



Numerische Implementierung der nichtlinearen FEM

Aufgabenblatt 2

Allgemeine Hinweise

Die Aufgabenblätter sind als Hausaufgaben in Gruppen zu bearbeiten. Die Abgabe erfolgt per E-Mail an: stefan.hildebrand@tu-berlin.de

Halten Sie Ihre Ergebnisse, Antworten und Erläuterungen für die Aufgaben in einem PDF-Dokument fest und reichen Sie dieses zusammen mit dem Quelltext in einem ZIP-Archiv auf ISIS ein.

Geben Sie auf der ersten Seite des PDF-Dokuments sowie im Quelltext die Namen und Matrikelnummern aller beteiligter Gruppenmitglieder an.

In dieser Lehrveranstaltung können Matlab und GNU Octave verwendet werden.

Quelltext ist zu kommentieren, mindestens sind Eingabe, Ausgabe und Zweck von Funktionen stichpunktartig anzugeben. Wo Algorithmen/ Formeln implementiert werden, ist auf die Quelle zu verweisen (z.B. Nummer der Gleichung im Aufgabenblatt und Name des Algorithmus). Bedeutungen von Variablen/ Parametern müssen eindeutig erkennbar sein, entweder durch gleiche Benennung wie im Aufgabenblatt (z.B. `gamma` für γ) oder durch Kommentare.

Am Ende der Aufgabenstellung finden Sie Materialien mit Formeln und Materialien, die Sie für die Bearbeitung der Aufgabe verwenden und als Referenz heranziehen können.

Implementierte Algorithmen/ Verfahren sind zu validieren. Die Methodik dazu können Sie selbst geeignet wählen. Die Ergebnisse der Validierung ist im PDF-Dokument zusammenzufassen und zu veranschaulichen. Dabei festgestellte Abweichungen sollen begründet und ihre Bedeutung für die Anwendbarkeit des gesamten Codes eingeschätzt werden.

Falls sich im Laufe der Bearbeitung inhaltliche oder organisatorische Schwierigkeiten ergeben, die nicht innerhalb der Gruppe gelöst werden können, findet sich vielleicht eine Lösung in einer individuellen Sprechstunde oder auch per E-Mail an: stefan.hildebrand@tu-berlin.de

Einleitung

Ein großer deutscher Klebstoffhersteller möchte erneut die Leistungsfähigkeit seiner Produkte unter Beweis stellen. Geplant ist, einen beladenen Pick-Up-Truck mit einer Gesamtmasse von 4,1 t mithilfe eines Kranauslegers anzuheben. Der Werbeeffect besteht darin, dass die Zugstange, die den Truck hält, durchgesägt und mit einer Portion Sekundenkleber gefügt wird.

Ihr Auftrag besteht zunächst in der Vorauslegung dieser Werbemaßnahme. Für die sicherheitstechnische Überprüfung und Abnahme wird ein separates Ingenieurbüro beauftragt, die Auslegung der Fügestelle übernimmt der Klebstoffhersteller selbst.¹

Da eine solche Werbeinszenierung selten durchgeführt wird und Erfahrungswerte fehlen, gehen Sie davon aus, dass vom Auftraggeber mehrere Varianten der Berechnung gefordert werden. Um Zeit zu sparen, nutzen Sie daher einen einfachen 2D-Finite-Elemente-Code, in den Sie die Details selbst implementieren.

Den geometrischen Aufbau bestimmen Sie schnell und Sie gelangen zu einem System, das wie in Abb. 1 aufgebaut ist.

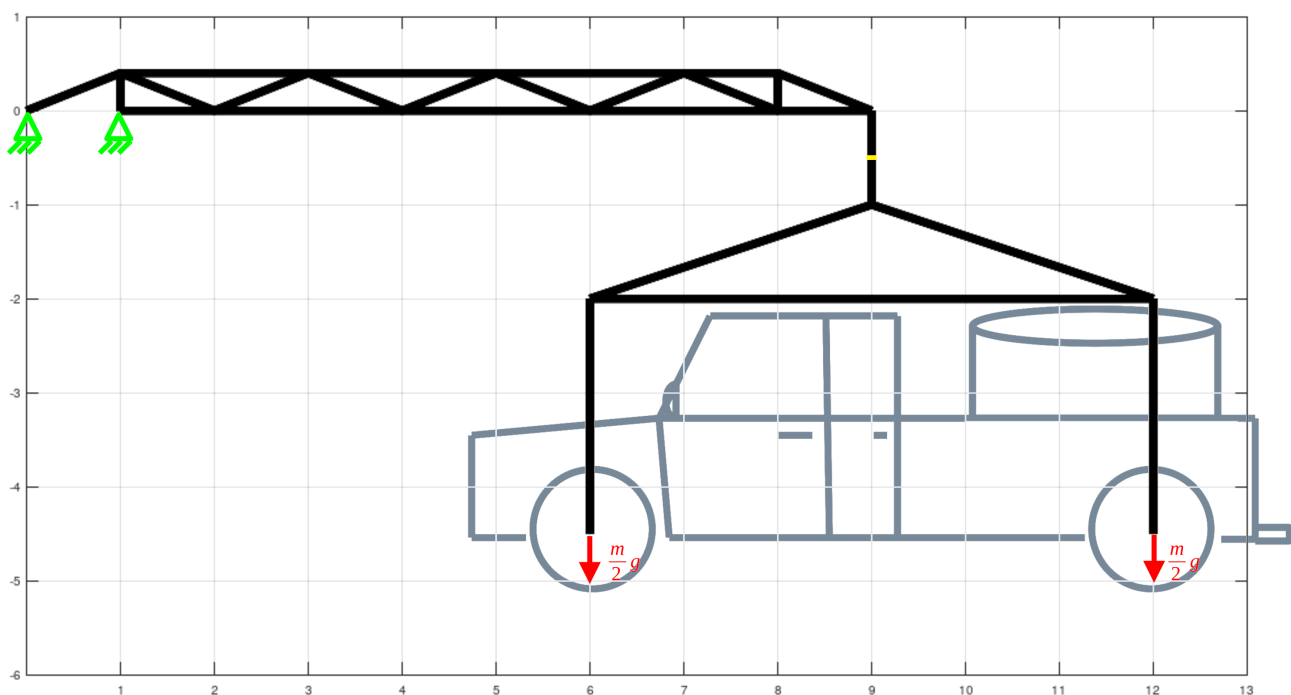


Abbildung 1: Geometrie des Experiments

Da das System in der 3. Dimension symmetrisch und nicht kritisch beansprucht ist, nähern Sie die reale, 3-dimensionale Situation mit einer Simulation in der Ebene an, indem Sie für alle Stäbe die doppelte Querschnittsfläche eingeben.

Ihre Recherche nach Werkstoffen für Mobilkräne ergibt, dass der hochfeste Konstruktionsstahl

¹Bei einem solchen Versuch im Jahre 2007 hatte der Stahlzylinder einen Durchmesser von 7cm. Beim Rekord aus 2019 wurde ein LKW mit 17.5 Tonnen Gesamtmasse angehoben.



S960QL weit verbreitet ist und erhalten von Ihrem Auftraggeber die Freigabe, einen speziellen Ausleger aus diesem Stahl anfertigen zu lassen. Für die Materialparameter finden Sie schnell ein Paper mit experimentellen Ergebnissen (z.B. doi:10.1016/j.proeng.2015.08.044). Sie ermitteln die relevanten Werte aus den dort angegebenen Diagrammen und geben Sie in Ihrer Dokumentation auch explizit an.

Aufgabe 1

Aus einem Ihrer früheren Studienfächer finden Sie einen FE-Code für die Modalanalyse linear-elastischer zweidimensionaler Strukturen. Sie entscheiden sich, den mit dem Fahrzeuggesamtgewicht am Ende belasteten Kran-Arm vereinfachend zunächst als Kragträger zu modellieren und ermitteln die ersten sechs Eigenformen und Eigenfrequenzen dieses Systems.

Aufgabe 2

Um sich den Auftrag des Klebstoffherstellers zu sichern, möchten Sie einen möglichst starken visuellen Eindruck erzeugen. Daher ergänzen Sie Ihren Code, um auch eine transiente Analyse durchführen zu können. Sie nutzen dabei den Newmark-Beta-Algorithmus zur Zeitintegration und simulieren das Verhalten des Kragarms bei plötzlichem Aufbringen des gesamten Fahrzeuggesamtgewichts auf den zuvor nur unverformten Kragarm.

Aufgabe 3

Aus einem Ihrer früheren Studienfächer finden Sie einen FE-Code für statisch belastete, linear-elastisch reagierende Stabwerke.

Um die Werbemaßnahme besonders beeindruckend zu gestalten, beschließen Sie, den Kranarm so auszulegen, dass er gerade plastisch verformt wird, die Zugfestigkeit des Materials aber noch nicht überschritten wird.

Dadurch wird ein Code für komplexere Materialmodelle erforderlich. Das bedeutet, Sie müssen die „residuale Form“ implementieren.

Da Sie noch nicht ganz sicher sind, welche Materialmodelle Sie konkret implementieren werden, ob diese transiente Effekte enthalten oder numerisch ungünstig auf sprunghafte Lastwechsel reagieren, erweitern Sie den Code zudem für quasistatische Berechnungen, d.h. dass zeitliche veränderliche Belastungen ermöglicht werden. Die Lastverläufe werden als Vektoren von Zeiten und Last-Skalierung angegeben und zwischen je zwei Punkten linear interpoliert.

Aufgabe 4

Für die grobe Abschätzung der Dimensionierung beginnen Sie mit dem einfachen Hooke'schen Materialmodell. Sie bestimmen ein E-Modul aus Ihrer Literatur (z.B. aus einem Spannungs-Dehnungs-Diagramm). Weiter schätzen Sie eine Fließgrenze und wählen eine während des Truck-Hebens maximale zu erreichende Spannung aus, die zwischen der Fließgrenze und der



Zugfestigkeit des Materials liegt. Nun ändern Sie den Stabquerschnitt, bis diese Spannung laut Simulation tatsächlich auch erreicht wird. Dazu geben Sie sich in jedem Quasi-Zeitschritt der Simulation den maximalen Betrag der Spannungen im Stabwerk aus.

Aufgabe 5

Nun möchten Sie herausfinden, wie sich die plastische Beanspruchungssituation auswirkt und implementieren dafür ein ideal-plastisches Materialmodell.

Aufgabe 6

Trotz sorgfältiger Validierung konvergiert Ihre FE-Rechnung mit dem ideal-plastischen Materialmodell nicht wie gewünscht. Daher implementieren Sie nun auch lineare Verhärtung und bestimmen aus der Literatur einen Verhärtungsmodul (hardening modulus, H).

Aufgabe 7

Übernächtigt vom Implementieren und Debuggen überlegen Sie sich, den gesamten Ausleger aus Sekundenkleber herzustellen, der in liebevoller Handarbeit zu einem Stabwerk geformt wird.² Sie nehmen an, dass Sie diesen mit einem viskoelastischen Materialmodell modellieren können. Immerhin bietet Ihr Code bereits die Möglichkeit, Zeitabhängigkeiten zu berücksichtigen. Sie finden in vom Wettbewerber gesponserten Vorlesungsmaterialien heraus, dass Sekundenkleber ein E-Modul von 500 bis 700 MPa aufweist. Die dynamische Viskosität nehmen Sie fiktiv als $10^{11} \text{ mPa} \cdot \text{s}$ an. Den Querschnitt der Stäbe passen Sie entsprechend an, um eine ähnliche Auslenkung zu erreichen wie zuvor.

Aufgabe 8

Ihr Auftraggeber ist vom Vorschlag, den gesamten Aufbau aus Kleber herzustellen, beeindruckt, aber doch unsicher, ob Ihre Rechnungen plausibel sind. Sie machen ein Wochenende frei, schlafen sich solide aus und entwerfen dann ein realistischeres Szenario: Der Kranarm wird aus Stahl modelliert, erhält aber zwei zusätzliche viskoelastische Dämpfer an den beiden Lagern, die Sie so gestalten, dass die Schwingungen des Kranarms bei plötzlichem Aufbringen des Fahrzeuggesamtgewichtes wirksam gedämpft werden.

²Dass dabei die Verbindungen der Stäbe starr modelliert werden müssten, übersehen Sie in Ihrer Hektik und verwenden weiter den FE-Code für ein ideales Stabwerk. Falls Ihr Auftraggeber nachfragt, nehmen Sie sich vor, zu behaupten, damit das prüfende Ingenieurbüro testen zu wollen.



Aufgabe 9

Um die Auswertung Ihrer Rechnungen zu untermauern, erstellen Sie einen soliden Vergleich der Materialmodelle. Zudem passen Sie die Belastungshistorie an, sodass nach einer Zeit der maximalen Belastung, in der das System ins statische Gleichgewicht kommt, eine vollständige Entlastung erfolgt. So können Sie u.U. verbleibende plastische Deformationen einschätzen.

Zum Vergleich suchen Sie jeweils nach dem Stab mit der betragsmäßig größten Dehnung und plotten für jedes Materialmodell

- die Spannungs-Dehnungs-Kurve
- die Dehnungs-Zeit-Kurve
- die Spannungs-Zeit-Kurve

und schätzen Ihre Ergebnisse kritisch ein.

Kinematics for 1d trusses in 2d space

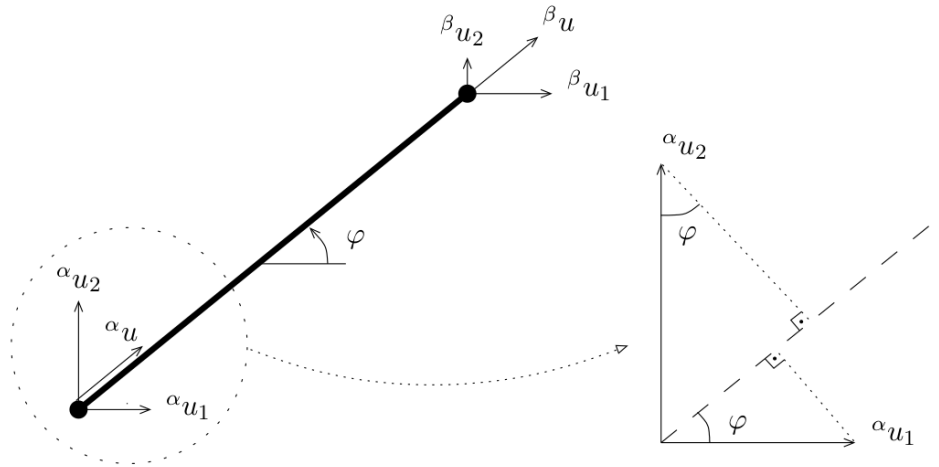


Figure 6.1: 1d truss in 2d-space with respective displacements in the global and local systems

Figure 6.1 shows a 1d truss in a 2d setting. In general, the two degrees of freedom on each node are given and in order to interpolate the field variables within the truss, a transformation onto the local truss axis has to be performed. Figure 6.1 also motivates the transformation equations, which are given as:

$$\begin{bmatrix} \alpha_u \\ \beta_u \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 & 0 \\ 0 & 0 & \cos(\varphi) & \sin(\varphi) \end{bmatrix}}_{=: \mathcal{G}} \cdot \begin{bmatrix} \alpha_{u1} \\ \alpha_{u2} \\ \beta_{u1} \\ \beta_{u2} \end{bmatrix}, \quad (6.1)$$

where the transformation matrix \mathcal{G} can be subdivided into

$$\hat{\mathcal{G}} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 & 0 \end{bmatrix} \quad (6.2)$$

$$\check{\mathcal{G}} = \begin{bmatrix} 0 & 0 & \cos(\varphi) & \sin(\varphi) \end{bmatrix}. \quad (6.3)$$

The trigonometrical quantities can be calculated via the nodal coordinates:

$$\sin(\varphi) = \frac{\beta x_2 - \alpha x_2}{l} \quad \cos(\varphi) = \frac{\beta x_1 - \alpha x_1}{l} \quad (6.4)$$

with the truss-element length

$$l = \sqrt{[\beta x_1 - \alpha x_1]^2 + [\beta x_2 - \alpha x_2]^2}. \quad (6.5)$$

The $[4 \times 4]$ element stiffness matrix can be performed in one equation:

$$\underline{\underline{K}}^e = \sum_{q=1}^{n_{qp}} \left[\left. \frac{\partial \alpha N}{\partial \xi} \right|_{\xi_q} \hat{\mathcal{G}}^e + \left. \frac{\partial \beta N}{\partial \xi} \right|_{\xi_q} \check{\mathcal{G}}^e \right]^t \mathbf{E} \mathbf{A} \left[\left. \frac{\partial \alpha N}{\partial \xi} \right|_{\xi_q} \hat{\mathcal{G}}^e + \left. \frac{\partial \beta N}{\partial \xi} \right|_{\xi_q} \check{\mathcal{G}}^e \right] \frac{\partial \xi}{\partial x} w_q \quad (6.6)$$

The residual format

The residual format changes the overall structure of the FEM code. Until now, the stiffness matrix \underline{K} has been calculated only once and afterwards the linear system of equations has been solved for displacements \underline{u} . As soon as the stiffness matrix is no longer independent from the displacements, the system of equations becomes nonlinear. To solve the nonlinear system of equations, a Newton-Raphson scheme is applied to calculate the displacement-solution iteratively. This iterative scheme has to be performed in each time step.

The structure of a FEM-code in residual format can be seen below.

Preprocessing

Define geometry, boundary conditions, material parameters and load curve

Analysis ^a

for $step = 1$ **to** $nsteps$ **do**

→ load BC's for current timestep

while $|\underline{r}| > tol$ **do**

Newton-Raphson iteration

for $e = 1$ **to** n_{el} **do**

for $q = 1$ **to** n_{qp} **do**

→ calculate stress and algorithmic tangent with material model

→ calculate internal forces

→ calculate entries of the stiffness matrix

→ save values of current quadrature point (stress, strain, internal variables)

end

→ assemble elemental matrices into global matrices

end

→ compute Newton-Raphson update for iterative solution

end

→ save converged results of current timestep

end

Postprocessing

Algorithm 1: Structure of a FEM-code in residual format

^a \underline{r} is the residual, tol the accepted error in the solution, n_{en} the number of element nodes, n_{qp} the number of quadrature points and n_{el} the number of elements.

The residual is the restructured weak form of the balance of linear momentum. The restructuring yields an equation which has to be zero. This residual reads:

$$\underline{r}(\underline{u}_{n+1}) = \underline{f}^{\text{int}}(\underline{u}_{n+1}) - \underbrace{(\underline{f}^{\text{vol}}(\underline{u}_{n+1}) + \underline{f}^{\text{pre}}(\underline{u}_{n+1}))}_{\underline{f}_F^{\text{ext}}} \stackrel{!}{=} \underline{0}. \quad (8.1)$$

For truss structures usually one assumes $f_F^{\text{vol}} = 0$.

The Newton-Raphson scheme starts with an initial guess \underline{u}^0 for the solution-vector and computes the root of a linearization in that point. The root is usually closer to the solution than the initial guess. The update formula is given as

$$\underline{J}^k \cdot \Delta \underline{u}^k = -\underline{r}^k \quad (8.2)$$

with k the counter of the solution iterations and the Jacobian of the residual

$$\underline{J}^k = \frac{\partial \underline{r}^k}{\partial \underline{u}^k} . \quad (8.3)$$

The new solution \underline{u}^{k+1} is then given as:

$$\underline{u}^{k+1} = \underline{u}^k + \Delta \underline{u}^k . \quad (8.4)$$

Based on this new solution, the residual can be calculated again and the update is repeated until the residual is close to zero. In the next load step, the old solution is used as initial guess for the new solution-iteration.

To take the Dirichlet-BC's into account, we decompose this system

$$\begin{bmatrix} \underline{J}_{FF}^k & \underline{J}_{FD}^k \\ \underline{J}_{DF}^k & \underline{J}_{DD}^k \end{bmatrix} \cdot \begin{bmatrix} \Delta \underline{u}_F^k \\ \Delta \underline{u}_D^k \end{bmatrix} = - \begin{bmatrix} \underline{r}_F^k \\ \underline{r}_D^k \end{bmatrix} \quad (8.5)$$

and consider the upper part

$$\underline{J}_{FF}^k \cdot \Delta \underline{u}_F^k = - \left[\underline{r}_F^k + \underline{J}_{FD}^k \cdot \Delta \underline{u}_D^k \right] . \quad (8.6)$$

The solution of (8.6) is a displacement increment and the total displacement is achieved via

$$\underline{u}_F^{k+1} = \underline{u}_F^k + \Delta \underline{u}_F^k . \quad (8.7)$$

The increment $\Delta \underline{u}_D^k$ in (8.6) is only nonzero in the first Newton-Raphson-step and is given by

$$\Delta \underline{u}_D^0 = \underline{u}_D^{(t)} - \underline{u}_D^{(t-\Delta t)} . \quad (8.8)$$

The internal forces within one element are calculated by

$$\underline{f}^{\text{int } e} = \sum_{q=1}^{n_{qp}} \left[\frac{\partial^\alpha N}{\partial \xi} \Big|_{\xi_q} \hat{\mathcal{G}}^e + \frac{\partial^\beta N}{\partial \xi} \Big|_{\xi_q} \check{\mathcal{G}}^e \right]^t \sigma \mathbf{A} w_q . \quad (8.9)$$

Note that often $\partial \underline{r}^k / \partial \underline{u}^k = \partial \underline{f}^{\text{int } k} / \partial \underline{u}^k$ can be assumed, since the volume forces usually are independent of the displacement and prescribed forces are assumed to be independent (neglecting so called follower forces). In comparison of $\underline{f}^{\text{int}}$ and \underline{K} one can see that the Jacobian matrix equals the stiffness matrix in case of linear elasticity. This format can be preserved for other material laws if the algorithmic tangent moduli is used instead of the elastic modulus in the stiffness matrix.

von Mises perfect elastoplasticity (1D)

Overview of the constitutive model

We consider a one-dimensional mechanical device consisting of one spring and one frictional element, arranged as illustrated in Figure 10.1. We let σ and ε denote the total stress and strain applied to the device, the spring constant is denoted by $E > 0$, and the Coulomb frictional element is characterised by the yield stress $\sigma_y > 0$.

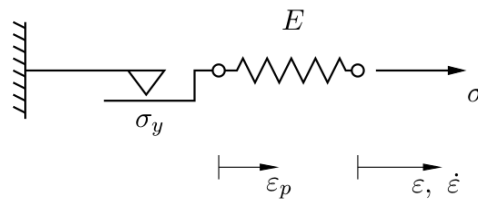


Figure 10.1: One-dimensional rheological model depicting an elastoplastic response.

The rheological model in Figure 10.1 has the following properties:

1. The total strain ε could be additively decomposed into *elastic strain*, ε_e on the spring with constant E , and *plastic strain*, ε_p on the friction device. This could be written down as

$$\varepsilon = \varepsilon_e + \varepsilon_p . \quad (10.1)$$

2. By equilibrium considerations, the stress on the spring with constant E is σ , and we have the elastic relationship

$$\sigma = E \varepsilon_e = E [\varepsilon - \varepsilon_p] . \quad (10.2)$$

The behavior of the friction element is described by the *yield criterion*, which could be written down as

$$f = |\sigma| - \sigma_y \leq 0 , \quad (10.3)$$

where σ_y is the yield stress. If the prescribed stress reaches the yield stress, the friction element is sliding. A higher stress is not possible. The values of the yield criterion can be interpreted as:

- $f < 0$: elastic behaviour
- $f = 0$: plastic behaviour
- $f > 0$: not possible.

The yield stress is considered as the threshold for both, tension and compression, therefore an additional parameter λ for the plastic slip rate is introduced. In compression, the plastic slip rate is negative $\dot{\varepsilon}_p = -\lambda$, and in tension it is positive $\dot{\varepsilon}_p = \lambda$. Combined to one equation this yields the

evolution equation for the plastic strains

$$\dot{\varepsilon}_p = \lambda \operatorname{sign}(\sigma) = \lambda \frac{\sigma^{[n+1]}}{|\sigma^{[n+1]}|} . \quad (10.4)$$

This forms the first equation in the set of *Karush-Kuhn-Tucker* equations to be solved. The other equations to be solved are given as

$$\lambda \geq 0 , \quad f(\sigma, \alpha) \leq 0 \quad \text{and} \quad \lambda f(\sigma, \alpha) = 0 . \quad (10.5)$$

Algorithmic implementation

The simplicity of the von Mises yield criterion Eq.(10.3) enables a closed form of solution (i.e. without an additional local Newton-iteration). This closed form solution is also known as *radial return method*. In the following, we briefly summarise the essential steps of the algorithmic derivation, for details again refer to the lecture notes or the first chapter in the textbook *Computational Inelasticity*, Springer, 1998 by J.C.SIMO AND T.J.R.HUGHES.

To solve the *evolution equation*, Eq.(10.4), we introduce intervals in time, $\Delta t = t^{[n+1]} - t^{[n]}$, and use the *implicit Euler backward integration scheme*. With the quantities at the previous time step known to us, the current plastic strain is calculated as

$$\varepsilon_p^{[n+1]} = \varepsilon_p^{[n]} + \underbrace{\Delta t \lambda^{[n+1]}}_{=:\Delta\lambda} \frac{\sigma^{[n+1]}}{|\sigma^{[n+1]}|} . \quad (10.6)$$

Herein, we have to determine the stress $\sigma^{[n+1]}$ and the consistency parameter $\Delta\lambda^{[n+1]}$. The stress can be obtained from relation

$$\sigma^{[n+1]} = E [\varepsilon^{[n+1]} - \varepsilon_p^{[n+1]}] , \quad (10.7)$$

where, however, $\varepsilon_p^{[n+1]}$ still depends on the unknown consistency parameter $\Delta\lambda^{[n+1]}$.

To determine this parameter, we substitute the euler-backward ansatz Eq. (10.6) into the stress formula (10.7) and get

$$\sigma^{[n+1]} = \underbrace{E [\varepsilon^{[n+1]} - \varepsilon_p^{[n]}]}_{\sigma_{\text{trial}}^{[n+1]}} - E \Delta\lambda \operatorname{sign}(\sigma^{[n+1]}) . \quad (10.8)$$

Within Eq. (10.8), we can identify the so-called “trial”-value of the stress, which can be seen as a first guess based on the old plastic strain¹. This trial state becomes important in the predictor-corrector scheme we will see later on. Decomposing the stress-values according to $\sigma = |\sigma| \operatorname{sign}(\sigma)$

¹This trial state is an auxiliary state, which need not correspond to an actual state, and is obtained by *freezing plastic flow*, i.e. $\dot{\varepsilon}_p = 0$ – a purely elastic (trial) state which can be determined solely in terms of the given quantities $\varepsilon^{[n+1]}$, $\varepsilon_p^{[n]}$ and $\alpha^{[n]}$.



and assuming that the sign of the trial values equals the sign of the real values, i.e. $\text{sign}(\sigma^{[n+1]}) = \text{sign}(\sigma_{\text{trial}}^{[n+1]})$, we obtain an equation for the unknown consistency parameter:

$$|\sigma^{[n+1]}| = |\sigma_{\text{trial}}^{[n+1]}| - E \Delta\lambda^{[n+1]} . \quad (10.9)$$

For an easier implementation we substitute that equation into the flow criterion

$$f^{[n+1]} = \underbrace{|\sigma_{\text{trial}}^{[n+1]}| - \sigma_y}_{f_{\text{trial}}^{[n+1]}} - E \Delta\lambda^{[n+1]} \stackrel{!}{=} 0 \quad (10.10)$$

and get

$$\Delta\lambda^{[n+1]} = \frac{f_{\text{trial}}^{[n+1]}}{E} . \quad (10.11)$$

Algorithmic tangent moduli

For the solution of the boundary value problem, again the *algorithmic consistent tangent moduli* $\mathbf{E}_t^{[n+1]}$ is needed:

$$d\sigma^{[n+1]} = \mathbf{E}_t^{[n+1]} d\varepsilon^{[n+1]}. \quad (10.12)$$

Formally, this follows from differentiation of the algorithmic stress σ_{n+1} with respect to the total deformation ε_{n+1} at t_{n+1} . Using (10.2) the total stress reads

$$\sigma = E [\varepsilon - \varepsilon_p] \quad (10.13)$$

In the current case of plasticity, the tangent is different for elasticity and for plasticity. In an elastic step, the plastic strains are not evolving and therefore does not depend on the total strain. Hence we get

$$\mathbf{E}_t^{[n+1]} = E. \quad (10.14)$$

In a plastic step, the derivation on basis of Eq. (10.8) and (10.9) yields

$$\mathbf{E}_t^{[n+1]} = 0, \quad (10.15)$$

which is in line with the horizontal stress strain curve within the plastic range.

*Note, that the zero-stiffness in the plastic region implies numerical difficulties: **a force driven finite-element code does not work for perfect plasticity**, since there is no corresponding equilibrium point for a prescribed stress value exceeding the yield limit.*

The full algorithmic procedure is summarised in Abb. 2. The structure of the algorithm is a predictor-corrector scheme, where firstly an elastic state is assumed and when this state violates the yield criterion then secondly the correct values are computed.

Algorithm for von Mises perfect plasticity

1. Given: Current total strain at $\varepsilon^{[n+1]}$
and internal variable at the previous equilibrium state $\varepsilon_p^{[n]}$

2. Trial values:

$$\text{Trial stress: } \sigma_{\text{trial}}^{[n+1]} = E [\varepsilon^{[n+1]} - \varepsilon_p^{[n]}]$$

$$\text{Trial yield criterion: } f_{\text{trial}}^{[n+1]} = |\sigma_{\text{trial}}^{[n+1]}| - \sigma_y$$

3. Check yield function

if $f_{\text{trial}}^{[n+1]} \leq 0$ then

$$\sigma^{[n+1]} = \sigma_{\text{trial}}^{[n+1]}$$

$$\varepsilon_p^{[n+1]} = \varepsilon_p^{[n]}$$

$$E_t^{[n+1]} = E$$

else

$$\Delta\lambda^{[n+1]} = \frac{f_{\text{trial}}^{[n+1]}}{E}$$

$$\sigma^{[n+1]} = \sigma_{\text{trial}}^{[n+1]} - \Delta\lambda^{[n+1]} E \frac{\sigma_{\text{trial}}^{[n+1]}}{|\sigma_{\text{trial}}^{[n+1]}|}$$

$$\varepsilon_p^{[n+1]} = \varepsilon_p^{[n]} + \Delta\lambda^{[n+1]} \frac{\sigma_{\text{trial}}^{[n+1]}}{|\sigma_{\text{trial}}^{[n+1]}|}$$

$$E_t^{[n+1]} = 0$$

4. Update the internal variables $\varepsilon_p^{[n]} \leftarrow \varepsilon_p^{[n+1]}$

Abbildung 2: Algorithm 2

Linear hardening

Overview of the constitutive model

The rheological model for plasticity with linear isotropic hardening is given in Figure 11.1.

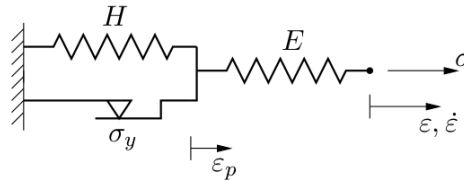


Figure 11.1: One-dimensional rheological model depicting an elastoplastic response.

Still a additive decomposition of strains

$$\varepsilon = \varepsilon_e + \varepsilon_p \quad (11.1)$$

and the elastic relationship

$$\sigma = E \varepsilon_e = E [\varepsilon - \varepsilon_p] \quad (11.2)$$

are valid.

In comparison to perfect plasticity, the yield stress in the plastic region is not constant but increases linearly due to the stiffness of the second spring. Therefore the yield criterion changes to

$$f = |\sigma| - [\sigma_y + H \alpha] \leq 0, \quad \alpha \geq 0. \quad (11.3)$$

H is hardening modulus with $H \geq 0$. The variable α , *internal hardening variable*, is a function of the amount of plastic flow (slip). The model is assumed to obey two conditions, i.e.,

(a.) The hardening is isotropic, and,

(b.) the hardening is linear in the amount of plastic flow (i.e., linear in $|\dot{\varepsilon}_p|$) and independent of $\text{sign}(|\dot{\varepsilon}_p|)$. The evolution equation for the hardening variable could be written as

$$\dot{\alpha} = |\dot{\varepsilon}_p|, \quad (11.4)$$

which is the easiest assumption.

The evolution of plastic strains is still given by

$$\dot{\varepsilon}_p = \lambda \text{sign}(\sigma), \quad (11.5)$$

where $\lambda \geq 0$ is the rate at which slip takes place. This forms the first equation in the set of *Karush-Kuhn-Tucker* equations to be solved. The rest of equations to be solved are given as

$$\lambda \geq 0, \quad f(\sigma, \alpha) \leq 0 \quad \text{and} \quad \lambda f(\sigma, \alpha) = 0. \quad (11.6)$$



Algorithmic implementation

The simplicity of the von Mises yield criterion Eq.(11.3) enables a closed form of solution (i.e. without an additional local Newton-iteration). This closed form solution is also known as *radial return method*. In the following, we briefly summarise the essential steps of the algorithmic derivation. For further details refer to the lecture notes or the first chapter in the textbook *Computational Inelasticity*, Springer, 1998 by J.C.SIMO AND T.J.R.HUGHES.

If the yield criterion is such that $f \leq 0$, we obtain a pure elastic response. However, if $f > 0$, we obtain a plastic response and consequently consider the *evolution equations*, Eq.(11.4) and Eq.(11.5), under consideration of the *Karush-Kuhn-Tucker* conditions which govern the evolution of plastic flow and hardening. To solve these evolution equations, we introduce intervals in time, $\Delta t = t^{[n+1]} - t^{[n]}$, and use the *implicit Euler backward integration scheme*. With the quantities at previous time step known to us, the plastic strains and the hardening variable, both at current time step, is calculated as

$$\varepsilon_p^{[n+1]} = \varepsilon_p^{[n]} + \Delta\lambda^{[n+1]} \frac{\sigma^{[n+1]}}{|\sigma^{[n+1]}|} \quad \text{and} \quad (11.7)$$

$$\alpha^{[n+1]} = \alpha^{[n]} + \Delta\lambda^{[n+1]}. \quad (11.8)$$

Herein, we have to determine the stress $\sigma^{[n+1]}$ and the consistency parameter $\Delta\lambda^{[n+1]}$. The stress can be obtained from relation

$$\sigma^{[n+1]} = E [\varepsilon^{[n+1]} - \varepsilon_p^{[n+1]}], \quad (11.9)$$

where, however, $\varepsilon_p^{[n+1]}$ still depends on the unknown consistency parameter $\Delta\lambda^{[n+1]}$.

To determine this parameter, we substitute the euler-backward ansatz Eq. (11.7) into the stress formula (11.9) and get

$$\sigma^{[n+1]} = \underbrace{E [\varepsilon^{[n+1]} - \varepsilon_p^{[n]}]}_{\sigma_{\text{trial}}^{[n+1]}} - E \Delta\lambda \text{sign}(\sigma^{[n+1]}). \quad (11.10)$$

Within Eq. (11.10), we can identify the so-called “trial”-value of the stress, which can be seen as a first guess based on the old plastic strain¹. This trial state becomes important in the predictor-corrector scheme we will see later on. Decomposing the stress-values according to $\sigma = |\sigma| \text{sign}(\sigma)$ and assuming that the sign of the trial values equals the sign of the real values, i.e. $\text{sign}(\sigma^{[n+1]}) = \text{sign}(\sigma_{\text{trial}}^{[n+1]})$, we obtain an equation for the unknown consistency parameter:

$$|\sigma^{[n+1]}| = |\sigma_{\text{trial}}^{[n+1]}| - E \Delta\lambda^{[n+1]}. \quad (11.11)$$

For an easier implementation we substitute that equation into the flow criterion

$$f^{[n+1]} = |\sigma_{\text{trial}}^{[n+1]}| - [\sigma_y + H\alpha^{[n+1]}] - \Delta\lambda^{[n+1]} E \quad (11.12)$$

¹This trial state is an auxiliary state, which need not correspond to an actual state, and is obtained by *freezing plastic flow*, i.e. $\dot{\varepsilon}_p = 0$ – a purely elastic (trial) state which can be determined solely in terms of the given quantities $\varepsilon^{[n+1]}$, $\varepsilon_p^{[n]}$ and $\alpha^{[n]}$.

Algorithmic tangent moduli

and additionally use Eq. (11.8)

$$f^{[n+1]} = \underbrace{|\sigma_{\text{trial}}^{[n+1]}| - [\sigma_y + H\alpha^{[n]}]}_{=f_{\text{trial}}^{[n+1]}} - \Delta\lambda^{[n+1]} [E + H] \stackrel{!}{=} 0, \quad (11.13)$$

we can finally determine an explicit expression for the consistency parameter $\Delta\lambda^{[n+1]}$ as

$$\Delta\lambda^{[n+1]} = \frac{f_{\text{trial}}^{[n+1]}}{E + H}. \quad (11.14)$$

For the solution of the boundary value problem, again the *algorithmic consistent tangent moduli* $\mathbf{E}_t^{[n+1]}$ is needed:

$$d\sigma^{[n+1]} = \mathbf{E}_t^{[n+1]} d\varepsilon^{[n+1]}. \quad (11.15)$$

Formally, this follows from differentiation of the algorithmic stress σ_{n+1} with respect to the total deformation ε_{n+1} at t_{n+1} . Using (11.10) and (11.14) the total stress can be written as:

$$\sigma^{[n+1]} = E \left[\varepsilon^{[n+1]} - \varepsilon_p^{[n]} - \left\langle \frac{f^{\text{trial}}}{E + H} \right\rangle \text{sign}(\sigma^{[n+1]}) \right] \quad (11.16)$$

The algorithmic tangent then reads

$$\mathbf{E}_t = \frac{d\sigma^{[n+1]}}{d\varepsilon^{[n+1]}} = E \frac{H + E [1 - \mathcal{H}(f^{\text{trial}})]}{E + H}, \quad (11.17)$$

with \mathcal{H} the Heaviside function ($\mathcal{H}(f) = 1$ if $f > 0$ and $\mathcal{H}(f) = 0$ else). The detailed derivation can be taken from the lecture notes.

The full algorithmic procedure is summarised in Abb. 3. The structure of the algorithm is a predictor-corrector scheme, where firstly an elastic state is assumed and when this state violates the yield criterion then secondly the correct values are computed.

Algorithm for von Mises plasticity with linear isotropic hardening

1. Given: Current total strain at $t^{[n+1]}$ and internal variables (plastic strain and hardening variable) at the previous equilibrium state t_n

$$\varepsilon^{[n+1]}, \varepsilon_p^{[n]}, \alpha^{[n]}$$

2. Trial values:

$$\text{Trial stress: } \sigma_{\text{trial}}^{[n+1]} = E [\varepsilon^{[n+1]} - \varepsilon_p^{[n]}]$$

$$\text{Trial yield criterion: } f_{\text{trial}}^{[n+1]} = |\sigma_{\text{trial}}^{[n+1]}| - [\sigma_y + H \alpha^{[n]}]$$

3. Check yield function

if $f_{\text{trial}}^{[n+1]} \leq 0$ then

$$\sigma^{[n+1]} = \sigma_{\text{trial}}^{[n+1]}$$

$$\varepsilon_p^{[n+1]} = \varepsilon_p^{[n]}$$

$$\alpha^{[n+1]} = \alpha^{[n]}$$

$$E_t^{[n+1]} = E$$

else

$$\Delta\lambda^{[n+1]} = \frac{f_{\text{trial}}^{[n+1]}}{E + H}$$

$$\sigma^{[n+1]} = \sigma_{\text{trial}}^{[n+1]} - \Delta\lambda^{[n+1]} E \frac{\sigma_{\text{trial}}^{[n+1]}}{|\sigma_{\text{trial}}^{[n+1]}|}$$

$$\varepsilon_p^{[n+1]} = \varepsilon_p^{[n]} + \Delta\lambda^{[n+1]} \frac{\sigma_{\text{trial}}^{[n+1]}}{|\sigma_{\text{trial}}^{[n+1]}|}$$

$$\alpha^{[n+1]} = \alpha^{[n]} + \Delta\lambda^{[n+1]}$$

$$E_t^{[n+1]} = \frac{E H}{E + H}$$

4. Update the internal variables $\varepsilon_p^{[n]} \leftarrow \varepsilon_p^{[n+1]}$ and $\alpha^{[n]} \leftarrow \alpha^{[n+1]}$

Abbildung 3: Algorithm 3