Appendix A

Numerical Integration

A Numerical Integration

A differential equation of order n

$$y^{(n)}(t) = f[t, y(t), y'(t), ..., y^{(n-1)}(t)] \tag{A.1}$$

can be transformed with the help of the functions

$$\begin{aligned} z_1(t) &= y(t) , \\ z_2(t) &= y'(t) , \\ ... &= ... , \\ z_n(t) &= y^{(n-1)}(t) \end{aligned}$$

to a system of n differential equations of first order:

$$\begin{pmatrix} z_1' \\ z_2' \\ ... \\ z_n' \end{pmatrix} = \begin{pmatrix} z_2 \\ z_3 \\ ... \\ f[t, z_1, z_2, ..., z_n] \end{pmatrix} \quad \rightarrow \quad \boldsymbol{z}' = \boldsymbol{f}[t, z_1, z_2, ..., z_n]. \tag{A.2}$$

To solve this system in the time interval $[t_s, t_e]$ we need the *initial conditions*

$$\{y(t_s), y'(t_s), ..., y^{(n-1)}(t_s)\} \quad \rightarrow \quad \{z_{1_0}, z_{2_0}, ..., z_{(n-1)_0}\} ,$$

which are summarized in the column vector $\boldsymbol{z}_0$.

given: $\boldsymbol{z}_0, t_s, t_e, n, \boldsymbol{f}[t, \boldsymbol{z}]$

time increment: $\triangle t = (t_e - t_s)/n$

loop over all time increments

For i From 0 To n Do

$t_i = t_s + i \triangle t$

$\boldsymbol{z}_{i+1} = \boldsymbol{z}_i + \boldsymbol{f}[t_i, \boldsymbol{z}_i] \triangle t$

End Do

Fig. A.1 Euler's method

D. Gross et al., *Engineering Mechanics 3,*
DOI 10.1007/978-3-642-53712-7,

The numerical solution of (A.2) can be obtained with Euler's method. Its algorithmic procedure is summarized in Fig. A.1.

For the numerical solution of (A.2) with the fourth-order Runge-Kutta method, we have to perform four function evaluations for each of the n first-order differential equations. We assemble these additional function evaluations k_{jl} for $l = 1, 2, 3, 4$ within the column vectors

$$\boldsymbol{k}_1 = \begin{pmatrix} k_{11} \\ k_{21} \\ \dots \\ k_{n1} \end{pmatrix}, \quad \boldsymbol{k}_2 = \begin{pmatrix} k_{12} \\ k_{22} \\ \dots \\ k_{n2} \end{pmatrix}, \quad \boldsymbol{k}_3 = \begin{pmatrix} k_{13} \\ k_{23} \\ \dots \\ k_{n3} \end{pmatrix}, \quad \boldsymbol{k}_4 = \begin{pmatrix} k_{14} \\ k_{24} \\ \dots \\ k_{n4} \end{pmatrix}.$$

A compact program flow of this method is given in Fig. A.2.

given: $\boldsymbol{z}_0, t_s, t_e, n, \boldsymbol{f}[t, \boldsymbol{z}]$

time increment: $\triangle t = (t_e - t_s)/n$

loop over all time increments

FOR i FROM 0 TO n DO

$t_i = t_s + i \, \triangle t$

$\boldsymbol{k}_1 = \boldsymbol{f}[t_i, \boldsymbol{z}_i]$

$\boldsymbol{k}_2 = \boldsymbol{f}[t_i + \frac{\triangle t}{2}, \boldsymbol{z}_i + \boldsymbol{k}_1 \, \frac{\triangle t}{2}]$

$\boldsymbol{k}_3 = \boldsymbol{f}[t_i + \frac{\triangle t}{2}, \boldsymbol{z}_i + \boldsymbol{k}_2 \, \frac{\triangle t}{2}]$

$\boldsymbol{k}_4 = \boldsymbol{f}[t_i + \triangle t, \boldsymbol{z}_i + \boldsymbol{k}_3 \, \triangle t]$

$\boldsymbol{z}_{i+1} = \boldsymbol{z}_i + \frac{\triangle t}{6}(\boldsymbol{k}_1 + 2\, \boldsymbol{k}_2 + 2\, \boldsymbol{k}_3 + \boldsymbol{k}_4)$

END DO

Fig. A.2 Fouth-order Runge-Kutta method

Index

D. Gross et al., *Engineering Mechanics 3*,
DOI 10.1007/978-3-642-53712-7, © Springer-Verlag Berlin Heidelberg 2014