

# Study Guide: Truncation Error Analysis

Hans Petter Langtangen<sup>1,2</sup>

<sup>1</sup>Center for Biomedical Computing, Simula Research Laboratory

<sup>2</sup>Department of Informatics, University of Oslo

Nov 29, 2013

## Contents

<b>1</b>	<b>Overview of what truncation errors are</b>	<b>1</b>
1.1	Abstract problem setting . . . . .	1
1.2	Various error measures . . . . .	2
<b>2</b>	<b>Truncation errors in finite difference formulas</b>	<b>2</b>
2.1	Example: The backward difference for $u'(t)$ . . . . .	2
2.2	Taylor series . . . . .	2
2.3	Taylor series inserted in the backward difference approximation . . . . .	3
2.4	The forward difference for $u'(t)$ . . . . .	3
2.5	The central difference for $u'(t)$ (1) . . . . .	3
2.6	The central difference for $u'(t)$ (1) . . . . .	4
2.7	Leading-order error terms in finite differences (1) . . . . .	4
2.8	Leading-order error terms in finite differences (2) . . . . .	4
2.9	Leading-order error terms in mean values (1) . . . . .	5
2.10	Leading-order error terms in mean values (2) . . . . .	5
2.11	Software for computing truncation errors . . . . .	5
2.12	Symbolic computing with difference operators . . . . .	6
<b>3</b>	<b>Truncation errors in exponential decay ODE</b>	<b>6</b>
3.1	Truncation error of the Forward Euler scheme . . . . .	6
3.2	Truncation error of the Crank-Nicolson scheme . . . . .	6
3.3	Test the understanding! . . . . .	7
3.4	Truncation error of the $\theta$ -rule . . . . .	7
3.5	Using symbolic software . . . . .	7
3.6	Empirical verification of the truncation error (1) . . . . .	8
3.7	Empirical verification of the truncation error (2) . . . . .	8
3.8	Empirical verification of the truncation error in the Forward Euler scheme . . . .	8
3.9	Empirical verification of the truncation error in the Forward Euler scheme . . . .	8
3.10	Increasing the accuracy by adding correction terms . . . . .	8
3.11	Lowering the order of the derivative in the correction term . . . . .	9
3.12	With a correction term Forward Euler becomes Crank-Nicolson . . . . .	9
3.13	Correction terms in the Crank-Nicolson scheme (1) . . . . .	10

3.14	Correction terms in the Crank-Nicolson scheme (2)	11
3.15	Extension to variable coefficients	11
3.16	Exact solutions of the finite difference equations	11
3.17	Computing truncation errors in nonlinear problems (1)	12
3.18	Computing truncation errors in nonlinear problems (2)	12
<b>4</b>	<b>Truncation errors in vibration ODEs</b>	<b>12</b>
4.1	Linear model without damping	12
4.2	Truncation errors in the initial condition	13
4.3	Computing correction terms	13
4.4	Model with damping and nonlinearity	13
4.5	Carrying out the truncation error analysis	14
4.6	Extension to quadratic damping	14
4.7	The truncation error for quadratic damping (1)	14
4.8	The truncation error for quadratic damping (2)	15
4.9	The general model formulated as first-order ODEs	15
4.10	The forward-backward scheme	15
4.11	Truncation error analysis	16
4.12	A centered scheme on a staggered mesh	16
4.13	Truncation error analysis (1)	16
4.14	Truncation error analysis (2)	17

## 1 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.

### 1.1 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:  $u^n$ ,  $n = 0, \dots, N_t$
- $u_e$  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

## 1.2 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

## 2 Truncation errors in finite difference formulas

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

Backward difference approximation to  $u'$ :

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

Define the truncation error of this approximation as

$$R^n = [D_t^- u]^n - u'(t_n). \quad (2)$$

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.

### 2.2 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  $\Delta t$  where the lowest power is  $\Delta t^3$
- Small  $\Delta t$ :  $\Delta t \gg \Delta t^3 \gg \Delta t^4$

### 2.3 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:

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

The difference approximation is of *first order* in  $\Delta t$ . It is exact for linear  $u_e$ .

### 2.4 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

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

### 2.5 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:

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

## 2.6 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

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

Note:

- Second-order accuracy since the leading term is  $\Delta t^2$
- Only even powers of  $\Delta t$

## 2.7 Leading-order error terms in finite differences (1)

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

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

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

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

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

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

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

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

## 2.8 Leading-order error terms in finite differences (2)

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

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

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

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

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

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

## 2.9 Leading-order error terms in mean values (1)

Weighted arithmetic mean:

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

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

Standard arithmetic mean:

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

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

## 2.10 Leading-order error terms in mean values (2)

Geometric mean:

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

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

Harmonic mean:

$$[\bar{u}^{t,h}]^n = u^n = \frac{2}{\frac{1}{u^{n-\frac{1}{2}}} + \frac{1}{u^{n+\frac{1}{2}}}} + R^{n+\frac{1}{2}}, \quad (25)$$

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

## 2.11 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](#)<sup>1</sup>.

## 2.12 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  $D_t D_t$ .

## 3 Truncation errors in exponential decay ODE

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

### 3.1 Truncation error of the Forward Euler scheme

The Forward Euler scheme:

$$[D_t^+ u = -au]^n. \quad (27)$$

Definition of the truncation error  $R^n$ :

$$[D_t^+ u_e + au_e = R]^n. \quad (28)$$

From (11)-(12):

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

Inserted in (28):

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

Note:  $u_e'(t_n) + au_e^n = 0$  since  $u_e$  solves the ODE. Then

$$R^n = \frac{1}{2}u_e''(t_n)\Delta t + \mathcal{O}(\Delta t^2). \quad (29)$$

---

<sup>1</sup>[http://tinyurl.com/jvzzcfn/trunc/truncation\\_errors.py](http://tinyurl.com/jvzzcfn/trunc/truncation_errors.py)

### 3.2 Truncation error of the Crank-Nicolson scheme

Crank-Nicolson:

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

Truncation error:

$$[D_t u_e + a\bar{u}_e^t = R]^{n+\frac{1}{2}}. \quad (31)$$

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

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

Inserted in the scheme we get

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

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

### 3.3 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).

### 3.4 Truncation error of the $\theta$ -rule

The  $\theta$ -rule:

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

Truncation error:

$$[\bar{D}_t u_e + a\bar{u}_e^{t,\theta} = R]^{n+\theta}.$$

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

$$\begin{aligned} R^{n+\theta} &= \left( \frac{1}{2} - \theta \right) u_e''(t_{n+\theta}) \Delta t + \frac{1}{2} \theta (1 - \theta) u_e''(t_{n+\theta}) \Delta t^2 + \\ &\quad \frac{1}{2} (\theta^2 - \theta + 3) u_e'''(t_{n+\theta}) \Delta t^2 + \mathcal{O}(\Delta t^3) \end{aligned} \quad (33)$$

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

### 3.5 Using symbolic software

Can use `sympy` and the tools in `truncation_errors.py`<sup>2</sup>:

```
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
```

<sup>2</sup>[http://tinyurl.com/jvzzcfn/trunc/truncation\\_errors.py](http://tinyurl.com/jvzzcfn/trunc/truncation_errors.py)



```
# 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,
}
```

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

### 3.6 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:

$$R^n = [D_t^+ u_e + a u_e]^n. \quad (34)$$

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

### 3.7 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,  $k = 1, 2, \dots$
- Transform  $R^n$  data to the coarsest mesh and estimate  $r$  for each coarse mesh point

See the [text](#)<sup>3</sup> for more details and an implementation.

### 3.8 Empirical verification of the truncation error in the Forward Euler scheme

### 3.9 Empirical verification of the truncation error in the Forward Euler scheme

### 3.10 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  $R \mathcal{O}(\Delta t^2)$ ?

<sup>3</sup><http://tinyurl.com/k3sdbuv/pub/sphinx-trunc/index.html>

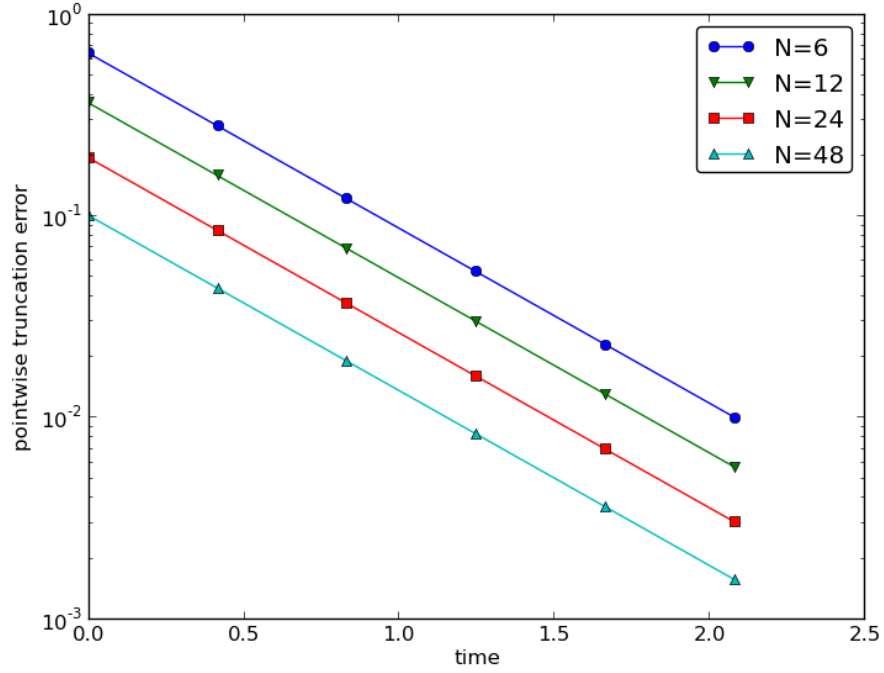


Figure 1: Estimated truncation error at mesh points for different meshes.

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

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

Choosing

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

makes

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

### 3.11 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  $u$

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

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

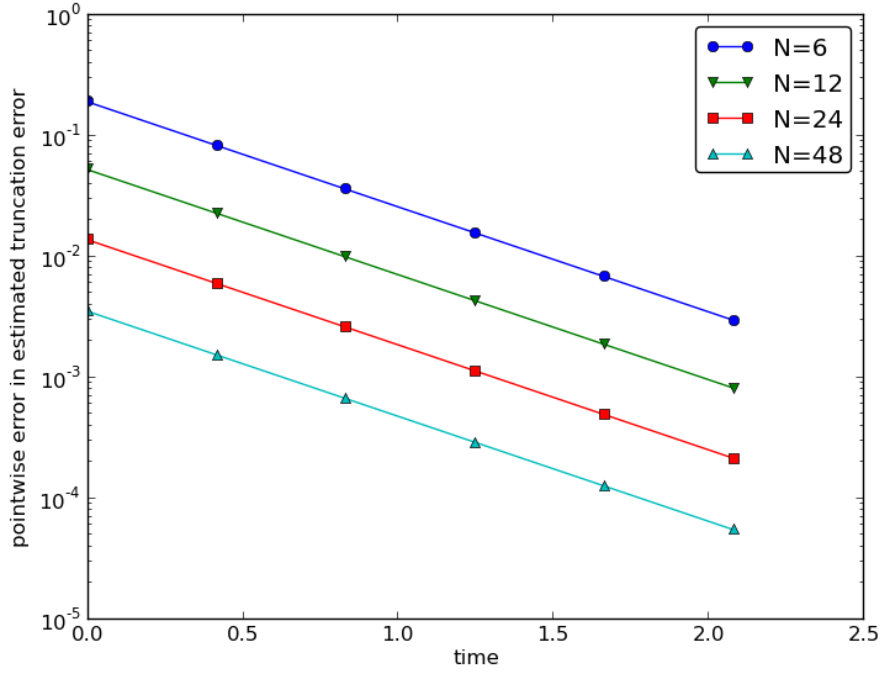


Figure 2: Difference between theoretical and estimated truncation error at mesh points for different meshes.

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

to make a second-order scheme!

### 3.12 With a correction term Forward Euler becomes Crank-Nicolson

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

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

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)!

### 3.13 Correction terms in the Crank-Nicolson scheme (1)

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

Definition of the truncation error  $R$  and correction terms  $C$ :

$$[D_t u_e + a\bar{u}_e^{-t} = C + R]^{n+\frac{1}{2}}.$$

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  $\Delta t^2$  terms:

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

### 3.14 Correction terms in the Crank-Nicolson scheme (2)

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

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

### 3.15 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

$$[D_t^+ u_e + au_e - b = R]^n. \tag{38}$$

Using (11)-(12):

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

Because of the ODE,  $u_e'(t_n) + a(t_n)u_e(t_n) - b(t_n) = 0$ , and

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

No problems with variable coefficients!

### 3.16 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  $u_e$ )

### 3.17 Computing truncation errors in nonlinear problems (1)

$$u' = f(u, t) \quad (40)$$

Crank-Nicolson scheme:

$$[D_t u' = \bar{f}^t]^{n+\frac{1}{2}}. \quad (41)$$

Truncation error:

$$[D_t u_e' - \bar{f}^t = R]^{n+\frac{1}{2}}. \quad (42)$$

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

$$\begin{aligned} [\bar{f}^t]^{n+\frac{1}{2}} &= \frac{1}{2}(f(u_e^n, t_n) + f(u_e^{n+1}, t_{n+1})) \\ &= f(u_e^{n+\frac{1}{2}}, t_{n+\frac{1}{2}}) + \frac{1}{8}u_e''(t_{n+\frac{1}{2}})\Delta t^2 + \mathcal{O}(\Delta t^4). \end{aligned}$$

### 3.18 Computing truncation errors in nonlinear problems (2)

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

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

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

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

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

## 4 Truncation errors in vibration ODEs

### 4.1 Linear model without damping

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

Centered difference approximation:

$$[D_t D_t u + \omega^2 u = 0]^n. \quad (44)$$

Truncation error:

$$[D_t D_t u_e + \omega^2 u_e = R]^n. \quad (45)$$

Use (17)-(18) to expand  $[D_t D_t u_e]^n$ :

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

Collect terms:  $u_e''(t) + \omega^2 u_e(t) = 0$ . Then,

$$R^n = \frac{1}{12} u_e''''(t_n) \Delta t^2 + \mathcal{O}(\Delta t^4). \quad (46)$$

### 4.2 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!

### 4.3 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  $\Delta t^2$  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  $\mathcal{O}(\Delta t^4)$ .

#### 4.4 Model with damping and nonlinearity

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

$$mu'' + \beta u' + s(u) = F(t). \quad (47)$$

Central difference discretization:

$$[mD_t D_t u + \beta D_{2t} u + s(u) = F]^n. \quad (48)$$

Truncation error is defined by

$$[mD_t D_t u_e + \beta D_{2t} u_e + s(u_e) = F + R]^n. \quad (49)$$

#### 4.5 Carrying out the truncation error analysis

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

$$\begin{aligned} [mD_t D_t u_e + \beta D_{2t} u_e]^n &= mu_e''(t_n) + \beta u_e'(t_n) + \\ &\quad \left( \frac{m}{12} u_e''''(t_n) + \frac{\beta}{6} u_e'''(t_n) \right) \Delta t^2 + \mathcal{O}(\Delta t^4) \end{aligned}$$

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

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

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

#### 4.6 Extension to quadratic damping

$$mu'' + \beta |u'|u' + s(u) = F(t). \quad (51)$$

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:

$$[mD_t D_t u]^n + \beta |[D_t u]^{n-\frac{1}{2}}|[D_t u]^{n+\frac{1}{2}} + s(u^n) = F^n. \quad (52)$$

#### 4.7 The truncation error for quadratic damping (1)

Definition of  $R^n$ :

$$[mD_t D_t u_e]^n + \beta |[D_t u_e]^{n-\frac{1}{2}}|[D_t u_e]^{n+\frac{1}{2}} + s(u_e^n) - F^n = R^n. \quad (53)$$

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

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

Using (5)-(6) for the  $D_t u_e$  factors results in

$$[|D_t u_e| D_t u_e]^n = |u'_e + \frac{1}{24} u_e'''(t_n) \Delta t^2 + \mathcal{O}(\Delta t^4)| \times (u'_e + \frac{1}{24} u_e'''(t_n) \Delta t^2 + \mathcal{O}(\Delta t^4))$$

#### 4.8 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  $m u'' + \beta(u')^2 + s(u) = F$ , 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  $\mathcal{O}(\Delta t^2)$

#### 4.9 The general model formulated as first-order ODEs

$$m u'' + \beta |u'| u' + s(u) = F(t). \quad (54)$$

Rewritten as first-order system:

$$u' = v, \quad (55)$$

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

To solution methods:

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



## 4.10 The forward-backward scheme

Forward step for  $u$ , backward step for  $v$ :

$$[D_t^+ u = v]^n, \quad (57)$$

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

- 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  $|v^n|v^{n+1}$

$$[D_t^+ u = v]^n, \quad (59)$$

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

## 4.11 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...

## 4.12 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):

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

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

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

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

### 4.13 Truncation error analysis (1)

Resulting scheme:

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

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

The truncation error in each equation is found from

$$\begin{aligned} [D_t u_e]^{n-\frac{1}{2}} &= v_e(t_{n-\frac{1}{2}}) + R_u^{n-\frac{1}{2}}, \\ [D_t v_e]^n &= \frac{1}{m}(F(t_n) - \beta|v_e(t_{n-\frac{1}{2}})|v_e(t_{n+\frac{1}{2}}) - s(u^n)) + R_v^n. \end{aligned}$$

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

$$u'_e(t_{n-\frac{1}{2}}) + \frac{1}{24}u'''_e(t_{n-\frac{1}{2}})\Delta t^2 + \mathcal{O}(\Delta t^4) = v_e(t_{n-\frac{1}{2}}) + R_u^{n-\frac{1}{2}},$$

and

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

### 4.14 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  $\mathcal{O}(\Delta t^2)$ !

# Index

correction terms, 8

decay ODE, 6

finite differences

backward, 2

centered, 3

forward, 3

truncation error

Backward Euler scheme, 2

correction terms, 8

Crank-Nicolson scheme, 3

Forward Euler scheme, 3

general, 1

table of formulas, 4

verification, 11