

# 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

Sep 15, 2013

## Contents

0.1	Motivation for studying truncation errors . . . . .	1
0.2	Abstract problem setting . . . . .	1
0.3	Truncation error for a differential equation problem . . . . .	2
<b>1</b>	<b>Computing truncation errors in finite difference formulas</b>	<b>2</b>
1.1	Example: The backward difference for $u'(t)$ . . . . .	2
1.2	Taylor series . . . . .	3
1.3	Taylor series inserted in the backward difference approximation . . . . .	3
1.4	The forward difference for $u'(t)$ . . . . .	3
1.5	The central difference for $u'(t)$ (1) . . . . .	4
1.6	The central difference for $u'(t)$ (1) . . . . .	4
1.7	Overview of leading-order error terms in finite difference formulas (1) . . . . .	5
1.8	Overview of leading-order error terms in finite difference formulas (2) . . . . .	5
1.9	Overview of leading-order error terms in averages . . . . .	5
1.10	Software for computing truncation errors . . . . .	6
1.11	Symbolic computing with difference operators . . . . .	6
<b>2</b>	<b>Truncation errors in an ODE for exponential decay</b>	<b>7</b>
2.1	Truncation error of the Forward Euler scheme . . . . .	7
2.2	Truncation error of the Crank-Nicolson scheme . . . . .	7
2.3	Test the understanding! . . . . .	8
2.4	Truncation error of the $\theta$ -rule . . . . .	8
2.5	Using symbolic software . . . . .	8
2.6	Empirical verification of the truncation error (1) . . . . .	9
2.7	Empirical verification of the truncation error (2) . . . . .	9
2.8	Empirical verification of the truncation error in the Forward Euler scheme . . . . .	9
2.9	Increasing the accuracy by adding correction terms . . . . .	9

2.10	Lowering the order of the derivative in the correction term . . . .	10
2.11	With a correction term Forward Euler becomes Crank-Nicolson .	10
2.12	Correction terms in the Crank-Nicolson scheme . . . . .	11
2.13	Extension to variable coefficients . . . . .	12
2.14	Exact solutions of the finite difference equations . . . . .	12
2.15	Computing truncation errors in nonlinear problems (1) . . . . .	13
2.16	Computing truncation errors in nonlinear problems (2) . . . . .	13
<b>3</b>	<b>Truncation errors in vibration ODEs</b>	<b>13</b>
3.1	Linear model without damping . . . . .	13
3.2	Computing correction terms . . . . .	14
3.3	Model with damping and nonlinearity . . . . .	14
3.4	Model with damping and nonlinearity: truncation error analysis	15
3.5	Extension to quadratic damping . . . . .	15
3.6	The truncation error for quadratic damping (1) . . . . .	15
3.7	The truncation error for quadratic damping (2) . . . . .	16
3.8	The general model formulated as first-order ODEs . . . . .	16
3.9	The forward-backward scheme . . . . .	16
3.10	Truncation error analysis . . . . .	17

## 0.1 Motivation for studying truncation errors

- 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.

## 0.2 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
- $u_e$  the exact solution of the differential equation

Then

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

The numerical solution is computed at mesh points:  $u^n$ ,  $n = 0, \dots, N_t$ .

### 0.3 Truncation error for a differential equation problem

- Dream:  $e^n = u_e(t_n) - u^n$
- Impossible, except for very simple problems
- Must find other error measures that are easier to calculate
- 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

## 1 Computing truncation errors in finite difference formulas

### 1.1 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

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

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.

## 1.2 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  $\Delta t$ :  $\Delta t \gg \Delta t^3 \gg \Delta t^4$

## 1.3 Taylor series inserted in the backward difference approximation

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

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  $u_e$ .

## 1.4 The forward difference for $u'(t)$

Forward difference:

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

Define the truncation error:

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

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

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

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

For the central difference approximation,

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

we write

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

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

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

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

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

Collecting terms in  $[u' = D_t u + R]^n$  we find

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

Note:

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

### 1.7 Overview of leading-order error terms in finite difference formulas (1)

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

$$R^n = \frac{1}{24} u'''(t_n) \Delta t^2 + \mathcal{O}(\Delta t^4) \quad (5)$$

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

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

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

$$R^n = \frac{1}{2} u''(t_n) \Delta t + \mathcal{O}(\Delta t^2) \quad (7)$$

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

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

### 1.8 Overview of leading-order error terms in finite difference formulas (2)

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

$$R^{n+\theta} = \frac{1}{2} (1 - 2\theta) u''(t_{n+\theta}) \Delta t - \frac{1}{6} ((1 - \theta)^3 - \theta^3) u'''(t_{n+\theta}) \Delta t^2 + \mathcal{O}(\Delta t^3) \quad (9)$$

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

$$R^n = -\frac{1}{3} u'''(t_n) \Delta t^2 + \mathcal{O}(\Delta t^3) \quad (10)$$

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

$$R^n = \frac{1}{12} u''''(t_n) \Delta t^2 + \mathcal{O}(\Delta t^4) \quad (11)$$

### 1.9 Overview of leading-order error terms in averages

Weighted arithmetic mean:

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

$$R^{n+\theta} = \frac{1}{2} u''(t_{n+\theta}) \Delta t^2 \theta (1 - \theta) + \mathcal{O}(\Delta t^3). \quad (12)$$

Standard arithmetic mean:

$$\begin{aligned} [\bar{u}]^n &= \frac{1}{2}(u^{n-\frac{1}{2}} + u^{n+\frac{1}{2}}) = u(t_n) + R^n, \\ R^n &= \frac{1}{8}u''(t_n)\Delta t^2 + \frac{1}{384}u'''(t_n)\Delta t^4 + \mathcal{O}(\Delta t^6). \end{aligned} \quad (13)$$

Geometric mean:

$$\begin{aligned} u^{n-\frac{1}{2}}u^{n+\frac{1}{2}} &= (u^n)^2 + R^n, \\ R^n &= -\frac{1}{4}u'(t_n)^2\Delta t^2 + \frac{1}{4}u(t_n)u''(t_n)\Delta t^2 + \mathcal{O}(\Delta t^4). \end{aligned} \quad (14)$$

Harmonic mean:

$$\begin{aligned} u^n &= \frac{2}{\frac{1}{u^{n-\frac{1}{2}}} + \frac{1}{u^{n+\frac{1}{2}}}} + R^{n+\frac{1}{2}}, \\ R^n &= -\frac{u'(t_n)^2}{4u(t_n)}\Delta t^2 + \frac{1}{8}u''(t_n)\Delta t^2. \end{aligned} \quad (15)$$

## 1.10 Software for computing truncation errors

Can use `sympy` to automate calculations with Taylor series.

```
>>> 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](http://tinyurl.com/jvzzcfn/trunc/truncation_errors.py)<sup>1</sup>.

## 1.11 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']
```

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

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

## 2 Truncation errors in an ODE for exponential decay

Model:

$$u'(t) = -au(t), \quad u(0) = I. \quad (16)$$

### 2.1 Truncation error of the Forward Euler scheme

The Forward Euler scheme:

$$[D_t^+ u = -au]^n. \quad (17)$$

Definition of the truncation error  $R^n$ :

$$[D_t^+ u_e + au_e = R]^n. \quad (18)$$

From (8):

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

Inserted in (18):

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

Key observation:  $u_e'(t_n) + au_e^n = 0$  since  $u_e$  solves the ODE. The remaining terms constitute the residual:

$$R^n = \frac{1}{2}u_e''(t_n)\Delta t + \mathcal{O}(\Delta t^2). \quad (19)$$

Largest term in  $R^n$  is proportional to  $\Delta t$ , hence a first-order (in  $\Delta t$ ) scheme.

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

Crank-Nicolson:

$$[D_t u = -au]^{n+\frac{1}{2}}, \quad (20)$$

Truncation error:

$$[D_t u_e + a\bar{u}_e^{-t} = R]^{n+\frac{1}{2}}. \quad (21)$$

From (5) and (13):

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



Inserted in the scheme we get

$$R^{n+\frac{1}{2}} = \left( \frac{1}{24} u_e'''(t_{n+\frac{1}{2}}) + \frac{1}{8} u_e''(t_n) \right) \Delta t^2 + \mathcal{O}(\Delta t^4) \quad (22)$$

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

## 2.3 Test the understanding!

Analyze the the truncation error of the Backward Euler scheme.

## 2.4 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 (9) and (12) along with  $u_e'(t_{n+\theta}) + a u_e(t_{n+\theta}) = 0$  to show

$$\begin{aligned} R^{n+\theta} = & \left( \frac{1}{2} - \theta \right) u_e''(t_{n+\theta}) \Delta t + \frac{1}{2} \theta (1 - \theta) u_e''(t_{n+\theta}) \Delta t^2 + \\ & \frac{1}{2} (\theta^2 - \theta + 3) u_e'''(t_{n+\theta}) \Delta t^2 + \mathcal{O}(\Delta t^3) \end{aligned} \quad (23)$$

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

## 2.5 Using symbolic software

Can use `sympy` and the tools in `trunc/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

    # 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)

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

## 2.6 Empirical verification of the truncation error (1)

Ideas:

- Compute  $R^n$  numerically
- Run a sequence of meshes, see how  $R^n$  converges

For the Forward Euler scheme:

$$R^n = [D_t^+ u_e + au_e]^n. \quad (24)$$

Insert correct  $u_e(t)$ , or add source term to fit any choice of  $u_e$ .

## 2.7 Empirical verification of the truncation error (2)

- Assume  $R = C\Delta t^r$
- $R$  varies with  $n$ ;  $C$  and  $r$  will vary with  $n$  - must estimate  $r$  for each mesh point
- Use a sequence of meshes:  $N_t = 2^{-k}N_0$ ,  $k = 1, 2, \dots$
- Transform  $R$  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.

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

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

$$[D_t^+ u_e + au_e = C + R]^n. \quad (25)$$

Taylor expand:

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

Observe: choosing

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

makes

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

---

<sup>3</sup><http://tinyurl.com/k3sdbuv/pub/>

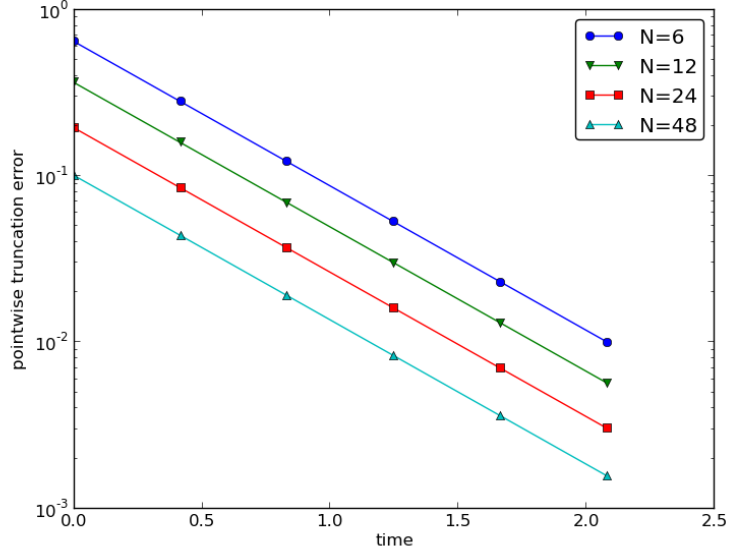


Figure 1: Estimated truncation error at mesh points for different meshes.

## 2.10 Lowering the order of the derivative in the correction term

- $C^n$  contains  $u''$
- Can discretize  $u''$  (one more time level)
- Can also rewrite to  $u'$  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*,

$$u' = -\hat{a}u, \quad \hat{a} = a\left(1 - \frac{1}{2}a\Delta t\right). \quad (26)$$

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

Use  $u'' = -au'$ :

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

Apply Forward Euler:

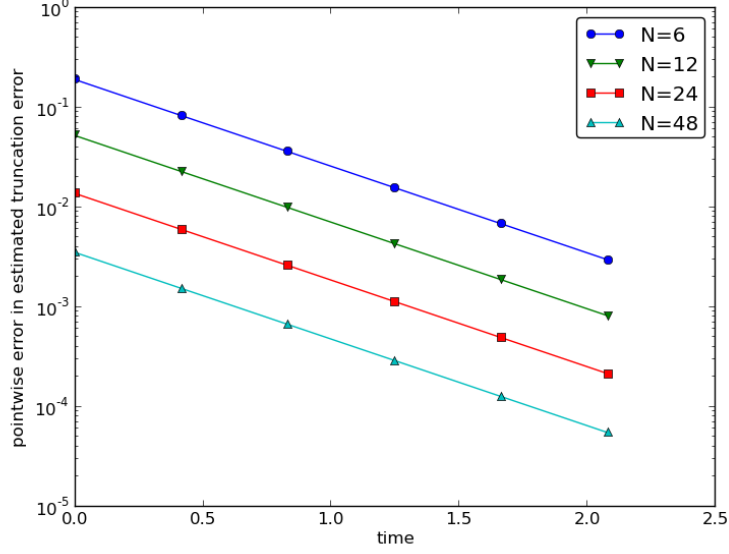


Figure 2: Difference between theoretical and estimated truncation error at mesh points for different meshes.

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

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 for  $u' = -au$ !

## 2.12 Correction terms in the Crank-Nicolson scheme

Standard Crank-Nicolson scheme:

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

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:

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

Using  $u' = -au$ , we have  $u'' = a^2u$  and  $u''' = -a^3u$ .

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

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

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

### 2.13 Extension to variable coefficients

$$u'(t) = -a(t)u(t) + b(t), \quad u(0) = I,$$

Forward Euler:

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

The truncation error fulfills

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

Using (8),

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

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

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

No problems with variable coefficients!

### 2.14 Exact solutions of the finite difference equations

- 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  $u_e$

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

$$u' = f(u, t), \quad (30)$$

Crank-Nicolson scheme:

$$[D_t u' = \bar{f}^t]^{n+\frac{1}{2}}. \quad (31)$$

Truncation error:

$$[D_t u'_e - \bar{f}^t = R]^{n+\frac{1}{2}}. \quad (32)$$

Using (13) for the arithmetic mean:

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

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

With (5), (32) leads to

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

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

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

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

## 3 Truncation errors in vibration ODEs

### 3.1 Linear model without damping

$$u''(t) + \omega^2 u(t) = 0, \quad u(0) = I, \quad u'(0) = 0. \quad (33)$$

Centered difference approximation:

$$[D_t D_t u + \omega^2 u = 0]^n. \quad (34)$$

Truncation error:

$$[D_t D_t u_e + \omega^2 u_e = R]^n. \quad (35)$$

Use (11) 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,

$$R^n = \frac{1}{12} u_e''''(t_n) \Delta t^2 + \mathcal{O}(\Delta t^4). \quad (36)$$

### 3.2 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:  $u''''$

Remedy: use the ODE  $u'' = -\omega^2 u$  to see that  $u'''' = \omega^4 u$ .

$$u'' + \omega^2 \left(1 - \frac{1}{12} \omega^2 \Delta t^2\right) u = 0. \quad (37)$$

Just apply the standard scheme to this modified ODE:

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

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

### 3.3 Model with damping and nonlinearity

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

$$m u'' + \beta u' + s(u) = F(t). \quad (38)$$

Central difference discretization:

$$[m D_t D_t u + \beta D_{2t} u + s(u) = F]^n. \quad (39)$$

Truncation error is defined by

$$[m D_t D_t u_e + \beta D_{2t} u_e + s(u_e) = F + R]^n. \quad (40)$$

### 3.4 Model with damping and nonlinearity: truncation error analysis

Using (11) and (6) we get

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

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

$$R^n = \left( \frac{m}{12} u_e''''(t_n) + \frac{\beta}{6} u_e'''(t_n) \right) \Delta t^2 + \mathcal{O}(\Delta t^4), \quad (41)$$

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

### 3.5 Extension to quadratic damping

$$mu'' + \beta |u'|u' + s(u) = F(t). \quad (42)$$

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:

$$[mD_t D_t u]^n + \beta |[D_t u]^{n-\frac{1}{2}}|[D_t u]^{n+\frac{1}{2}} + s(u^n) = F^n. \quad (43)$$

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

Definition of  $R^n$ :

$$[mD_t D_t u_e]^n + \beta |[D_t u_e]^{n-\frac{1}{2}}|[D_t u_e]^{n+\frac{1}{2}} + s(u_e^n) - F^n = R^n. \quad (44)$$

Truncation error of the geometric mean, see (14),

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

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

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



### 3.7 The truncation error for quadratic damping (2)

For simplicity, remove the absolute value. Computing the product leads to

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

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! But why?

- Difference approximation with truncation error  $\mathcal{O}(\Delta t^2)$
- Geometric mean approximation with truncation error  $\mathcal{O}(\Delta t^2)$

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

$$mu'' + \beta|u'|u' + s(u) = F(t). \quad (45)$$

Rewritten as first-order system:

$$u' = v, \quad (46)$$

$$v' = \frac{1}{m} (F(t) - \beta|v|v - s(u)). \quad (47)$$

To solution methods:

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

### 3.9 The forward-backward scheme

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

$$[D_t^+ u = v]^n, \quad (48)$$

$$[D_t^- v = \frac{1}{m} (F(t) - \beta|v|v - s(u))]^{n+1}. \quad (49)$$

- Note:
  - step  $u$  forward with known  $v$  in (48)
  - step  $v$  forward with known  $u$  in (49)

- Problem:  $|v|v$  gives nonlinearity  $|v^{n+1}|v^{n+1}$ .
- Remedy: linearized as  $|v^n|v^{n+1}$

$$[D_t^+ u = v]^n, \quad (50)$$

$$[D_t^- v]^{n+1} = \frac{1}{m}(F(t_{n+1}) - \beta|v^n|v^{n+1} - s(u^{n+1})). \quad (51)$$

### 3.10 Truncation error analysis

- Aim (as always): turn difference operators into derivatives + truncation error terms
- One-sided forward/backward differences: truncation error  $\mathcal{O}(\Delta t)$
- Linearization of  $|v^{n+1}|v^{n+1}$  to  $|v^n|v^{n+1}$ : truncation 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 (46)-(46):

$$[D_t u = v]^{n-\frac{1}{2}}, \quad (52)$$

$$[D_t v = \frac{1}{m}(F(t) - \beta|v|v - s(u))]^n. \quad (53)$$

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

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

Resulting scheme:

$$[D_t u]^{n-\frac{1}{2}} = v^{n-\frac{1}{2}}, \quad (54)$$

$$[D_t v]^n = \frac{1}{m}(F(t_n) - \beta|v^{n-\frac{1}{2}}|v^{n+\frac{1}{2}} - s(u^n)). \quad (55)$$

**A centered scheme on a staggered mesh: truncation error analysis.**  
The truncation error in each equation fulfills

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

Using (5) for derivatives and (14) 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

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

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 (54)-(55) and (50)-(51) are equivalent. Therefore, the forward/backward scheme with ad hoc linearization is also  $\mathcal{O}(\Delta t^2)$ !

## Index

- correction terms, 9
- decay ODE, 7
- finite differences
  - backward, 2
  - centered, 4
  - forward, 3
- truncation error
  - Backward Euler scheme, 2
  - correction terms, 9
  - Crank-Nicolson scheme, 4
  - Forward Euler scheme, 3
  - general, 1
  - table of formulas, 5
- verification, 12