

ÜBUNG 7: APDL SKRIPT

INHALT

Ziele.....	2
A. Bedienung von ANSYS Classical	2
A.1 Skriptgliederung	2
B. Unser Lieblingsmodell als APDL Skript.....	3
B.1 Skript erstellen	3
B.2 Netzverfeinerung.....	5
C. Anwendung: Remodeling Algorithm.....	5

ZIELE

Wir erstellen ein Modell in ANSYS durch die programmeigene Skriptsprache. Dafür lernen wir einige Konventionen der Skriptsprache und die relevanten Kommandos kennen.

A. Bedienung von ANSYS Classical

Um ANSYS Classical zu starten wählen wir Mechanical APDL (ANSYS). Das Programm lässt sich auf drei unterschiedliche Arten bedienen.

Zum einen kann man über das **Main Menu** und das **Utility Menu** sämtliche Modell-, Modellierungs- und Simulationseinstellungen vornehmen (ähnlich wie Workbench). Diese Menus sind baumartig aufgebaut, am Ende öffnet sich in der Regel jeweils ein kleines Fenster, in dem man einige Einstellungen vornehmen kann.

Alternativ kann man die **Command Line** verwenden. In den oben erwähnten Einstellungsfenstern steht häufig in [eckigen Klammern] ein Befehl, mit dessen Hilfe man die entsprechenden Einstellungen auch direkt angeben lassen. Die Form ist dabei in der Regel „*COMMAND, input1, input2, ...*“. Steht kein Befehl explizit im Fenster dabei, so kann man diesen in der Regel auch über die Hilfe-Funktion erhalten. Dort erfährt man auch mehr über die erwarteten Input-Argumente. ANSYS ist nicht Case-Sensitiv.

Sämtliche Befehle lassen sich auch zu sogenannten **APDL Scripts** zusammenfassen, in der Regel einfache Textdateien mit Dateiendung „.inp“. Hier lassen sich Befehle zusammenfassen und in der Command Line mittels */INPUT, filename.inp* komplett ausführen.

A.1 Skriptgliederung

Die Schritte, die zur Modellerstellung durchgeführt werden müssen spiegeln sich natürlich in der Skriptgliederung wieder.

1. Preprocessor

Im Preprocessor wird die **Geometrie** durch eine Skizze (analog Sketch in Workbench) erstellt. Das Bauteil wird dann zusätzlich **vernetzt**, und die **Materialeigenschaften zugewiesen**. Als letzten Schritt werden dann noch die **Kräfte und Randbedingen** festgelegt.

2. Solution

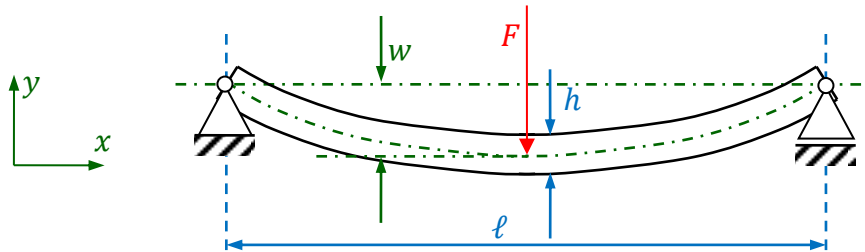
Hier wird der Lösungsmodus angegeben. In unserem Fall ist das ein statisches Modell – in Workbench ist das die Analyseart die als Modell auf die Benutzeroberfläche gezogen wird.

3. Postprocessor

Hier können alle Plots erstellt werden.

B. Unser Lieblingsmodell als APDL Skript

Wir lernen unseren alten Bekannten, den Kragbalken, in einem ganz neuen Licht kennen: diesmal als APDL Skript. Gegeben sei das folgende Modell:



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

ℓ	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
ν	0,3	Querkontraktionszahl

Erinnert euch, das Modell lässt sich unter Symmetrieeigenschaften vereinfachen.

B.1 Skript erstellen

Wir wollen nun unter Benutzung des vereinfachten Modells die maximale Auslenkung w des Balkens sowie Dehnungen und Spannungen mit Hilfe eines APDL-(ANSYS Parametric Design Language)-Skripts berechnen.

Starte zuerst ANSYS Classical. Es existiert bereits ein APDL-Programmrumpf, der von der Homepage heruntergeladen werden kann. 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, ...

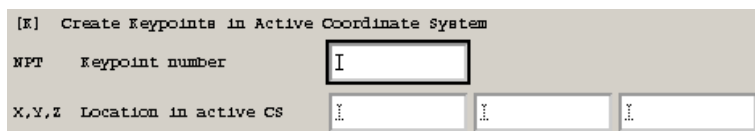
Teilweise sind nicht alle Parameter zu übergeben, dann hat ein Aufruf zum Beispiel die Form

BEFEHLSNAME, parameter1, , , parameter4, ...

Die APDL-Sprache ist nicht case-sensitive. Manche Befehle beginnen auch mit einem Schrägstrich (/) oder einem Stern (*). Generell wird der Befehl im Skript in Großbuchstaben geschrieben, die übergebenen Parameter klein.

ANSYS besitzt sehr umfangreiche Hilfe-Seiten auf denen die Befehle erklärt sind. Öffnet man ein GUI Dialogfenster, 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 bestimmten Befehlen, so lohnt es sich oft unter Options die Auswahl Section Title Search auszuwählen!

A.1 Geometrie erstellen: Wir erstellen eine Fläche mit Hilfe von mit Linien verbundener Keypoints. Um den APDL-Befehl zum Erstellen von Keypoints herauszufinden gehen wir in ANSYS Main menu zu *Preprocessor → Modeling → Create → Keypoints → In Active CS*. Im Fenster findet sich in eckigen Klammern bereits der geeignete Befehl K und darunter die anzugebenden



[K] Create Keypoints in Active Coordinate System			
NPT	Keypoint number	1	
X,Y,Z	Location in active CS		

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. Die weiteren Befehle, die ihr benötigt sind *LSTR* und *AL*. Benutzt die Hilfeseiten um den Syntax der Befehle zu finden. 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! Wir simulieren ein 2D-Modell mit Dicke, d.h. einen „ebenen Spannungszustand mit Dicke“. Diesen definieren wir über sogenannte Keyoptions. Wir benutzen 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 Referenznummer, 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.

Als Netz benutzen wir *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. 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 im ANSYS Main Menu *Preprocessor* → *Material Props* → *Material Models* 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 heraus wie man den Befehl, *F*, für unsere Kraft richtig implementiert. 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 setzen 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?

B.2 Netzverfeinerung

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 divisions on line 1
LESIZE, 4,,,nofel_vertical      ! Vertical: Element divisions on line 4
```

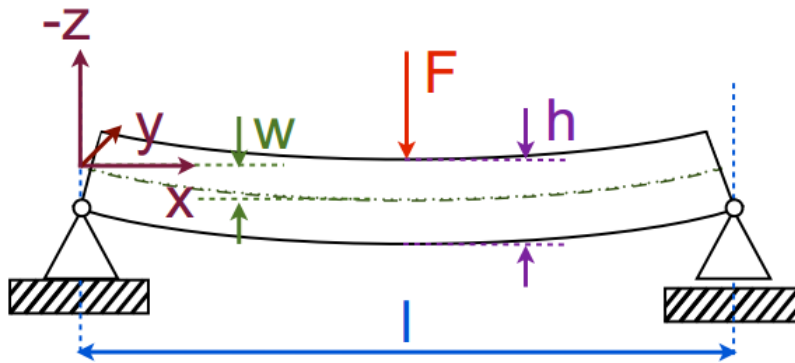
wobei die jeweiligen letzteren Angaben vorher definierte Variablen sind. Danach wird wieder mit *AMESH,1* vernetzt. Siehe auch die Hilfe zu *LESIZE*.

C. Anwendung: Remodeling Algorithm

Der größte Vorteil eines Skripts, abgesehen von einfacher und schneller Änderung des Projekts, ist die Möglichkeit innerhalb des Programms zwischen Preprocessor, Solution und Postprocessor

hin- und herzuwechseln. Das erlaubt uns ein Programm zu schreiben welches iterativ Materialeigenschaften ändert, und so ein Bauteil nach bestimmten Kriterien optimiert.

Gegeben sei, zum Beispiel, ein einfacher Balken, dessen Dicke bei einer vertikal wirkenden Last optimiert werden soll... man kann sich das vorstellen wie eine Brücke. Das Optimierungskriterium, in diesem Fall, ist Strain Energy Density, eine berechnete Größe die für jedes Element die Belastung bezüglich Spannung und Dehnung widerspiegelt. So soll also durch Änderung der Dicke, die SED minimiert werden.



Auf der Homepage findet ihr ein Skript welches diesen Algorithmus implementiert.