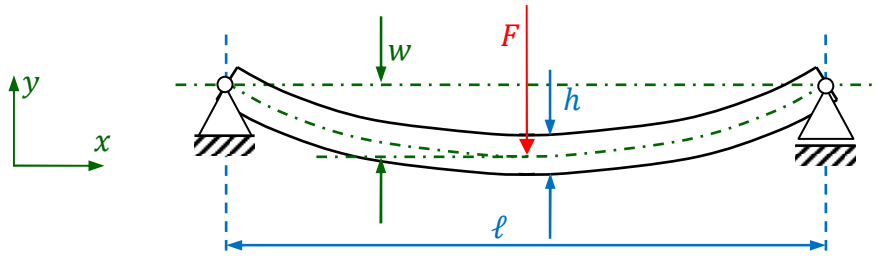
# Übung 4: Drei-Punkt-Balkenbiegung, APDL Skripts in ANSYS Classical

## Teil I: Modell

Wir betrachten einen Balken unter 3-Punkt-Biegung mit einer mittig aufgetragenen Kraft $F$ wie in nachfolgender Grafik veranschaulicht.




Sämtliche relevanten geometrischen und materialspezifischen Angaben sind in folgender Tabelle zusammengefasst:

| | | |
|---|---|---|
| $\ell$ | 2000 mm | Länge des Balkens |
| $h$ | 60 mm | Höhe des Balkens |
| $t$ | 20 mm | Tiefe/Breite/Dicke des Balkens |
| $F$ | 5000 N | Kraft |
| $E$ | 210.000 $\frac{\text{N}}{\text{mm}^2}$ | E-Modul, Stahl |
| $\nu$ | 0,3 | Querkontraktionszahl |

**Aufgaben:**

a. Kann man eventuell Symmetrieeigenschaften nutzen? Zeichne ein vereinfachtes Modell, das die Symmetrie des Balkens berücksichtigt. Wie müssen dann Einspannungen und Kräfte bzw. Momente gewählt werden.
b. Berechne die analytische Abschätzung der maximalen Normalspannung in x-Richtung und der maximalen Auslenkung $w$ (siehe Vorlesung).

## Teil II: APDL Skript erstellen

Wir wollen nun unter Benutzung des vereinfachten Modells die maximale Auslenkung $w$ des Balkens sowie Dehnungen und Spannungen berechnen. Dies werden wir wieder mit ANSYS Classical umsetzen, allerdings dieses Mal mit Hilfe eines APDL-(ANSYS Parametric Design Language)-Skripts.

Starte zuerst, wie in der letzten Übung, ANSYS Classical. Es existiert bereits ein APDL-Programmrumpf, der von der Homepage heruntergeladen werden kann. Fall Ansys über das Netzwerk im Terminal gesttartet wurde bietet sich dazu das Kommando *wget* an, damit das Programm direkt auf dem richtigen Rechner liegt. Aus dem Terminal wechselt man ins Zielverzeichnis und ruft dann den Befehl

*wget http://www.uni-ulm.de/usw.../ blatt4_rumpf.inp*

mit der passenden URL auf. Öffne die Datei in einem geeigneten Texteditor, zum Beispiel *vim, gedit, …*

Nach dem Wechsel in das passende Verzeichnis mittels „File → *Change*“ kann das Programm unter Ansys zum Beispiel durch die Eingabe des Befehls

*/input, blatt4_rumpf.inp*

in der Command Line aufgerufen werden. ANSYS arbeitet dann sämtliche Befehle aus dem Skript ab (Bisher funktioniert das natürlich noch nicht, die wichtigen Teile im Skript fehlen schließlich noch).

APDL-Befehlsaufrufe setzten sich in der Regel aus einem Befehlsnamen mit durch Kommas getrennten, zu übergebenden Parametern zusammen. D.h. sie haben in der Regel die Form

*Befehlsname, Parameter1, Parameter2, …*

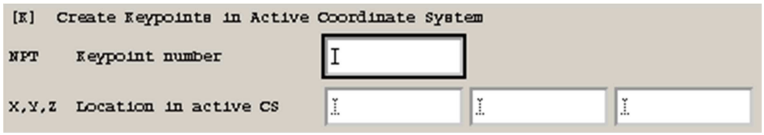
Teilweise sind nicht alle Parameter zu übergeben, dann hat ein Aufruf zum Beispiel die Form

*Befehlsname, Parameter1, , , Parameter4, …*

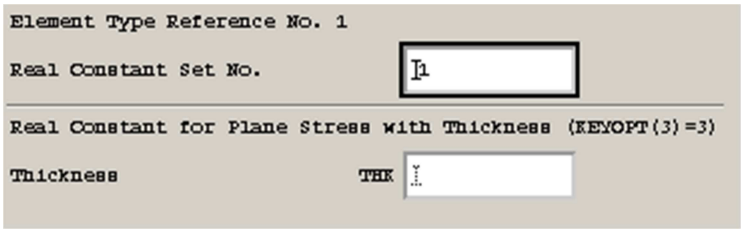
Die APDL-Sprache ist nicht case-sensitive. Manche Befehle beginnen auch mit einem Schrägstrich (/) oder einem Stern (*).

ANSYS besitzt sehr umfangreiche Hilfe-Seiten auf denen die Befehle erklärt sind. Öffnet man ein GUI Dialogsfenster, so gibt es in der Regel eine *Help*-Schaltfläche mit der man den entsprechenden APDL-Befehl finden kann. Alternativ kann man auch in der Command Line mittels „*HELP, command*“ die entsprechende Hilfeseite öffnen, falls man den Befehlsnamen schon kennt oder ahnt. Im geöffneten Hilfefenster kann man auch direkt Suchbegriffe eingeben. Sucht man hier nach bestimten Befehlen, so lohnt es sich oft unter *Options* die Auswahl *Section Title Search* auszuwählen!

***A.1 Geometrie erstellen*:** Wir gehen wie in der letzten Übung vor und erstellen eine Fläche mit Hilfe von mit Linien verbundener Keypoints. Um den APDL-Befehl zum Erstellen von Keypoints herauszufinden gehen wir wieder zu „*Preprocessor → Modeling → Create → Keypoints → In Active CS*“. Im Fenster findet sich in eckigen Klammern bereits der geeignete Befehl *K* und darunter die anzugebenden Parameter. Zur genaueren Beschreibung, öffne die Hilfe und erstelle wie dort beschrieben im APDL-Rumpf die vier Keypoints an den angegebenen Stellen. Dabei wollen wir den Balken so bauen, dass er bei $x = 0$ seine linke Kante hat und die mittlere Linie genau auf der $x$-Achse liegt. Gehe nun genauso vor um die Linien und die Fläche wie in der letzten Übung zu erstellen. Im APDL-Rumpf sind anschließend einige Befehle angegeben, um die erstellten Objekte mit Nummerierung anzuzeigen. Du kannst nun testen, ob du bisher alles richtig gemacht hast. Setze im Skript an der Stelle bis zu der du es ausführen möchtest den Befehl */eof* (= *end of file*) und führe das Programm über die Command Line mit dem Befehl */input, blatt4_rumpf.inp* aus.



***A.2 Vernetzen:*** Zuerst legen wir den Element-Typ auf *plane42* fest. Der passende Befehl ist *ET,1,plane42* und schon im APDL-Rumpf eingetragen. Die 1 ist dabei eine Referenz-Nummer auf diesen Element Typ! In der letzten Übung hatten wir eine 2D-Option gewählt, d.h. einen „ebenen Spannungszustand mit Dicke“. Diese definieren wir über sogenannte Keyoptions. Wir benutzten den Befehl *KEYOPT,1,3,“?“* oder legen diese direkt im *ET* Kommando an. Siehe dazu auch die Ansys Hilfe. Die 1 ist wieder die Referenz-Nummer auf den oben gewählten Element-Typ, die 3 steht für die Nummer der Option. Finde über die Suche nach *plane42* die richtige Nummer für „*?*“.
Die unter A.2.2 aufgeführten Schritte dienen dazu, festzulegen, welche Element Typ Referenzummer, welche Materialnummer und welche Menge an Konstanten die im nachfolgenden Schritt erzeugten Elemente zugewiesen bekommen sollen. Dabei ist nicht wichtig, dass die Konstanten (siehe A.3.1) und das Material (siehe A.3.2) noch nicht definiert wurden, da die Elemente hier zuerst nur die Nummer übergeben bekommen. Die Angabe wäre hier nicht absolut erforderlich, da falls keine Angaben gemacht werden als Default für alle die Nummer 1 genommen würden. Wird aber auf späteren Übungsblättern wichtig.
Wie auf dem letzten Blatt benutzen wir wieder das *SmartSizing* mit Feinheit 4. Versuche selbständig den richtigen Befehl zu finden. Hinweis: wir benötigen zu diesem Befehl nur den ersten Parameter. Die Fläche (mit der Nummer 1, da wir nur eine haben) wird nun mit *AMESH, 1* vernetzt. Teste wieder (mit Hilfe des */eof* Befehls) ob bisher alles funktioniert.




***A.3 Materialeigenschaften:*** Die Dicke des Balkens muss als *Real Constant* angegeben werden. Über den GUI hatten wir diese wie abgebildet für das *Real Set No.1* gesetzt. Dies geschieht nun über den Befehl *R,1,thickness*, d.h. R steht für die Definition einer *Real Constant*, die 1 für das *Set*, bzw. dessen Referenznummer. Wir definieren nun noch E-Modul und Poisson-Zahl (Querkontraktionszahl). Wähle wie in der letzten Übung das Material „*Structural → Linear → Elastic → Isotropic*“ und finde über die Hilfe die passenden Befehle für die beiden Materialangaben.

***A.4 Einspannung und Kräfte:*** Feste Einspannungen werden festgelegt, indem man die Auslenkungen (*Displacement*) an den entsprechenden Punkten auf null setzt. Mit dem Befehl *NSEL,S,LOC* kann man alle zu fixierenden Knoten selektieren, wie im APDL-Rumpf angegeben. Finde über die Hilfe heraus, wofür *S* und *LOC* stehen. Die Auslenkung wird nun mit dem Befehl *D* auf null gesetzt. Der Parameter *ALL* bedeutet dabei, dass man alle zuvor selektierten Knoten fixiert. Mit *NODE(length, 0.0, 0.0)* wählen wir nun den mittleren Knoten am rechten Rand des Balkens aus. Finde den Befehl, wie die Kraft *F* auf diesen Knoten aufgeprägt werden kann. Gerne kannst du nun wieder testen, ob dein Programm bis hierher funktioniert.

***B. Solution:*** Mittels */SOLU* wechseln wir in den Lösungs-Modus. Das ist wichtig! Wollten wir danach weiter etwas verändern, müssten wir wieder mit dem Befehl */PREP7* zurückwechseln Der Analysis Typ *Static* hat den Index 0 und wird als nächstes festgelegt. Mit *SOLV* lösen wir das System. Fertig.

***C. Postprocessor:*** In den Postprocessor wechselt man mit dem Befehl */POST1*. Zunächst setzten wir die Ausgaben mittels */DSCALE,1,1.0* auf *True Scale.* Mit *PLNSOL* und *PLESOL* können *Nodal Solutions* und *Element Solutions* geplottet werden. Beispielhaft werden hier die Auslenkung in $y$-Richtung sowie die Normalspannung in $x$-Richtung ausgegeben. Für was steht die *2* im Aufruf? Wie kann man andere Spannungen plotten und wie Dehnungen?

## Teil III: Konvergenzanalyse

Aus Aufgabenteil b kennen wir die erwartete maximale Normalspannung $\sigma_{xx,max}$. Wir wollen nun analysieren, inwieweit die Anzahl der Elemente auf das ANSYS-Ergebnis Einfluss hat und wie nahe wir an den zuvor berechneten Wert kommen. Da die Vernetzungssteuerung mit SMRTSIZE nur wenig Flexibilität zulässt, wollen wir nun auf eine andere Vernetzungssteuerung zugreifen. Dazu setzen wir unter A.2.2 zuerst *MSHKEY,1*. Der Default-Wert 0 steht für „free meshing“, die 1 für „mapped meshing“. Mit dem Befehl *LESIZE* können wir nun genau festlegen, wie viele Elemente wir in $x$- und $y$-Richtung haben wollen. Dazu wird jeweils eine Linie in die gewünschte Anzahl zerteilt, z.B.:

```
LESIZE, 1,,,nofel_horizontal     ! Horizontal: Element devisions on line 1
LESIZE, 4,,,nofel_vertical       ! Vertical: Element devisions on line 4
```

wobei die jeweiligen letzteren Angaben vorher definierte Variablen sind. Danach wird wieder mit *AMESH,1* vernetzt. Siehe auch die Hilfe zu *LESIZE*.

**Aufgaben:**

c. Löse das Problem einige Male mit unterschiedlich vielen Elementen. Schreibe in eine Tabelle die Ausgabe
   - Gesamtanzahl der Elemente
   - Anzahl Elemente Horizontal
   - Anzahl Elemente Vertikal
   - Maximale Auslenkung $w$
   - Maximale Normalspannung $\sigma_{xx}$ in x-Richtung
   - Maximale Normalspannung $\sigma_{yy}$ in y-Richtung

d. Stelle die Ergebnisse $w$ und $\sigma_{xx}$ in Abhängigkeit der Gesamtanzahl der Elemente grafisch(1) dar und vergleiche mit der analytischen Lösung. Was fällt dabei auf?

e. Prüfe für $\sigma_{yy}$ an welchen Orten (Plural!) die maximalen Werte auftreten. Was für Schlüsse kann man daraus ziehen? Wie könnte man diesen Effekt erklären?

f. Verändere die Einspannung am linken Rand in einer Art und Weise, dass nur noch an einem einzelnen Knoten in $y$-Richtung festgehalten wird (versuche verschiedene Knoten). Erweitere die Tabelle von oben mit den neuen Werten für $\sigma_{xx}$ und $\sigma_{yy}$und stelle auch diese Spalten grafisch dar. Versuche eine Erklärung für den Effekt abzugeben.

(1) Zeichnungen von Hand sind ausreichend. Schöner (und evtl. sogar einfacher) wären allerdings Matlab-Grafiken. Zum starten von Matlab, öffne ein Terminal und gebe *matlab &* ein. Plots können sehr einfach erstellt werden. Lege einfach zwei Vektoren an, z.B. *nofElements = [20 40 60 120]* und *sigma_xx = [10.7 11.5 12.0 12.1 12.2]* und erstelle die Grafik mit *plot(nofElements, sigma_xx).*