtp'] + a*u
    CN = diffop['Dt' ] + a*u
    BE = diffop['Dtm'] + a*u
    # Residuals (truncation errors)
    R = {'FE': FE-ODE, 'BE': BE-ODE, 'CN': CN-ODE}
    return R
```

The returned dictionary becomes

```
decay: {
 'BE': D2u*dt/2 + D3u*dt**2/6,
 'FE': -D2u*dt/2 + D3u*dt**2/6,
 'CN': D3u*dt**2/24,
}
```

θ-rule: see `truncation_errors.py` (long expression, very advantageous to automate the math!)

Empirical verification of the truncation error (1)

Ideas:

- Compute R^n numerically
- Run a sequence of meshes
- Estimate the convergence rate of R^n

For the Forward Euler scheme:

$$R^n = [D_t^+ u_e + au_e]^n\,. \tag{34}$$

Insert correct $u_e(t) = Ie^{-at}$ (or use method of manufactured solution in more general cases).

Empirical verification of the truncation error (2)

- Assume $R^n = C\Delta t^r$
- C and r will vary with n - must estimate r for each mesh point
- Use a sequence of meshes with $N_t = 2^{-k}N_0$ intervals, $k = 1, 2, \ldots$
- Transform R^n data to the coarsest mesh and estimate r for each coarse mesh point

See the text for more details and an implementation.

Empirical verification of the truncation error in the Forward Euler scheme

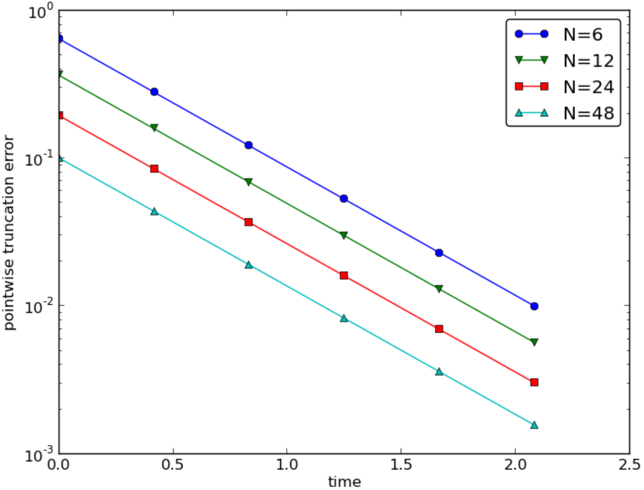

Figure: Estimated truncation error at mesh points for different meshes.

Empirical verification of the truncation error in the Forward Euler scheme

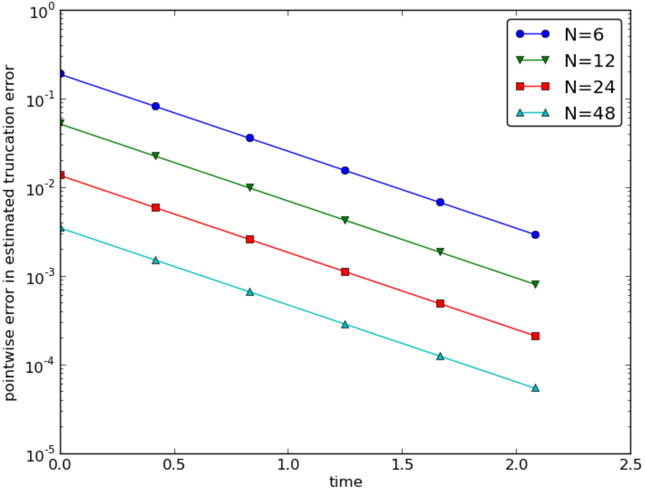

Figure: Difference between theoretical and estimated truncation error at

Increasing the accuracy by adding correction terms

Question.

Can we add terms in the differential equation that can help increase the order of the truncation error?
To be precise for the Forward Euler scheme, can we find C to make R $\mathcal{O}(\Delta t^2)$?

$$[D_t^+ u_e + a u_e = C + R]^n\,. \qquad (35)$$

$$\frac{1}{2}u_e''(t_n)\Delta t - \frac{1}{6}u_e'''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^3) = C^n + R^n\,.$$

Choosing

$$C^n = \frac{1}{2}u_e''(t_n)\Delta t,$$

makes

$$R^n = \frac{1}{6}u_e'''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^3)\,.$$

Lowering the order of the derivative in the correction term

- C^n contains u''
- Can discretize u'' (requires u^{n+1}, u^n, and u^{n-1})
- Can also express u'' in terms of u' or u

$$u' = -au, \quad \Rightarrow \quad u'' = -au' = a^2 u\,.$$

Result for $u'' = a^2 u$: apply Forward Euler to a *perturbed ODE*,

$$u' = -\hat{a}u, \quad \hat{a} = a(1 - \frac{1}{2}a\Delta t), \tag{36}$$

to make a second-order scheme!

With a correction term Forward Euler becomes Crank-Nicolson

Use the other alternative $u'' = -au'$:

$$u' = -au - \frac{1}{2}a\Delta t u' \quad \Rightarrow \quad \left(1 + \frac{1}{2}a\Delta t\right) u' = -au\,.$$

Apply Forward Euler:

$$\left(1 + \frac{1}{2}a\Delta t\right) \frac{u^{n+1} - u^n}{\Delta t} = -au^n,$$

which after some algebra can be written as

$$u^{n+1} = \frac{1 - \frac{1}{2}a\Delta t}{1 + \frac{1}{2}a\Delta t} u^n\,.$$

This is a Crank-Nicolson scheme (of second order)!

Correction terms in the Crank-Nicolson scheme (1)

$$[D_t u = -a\overline{u}^t]^{n+\frac{1}{2}},$$

Definition of the truncation error R and correction terms C:

$$[D_t u_e + a\overline{u_e}^t = C + R]^{n+\frac{1}{2}}\,.$$

Must Taylor expand

- the derivative
- the arithmetic mean

$$C^{n+\frac{1}{2}} + R^{n+\frac{1}{2}} = \frac{1}{24}u_e'''(t_{n+\frac{1}{2}})\Delta t^2 + \frac{a}{8}u_e''(t_{n+\frac{1}{2}})\Delta t^2 + \mathcal{O}(\Delta t^4)\,.$$

Let $C^{n+\frac{1}{2}}$ cancel the Δt^2 terms:

$$C^{n+\frac{1}{2}} = \frac{1}{24}u_e'''(t_{n+\frac{1}{2}})\Delta t^2 + \frac{a}{8}u_e''(t_n)\Delta t^2\,.$$

Correction terms in the Crank-Nicolson scheme (2)

- Must replace u''' and u'' in correction term
- Using $u' = -au$: $u'' = a^2 u$ and $u''' = -a^3 u$

Result: solve the perturbed ODE by a Crank-Nicolson method,

$$u' = -\hat{a}u, \quad \hat{a} = a(1 - \frac{1}{12}a^2\Delta t^2)\,.$$

and experience an error $\mathcal{O}(\Delta t^4)$.

Extension to variable coefficients

$$u'(t) = -a(t)u(t) + b(t)$$

Forward Euler:

$$[D_t^+ u = -au + b]^n\,. \tag{37}$$

The truncation error is found from

$$[D_t^+ u_{\rm e} + au_{\rm e} - b = R]^n\,. \tag{38}$$

Using (11)-(12):

$$u_{\rm e}'(t_n) - \frac{1}{2}u_{\rm e}''(t_n)\Delta t + \mathcal{O}(\Delta t^2) + a(t_n)u_{\rm e}(t_n) - b(t_n) = R^n\,.$$

Because of the ODE, $u_{\rm e}'(t_n) + a(t_n)u_{\rm e}(t_n) - b(t_n) = 0$, and

$$R^n = -\frac{1}{2}u_{\rm e}''(t_n)\Delta t + \mathcal{O}(\Delta t^2)\,. \tag{39}$$

No problems with variable coefficients!

Exact solutions of the finite difference equations

How does the truncation error depend on u_e in finite differences?

- One-sided differences: $u_e''\Delta t$ (lowest order)
- Centered differences: $u_e'''\Delta t^2$ (lowest order)
- Only harmonic and geometric mean involve u_e' or u_e

Consequence:

- $u_e(t) = ct + d$ will very often give exact solution of the discrete equations ($R = 0$)!
- Ideal for verification
- Centered schemes allow quadratic u_e

Problem: harmonic and geometric mean (error depends on u_e' and u_e)

Computing truncation errors in nonlinear problems (1)

$$u' = f(u, t) \tag{40}$$

Crank-Nicolson scheme:

$$[D_t u' = \overline{f}^t]^{n+\frac{1}{2}}\,. \tag{41}$$

Truncation error:

$$[D_t u'_{\rm e} - \overline{f}^t = R]^{n+\frac{1}{2}}\,. \tag{42}$$

Using (21)-(22) for the arithmetic mean:

$$\begin{aligned}
[\overline{f}^t]^{n+\frac{1}{2}} &= \frac{1}{2}(f(u_{\rm e}^n, t_n) + f(u_{\rm e}^{n+1}, t_{n+1}))\\
&= f(u_{\rm e}^{n+\frac{1}{2}}, t_{n+\frac{1}{2}}) + \frac{1}{8}u''_{\rm e}(t_{n+\frac{1}{2}})\Delta t^2 + \mathcal{O}(\Delta t^4)\,.
\end{aligned}$$

Computing truncation errors in nonlinear problems (2)

With (5)-(6), (42) leads to $R^{n+\frac{1}{2}}$ equal to

$$u_{\rm e}'(t_{n+\frac{1}{2}}) + \frac{1}{24}u_{\rm e}'''(t_{n+\frac{1}{2}})\Delta t^2 - f(u_{\rm e}^{n+\frac{1}{2}}, t_{n+\frac{1}{2}}) - \frac{1}{8}u_{\rm e}''(t_{n+\frac{1}{2}})\Delta t^2 + \mathcal{O}(\Delta t^4)\,.$$

Since $u_{\rm e}'(t_{n+\frac{1}{2}}) - f(u_{\rm e}^{n+\frac{1}{2}}, t_{n+\frac{1}{2}}) = 0$, the truncation error becomes

$$R^{n+\frac{1}{2}} = (\frac{1}{24}u_{\rm e}'''(t_{n+\frac{1}{2}}) - \frac{1}{8}u_{\rm e}''(t_{n+\frac{1}{2}}))\Delta t^2\,.$$

The computational techniques worked well even for this *nonlinear* ODE!

Truncation errors in vibration ODEs

Linear model without damping

$$u''(t) + \omega^2 u(t) = 0, \quad u(0) = I,\ u'(0) = 0\,. \tag{43}$$

Centered difference approximation:

$$[D_t D_t u + \omega^2 u = 0]^n\,. \tag{44}$$

Truncation error:

$$[D_t D_t u_{\mathrm{e}} + \omega^2 u_{\mathrm{e}} = R]^n\,. \tag{45}$$

Use (17)-(18) to expand $[D_t D_t u_{\mathrm{e}}]^n$:

$$[D_t D_t u_{\mathrm{e}}]^n = u_{\mathrm{e}}''(t_n) + \frac{1}{12} u_{\mathrm{e}}''''(t_n)\Delta t^2,$$

Collect terms: $u_{\mathrm{e}}''(t) + \omega^2 u_{\mathrm{e}}(t) = 0$. Then,

$$R^n = \frac{1}{12} u_{\mathrm{e}}''''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4)\,. \tag{46}$$

Truncation errors in the initial condition

- Initial conditions: $u(0) = I$, $u'(0) = V$
- Need discretization of $u'(0)$
- Standard, centered difference: $[D_{2t}u = V]^0$, $R^0 = \mathcal{O}(\Delta t^2)$
- Simpler, forward difference: $[D_t^+ u = V]^0$, $R^0 = \mathcal{O}(\Delta t)$
- Does the lower order of the forward scheme impact the order of the whole simulation?
- Answer: run experiments!

Computing correction terms

- Can we add terms to the ODE such that the truncation error is improved?

$$[D_t D_t u_e + \omega^2 u_e = C + R]^n,$$

- Idea: choose C^n such that it absorbs the Δt^2 term in R^n,

$$C^n = \frac{1}{12} u_e''''(t_n)\Delta t^2 .$$

- Downside: got a u'''' term
- Remedy: use the ODE $u'' = -\omega^2 u$ to see that $u'''' = \omega^4 u$.
- Just apply the standard scheme to a modified ODE:

$$[D_t D_t u + \omega^2 (1 - \frac{1}{12}\omega^2 \Delta t^2) u = 0]^n,$$

- Accuracy is $\mathcal{O}(\Delta t^4)$.

Model with damping and nonlinearity

Linear damping $\beta u'$, nonlinear spring force $s(u)$, and excitation F:

$$mu'' + \beta u' + s(u) = F(t)\,. \tag{47}$$

Central difference discretization:

$$[mD_tD_tu + \beta D_{2t}u + s(u) = F]^n\,. \tag{48}$$

Truncation error is defined by

$$[mD_tD_tu_{\mathrm{e}} + \beta D_{2t}u_{\mathrm{e}} + s(u_{\mathrm{e}}) = F + R]^n\,. \tag{49}$$

Carrying out the truncation error analysis

Using (17)-(18) and (7)-(8) we get

$$
[mD_tD_tu_e + \beta D_{2t}u_e]^n = mu_e''(t_n) + \beta u_e'(t_n) + \\
\left(\frac{m}{12}u_e''''(t_n) + \frac{\beta}{6}u_e'''(t_n)\right)\Delta t^2 + \mathcal{O}(\Delta t^4)
$$

The terms

$$
mu_e''(t_n) + \beta u_e'(t_n) + \omega^2 u_e(t_n) + s(u_e(t_n)) - F^n,
$$

correspond to the ODE (= zero).
Result: accuracy of $\mathcal{O}(\Delta t^2)$ since

$$
R^n = \left(\frac{m}{12}u_e''''(t_n) + \frac{\beta}{6}u_e'''(t_n)\right)\Delta t^2 + \mathcal{O}(\Delta t^4), \tag{50}
$$

Correction terms: complicated when the ODE has many terms...

Extension to quadratic damping

$$mu'' + \beta|u'|u' + s(u) = F(t)\,. \tag{51}$$

Centered scheme: $|u'|u'$ gives rise to a nonlinearity.
Linearization trick: use a geometric mean,

$$[|u'|u']^n \approx |[u']^{n-\frac{1}{2}}|[u']^{n+\frac{1}{2}}\,.$$

Scheme:

$$[mD_tD_tu]^n + \beta|[D_tu]^{n-\frac{1}{2}}|[D_tu]^{n+\frac{1}{2}} + s(u^n) = F^n\,. \tag{52}$$

The truncation error for quadratic damping (1)

Definition of R^n:

$$[mD_tD_tu_{\rm e}]^n + \beta|[D_tu_{\rm e}]^{n-\frac{1}{2}}|[D_tu_{\rm e}]^{n+\frac{1}{2}} + s(u_{\rm e}^n) - F^n = R^n\,. \quad (53)$$

Truncation error of the geometric mean, see (23)-(24),

$$|[D_tu_{\rm e}]^{n-\frac{1}{2}}|[D_tu_{\rm e}]^{n+\frac{1}{2}} = [|D_tu_{\rm e}|D_tu_{\rm e}]^n - \frac{1}{4}u'(t_n)^2\Delta t^2 + \frac{1}{4}u(t_n)u''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4)\,.$$

Using (5)-(6) for the $D_tu_{\rm e}$ factors results in

$$[|D_tu_{\rm e}|D_tu_{\rm e}]^n = |u_{\rm e}' + \frac{1}{24}u_{\rm e}'''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4)|\times (u_{\rm e}' + \frac{1}{24}u_{\rm e}'''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4))$$

The truncation error for quadratic damping (2)

For simplicity, remove the absolute value. The product becomes

$$[D_t u_e D_t u_e]^n = (u_e'(t_n))^2 + \frac{1}{12} u_e(t_n) u_e'''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4)\,.$$

With

$$m[D_t D_t u_e]^n = m u_e''(t_n) + \frac{m}{12} u_e''''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4),$$

and using $mu'' + \beta(u')^2 + s(u) = F$, we end up with

$$R^n = (\frac{m}{12} u_e''''(t_n) + \frac{\beta}{12} u_e(t_n) u_e'''(t_n))\Delta t^2 + \mathcal{O}(\Delta t^4)\,.$$

Second-order accuracy! Thanks to

- difference approximation with error $\mathcal{O}(\Delta t^2)$
- geometric mean approximation with error $\mathcal{O}(\Delta t^2)$

The general model formulated as first-order ODEs

$$mu'' + \beta|u'|u' + s(u) = F(t)\,. \tag{54}$$

Rewritten as first-order system:

$$u' = v, \tag{55}$$

$$v' = \frac{1}{m}\left(F(t) - \beta|v|v - s(u)\right)\,. \tag{56}$$

To solution methods:

- Forward-backward scheme
- Centered scheme on a staggered mesh

The forward-backward scheme

Forward step for u, backward step for v:

$$[D_t^+ u = v]^n, \tag{57}$$

$$[D_t^- v = \frac{1}{m}(F(t) - \beta|v|v - s(u))]^{n+1} \,. \tag{58}$$

- Note:
  - step u forward with known v in (57)
  - step v forward with known u in (58)
- Problem: $|v|v$ gives nonlinearity $|v^{n+1}|v^{n+1}$.
- Remedy: linearized as $|v^n|v^{n+1}$

$$[D_t^+ u = v]^n, \tag{59}$$

$$[D_t^- v]^{n+1} = \frac{1}{m}(F(t_{n+1}) - \beta|v^n|v^{n+1} - s(u^{n+1}))\,. \tag{60}$$

Truncation error analysis

- Aim (as always): turn difference operators into derivatives + truncation error terms
- One-sided forward/backward differences: error $\mathcal{O}(\Delta t)$
- Linearization of $|v^{n+1}|v^{n+1}$ to $|v^n|v^{n+1}$: error $\mathcal{O}(\Delta t)$
- All errors are $\mathcal{O}(\Delta t)$
- First-order scheme? No!
- "Symmetric" use of the $\mathcal{O}(\Delta t)$ building blocks yields in fact a $\mathcal{O}(\Delta t^2)$ scheme (!)
- Why? See next slide...

A centered scheme on a staggered mesh

Staggered mesh:

- u is computed at mesh points t_n
- v is computed at points $t_{n+\frac{1}{2}}$

Centered differences in (55)-(55):

$$[D_t u = v]^{n-\frac{1}{2}}, \tag{61}$$

$$[D_t v = \frac{1}{m}(F(t) - \beta|v|v - s(u))]^n\,. \tag{62}$$

- Problem: $|v^n|v^n$, because v^n is not computed directly
- Remedy: Geometric mean,

$$|v^n|v^n \approx |v^{n-\frac{1}{2}}|v^{n+\frac{1}{2}}\,.$$

Truncation error analysis (1)

Resulting scheme:

$$[D_t u]^{n-\frac{1}{2}} = v^{n-\frac{1}{2}}, \tag{63}$$

$$[D_t v]^n = \frac{1}{m}(F(t_n) - \beta|v^{n-\frac{1}{2}}|v^{n+\frac{1}{2}} - s(u^n))\,. \tag{64}$$

The truncation error in each equation is found from

$$[D_t u_{\mathrm{e}}]^{n-\frac{1}{2}} = v_{\mathrm{e}}(t_{n-\frac{1}{2}}) + R_u^{n-\frac{1}{2}},$$

$$[D_t v_{\mathrm{e}}]^n = \frac{1}{m}(F(t_n) - \beta|v_{\mathrm{e}}(t_{n-\frac{1}{2}})|v_{\mathrm{e}}(t_{n+\frac{1}{2}}) - s(u^n)) + R_v^n\,.$$

Using (5)-(6) for derivatives and (23)-(24) for the geometric mean:

$$u_{\mathrm{e}}'(t_{n-\frac{1}{2}}) + \frac{1}{24}u_{\mathrm{e}}'''(t_{n-\frac{1}{2}})\Delta t^2 + \mathcal{O}(\Delta t^4) = v_{\mathrm{e}}(t_{n-\frac{1}{2}}) + R_u^{n-\frac{1}{2}},$$

and

$$v_{\mathrm{e}}'(t_n) = \frac{1}{m}(F(t_n) - \beta|v_{\mathrm{e}}(t_n)|v_{\mathrm{e}}(t_n) + \mathcal{O}(\Delta t^2) - s(u^n)) + R_v^n\,.$$

Truncation error analysis (2)

Resulting truncation error is $\mathcal{O}(\Delta t^2)$:

$$R_u^{n-\frac{1}{2}} = \mathcal{O}(\Delta t^2), \quad R_v^n = \mathcal{O}(\Delta t^2)\,.$$

Observation.

Comparing The schemes (63)-(64) and (59)-(60) are equivalent. Therefore, the forward/backward scheme with ad hoc linearization is also $\mathcal{O}(\Delta t^2)$!