

## Teil 2 Berechnungsablauf

### 1 Struktur einer Berechnung

#### 1.1 Berechnungsgang

Die Lösung einer anstehenden Aufgabe mit ANSYS erfordert nicht unbedingt die Entwicklung eines sauberen, leicht lesbaren, leicht nachvollziehbaren, lauffähigen Programms. Sie kann auch mit Hilfe des Menüs abgearbeitet werden. Dabei werden automatisch alle Operationen im `log.file` abgespeichert. Diese Datei ist mit jedem Editor les- und bearbeitbar. Sie enthält aber auch alle Irrwege. Insbesondere die Pick-Operationen mit der Maus sind schwer nachvollziehbar. Aus einem solchen Protokoll lässt sich aber ein lauffähiges Programm machen.

Die Vorteile eines Programms liegen vor allem in der einfachen Möglichkeit, die einmal gelöste Aufgabe mit veränderten Parametern neu automatisch laufen zu lassen. Außerdem ist das Programm die kürzeste und zweckmäßigste Dokumentation des Lösungsweges. Mit ihm kann sich auch ein neuer Bearbeiter schnell in die Aufgabe einarbeiten.

Das zu entwickelnde Programm wird mit einem beliebigen Text-Editor geschrieben. In der Regel geht man von einem vorhanden lauffähigen Programm aus und ändert es für die neue Aufgabenstellung ab. Sequenzen, die mit dem Menü erarbeitet werden, können aus dem `log.file` herauskopiert und in das Programm übernommen werden.

Das zu bearbeitende Programm wird zweckmäßig im aktuellen Verzeichnis (beispielsweise unter dem Namen `programm1.inp`) abgelegt.

#### Programmkopf

Jedes Programm sollte mit einigen Kommentarzeilen beginnen, die den Zweck des Programms, den Namen des Programmierers und das Datum der letzten Änderung enthalten.

```
! Zweck des Programms, zu lösende Aufgabe  
! Name des Programmentwicklers  
! Datum der letzten Änderung
```

Wenn während der Programmentwicklung eine Änderung vorgenommen wird, und deren Auswirkung getestet werden soll, ist es erforderlich, die Datenbasis des vorangegangenen Versuchs zu löschen. Dies erledigen die folgenden Anweisungen:

```
finish  
/clear
```

Um diesen neuen Programmstart möglichst einfach ausführen zu können, sorgt die folgende Anweisung dafür, dass im Fenster ANSYS Toolbar ein mit der Maus zu bedienendes Startfeld angeboten wird.

```
*abbr,programm1,/input,programm1.inp
```

## Vereinbarungsteil

Ein Programm lässt sich leichter überschaubar halten, wenn auffällige Kommentarzeilen eingefügt werden. So kann an dieser Stelle die Zeile

```
!-----Variablenvereinbarung-----
```

den neuen Abschnitt leicht erkennbar gestalten.

An dieser Stelle sollten alle im Programm verwendeten Werte als Variablen bzw. Konstanten vereinbart und ihnen der Startwert zugewiesen werden. Der Zahlenwert eines Parameters der Aufgabenstellung hat im späteren Programmverlauf nichts mehr zu suchen. Nur so kann sichergestellt werden, dass bei einer Parameteränderung kein Fehler unterläuft. (Wenn der Wert an verschiedenen Stellen des Programms benötigt wird, würde man diese mit Sicherheit nicht alle finden.) Der Parameter wird nur über seinen Variablennamen verwendet. Als Variablennamen sind möglichst aussagefähige Kürzel (Beginn mit Alpha-Zeichen, keine verbotenen Zeichen) zu wählen; beispielsweise für die Breite und Höhe eines Rechtecks

```
breit=5      !Breite des Rechtecks
hoch=3       !Höhe des Rechtecks
mw=breit/5   !Maschenweite (für Vernetzung)
```

oder für Kreisinvariante  $\pi$

```
pi=4*atan(1) .
```

Wenn nicht gewichtige Gründe dagegen sprechen, sollten grundsätzlich alle Zahlenwerte für die Größen in SI-Einheiten verwendet werden.

Die Anweisungen sind durch Kommentare ergänzbar.

Dann folgt der Aufruf des Preprozessors und die Vereinbarung der benutzten Elementtypen und die Zuweisung der Materialeigenschaften, bspw.

```
/prep7      !Aufruf des Preprozessors
et,1,plane53 !Freiheitsgrad AZ
et,2,infin110 !Randelement
mp,murx,1,1 !Leiter
mp,murx,2,1 !Luft
```

## Geometriegenerierung

Nach einer weiteren Kommentarzeile kann die Geometriedefinition erfolgen.

```
!----Geometrie----
```

Die Verwendung von Primitiv (Rechtecke, Kreise, Quader, Zylinder,...) für die Geometriedefinition ist meist zweckmäßig. Sie erleichtern auch das Lesen des Programms.

```
rectng,0,breit,0,hoch
```

## Vernetzung

Es folgen dann die Anweisungen für die Vernetzung. Mit den Voreinstellungen kann meist irgendeine Vernetzung der Geometrie erfolgen (**lmesh**, **amesh**, **vmesh**). Für die Anpassung an die Aufgabenstellung können die Anweisungen (**mshape**, **mshkey**, **mshmid**, **esize**, **kesize**, **lesize**, **aesize**) benutzt werden.

Während der Programmentwicklung kann mit einer groben Maschenweite gearbeitet werden. Sie spart Zeit. Läuft das Programm, kann die Maschenweite an die Erfordernisse der Aufgabe angepasst werden.

```
!-----Vernetzung-----
esize,mw          !Elementgröße
amesh,1           !Vernetzung der Fläche 1
```

## Randbedingungen

Auf die vernetzte Geometrie werden die Randbedingungen (Lasten) aufgebracht. Dazu werden an Knoten bestimmte Freiheitsgrade gesetzt oder miteinander gekoppelt (**d**, **cp**, **ce**). Es können auch Kräfte auf Knoten und auf Elemente aufgebracht werden (**f**, **bf**, **bfe**).

Bevor die Lösung angefordert wird, darf die Aufhebung der gemachten Selektionen nicht vergessen werden, wenn tatsächlich alle Elemente und Knoten einbezogen werden sollen

```
allsel
```

## Lösung

Der Lösungsprozessor wird aufgerufen und die Lösung angefordert. Für bestimmte Aufgaben stehen unterschiedliche Löser und Makros zur Verfügung.

```
!-----Lösung-----
/solu          !Aufruf des Lösungsprozessors
solve         !Anforderung der Lösung
```

## Auswertung

Für die Auswertung stehen Postprozessoren (`/post1` , `/post26` ) zur Verfügung. Sie liefern grafische Darstellungen von Feldern und Einzelgrößen über dem Ort oder über der Zeit.

```
!-----Auswertung-----
/post1          !ruft Post-Prozessor auf
plnsol,volt     !stellt das elektrische Potenzial dar
plnsol,ef,sum   !stellt den Betrag der Feldstärke dar
plvect,ef       !stellt Vektoren der Feldstärke dar
```

## Akustisches Signal

Wenn zur Fortschrittskontrolle einer Programmabarbeitung akustische Signale ausgegeben werden sollen, kann die Tonausgabe über das Betriebssystem genutzt werden. Dazu muss eine batch-Datei (z.B. `pieps.bat`) angelegt werden, in der der Tonaufruf steht. Diese kann dann über

```
/sys,pieps,bat
```

aufgerufen werden.

## Hinweise zur Programmabarbeitung

Teile des Programms kann man mit dem Editor in die Zwischenablage kopieren und dann in die Eingabezeile von ANSYS einfügen und dort starten. So kann das Programm für Prüf- und Einarbeitungszwecke zeilen- bzw. abschnittsweise durchgearbeitet werden.

Soll das Programm beim Start für Testzwecke nur bis zu einer bestimmten Stelle laufen, kann das durch die Anweisung

```
/eof      !end of file
```

erreicht werden.

Wenn mehrere Anweisungen in einer Programmzeile angeordnet werden sollen, ist dies das Trennzeichen `$` möglich.

Spare nie mit Kommentar. Dein Nachfolger wird es Dir danken. (Du selbst aber auch nach einem halben Jahr.)

## 1.2 Geometrie-Generierung

## Vieleck

Es soll die Vorgehensweise der Generierung eines Vieleckes mit Hilfe der APDL-Parametersprache beispielhaft beschrieben werden. Eingangsgrößen dafür sind der Radius und die Eckenanzahl. Der Wert für  $\pi$  wird durch eine mathematische Operation zugewiesen.

```
!Generierung eines Vielecks
rad=30e-3      !Radius der Ecken
eck=6         !Eckenanzahl
pi=4*atan(1)
```

Die folgenden Operationen laufen nur im Preprozessor

```
/prep7
```

Für das Vieleck werden zuerst die Eckpunkte und danach die Verbindungslinien zwischen ihnen generiert. Bei der Definition eines Punktes (Keypoint) sollte seine Nummer bekannt sein, um ihn über sie wieder anzusprechen zu können. Wenn innerhalb des Programms schon Punkte vereinbart wurden, kennt man in der Regel ihre Anzahl nicht. Diese Punkte dürfen bei der Neufestlegung auch nicht überschrieben werden. Deshalb fragt man zweckmäßig das System nach der höchsten bisher festgelegten Keypoint-Nummer (alle Keypoints selektiert) und weist diese der Variablen `kpmax` zu. Dieser Wert um 1 erhöht wird `kpstart` zugewiesen. Mit dieser Nummer beginnend können nun die weiteren Punkte definiert werden. Diese Maßnahmen sind notwendig, damit ein solches Programmstück auch innerhalb eines größeren Programms an beliebiger Stelle laufen kann.

```
ksel,all
*get,kpmax,kp,,num,max
kpstart=kpmax+1
```

Die Generierung der Punkte und Linien erfolgt innerhalb einer Laufanweisung. Damit kann auch die variable Eckenanzahl bewältigt werden.

```
*do,i,kpstart,eck+kpstart-1
  phi=2*pi/eck*(i-kpstart)
  xp=rad*cos(phi)
  yp=rad*sin(phi)
  k,i,xp,yp      !Keypoint
```

Nach Festlegung des ersten Punktes kann noch keine Linie definiert werden. Erst ab dem zweiten Punkt kann eine Linie zwischen dem vorigen und dem aktuellen Punkt vereinbart werden.

```
*if,i,gt,kpstart,then
  l,i-1,i      !Linien
*endif
```

Damit kann die Laufanweisung abgeschlossen werden.

```
*enddo
```

Das Vieleck ist jetzt bis auf die letzte Linie fertig. Es wird nun durch eine Linie vom aktuellen Punkt zum Startpunkt geschlossen.

```
l,i,kpstart !letzte Linie
```

Für die im Vereinbarungsteil festgelegten 6 Ecken entsteht das in Abb. 1.2-1 dargestellte Gebilde.

Soll aus den Linien eine Fläche gebildet werden, müssen die beteiligten Linien selektiert und der Anweisung für die Flächengenerierung aus Linien übergeben werden.

```
kselect,s,,,kpstart,kpstart+eck-1  
lselect,s,1  
al,all !Fläche des Vielecks  
lselect,all
```

Danach ist die Selektierung zurückzunehmen.

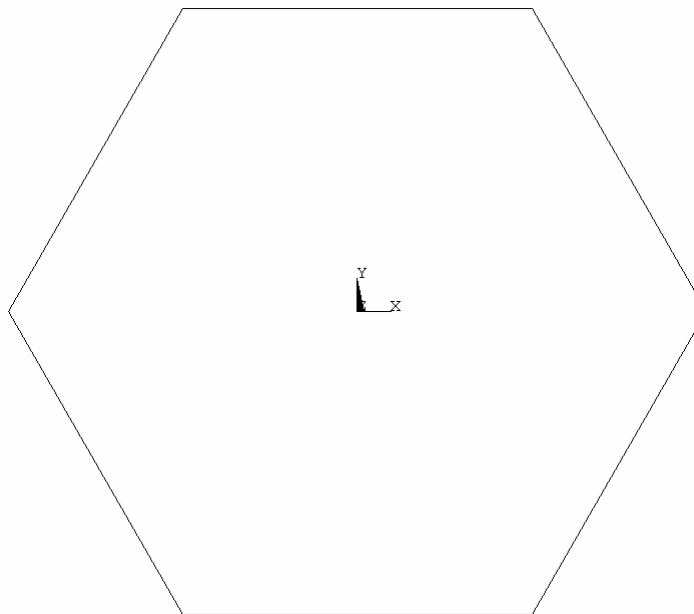


Abb. 1.2-1 Vieleck

### 1.3 Vernetzung für Stromverdrängungsprobleme

Durch den Skin-Effekt bedingt fließen die Ströme in Leitern vorwiegend in oberflächennahen Bereichen. Um Rechenzeit und Speicherplatz zu sparen, kann die Vernetzungsdichte im Leiterinneren wesentlich geringer sein als an der Oberfläche. Die Eindringtiefe  $\delta$  bestimmt die erforderliche Vernetzungsdichte in Oberflächennähe.

$$\delta = \sqrt{\frac{2}{\omega \mu \gamma}}$$

(Für Kupfer beträgt sie bei 10 kHz beispielsweise 0,66 mm.)

Für einen kreisförmigen Leiter mit 10 mm Durchmesser soll eine mögliche Vernetzung gezeigt werden.

### Programm

```
!Vernetzung für Stromverdrängung
!Kreis
!-----Parameter---
rad=5e-3      !Radius des Kreises
```

Die Eindringtiefe kann aus Eingabeparametern errechnet werden oder direkt vorgegeben werden. Die Maschenweite für die Oberflächenschicht wird von der Eindringtiefe abhängig gemacht. (Hier ein Drittel)

```
ed=0.5e-3     !Eindringtiefe
mw=ed/3       !Maschenweite
```

Nach Aufruf des Preprozessors wird ein für solche Aufgabestellungen geeigneter Elementtyp vereinbart.

```
/prep7
et,1,plane53,1  !AZ,Volt
```

Für die angestrebte Vernetzung wird die Geometrie aus mehreren Teilen zusammengesetzt. Die äußere Schicht mit der Dicke der Eindringtiefe und das Leiterinnere werden in jeweils 4 Teile geteilt.

```
!-----Geometrie-----
pcirc,rad,, 0, 90
pcirc,rad,, 90,180
pcirc,rad,,180,270
pcirc,rad,,270,360
pcirc,rad-ed
aovlap,all
!A10,11,12,13: Rand  A6,7,8,9: innen
```

Die entstandenen Flächen zeigt Abb. 1.3-1.

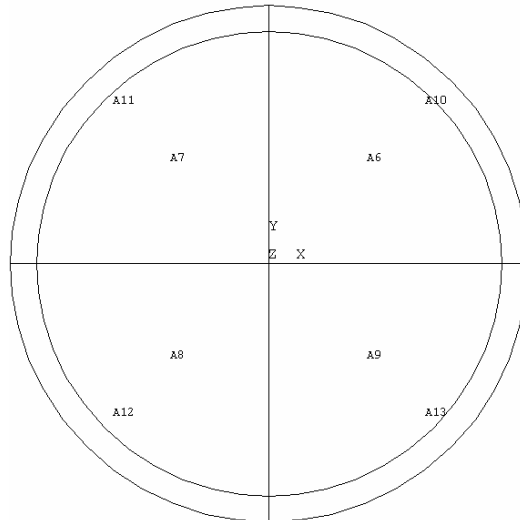


Abb. 1.3-1 generierte Flächen

Zuerst erfolgt die Vernetzung der äußeren Schicht mit der vorgegebenen Maschenweite mit quadratähnlichen Elementen.

```
!-----Vernetzung-----
!-----außen-----
mshape,0,2d !Vierecke
mshkey,1 !mapped meshing
esize,mw
amesh,10,13
```

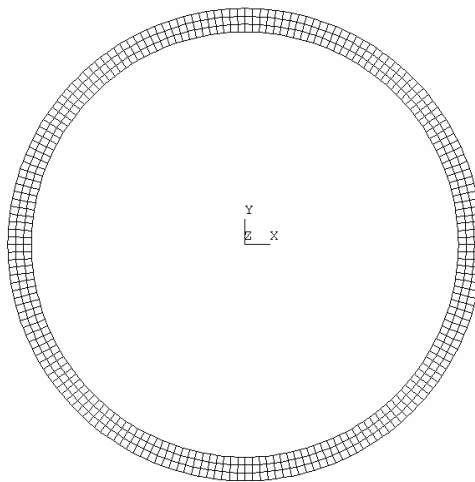


Abb. 1.3-2 Vernetzung der äußeren Schicht

Dann wird auf freie Vernetzung mit dreieckigen Elementen umgeschaltet und die Maschenweite erheblich vergrößert und der innere Teil vernetzt.

```
!-----innen-----
mshape,1,2d !Dreiecke
mshkey,0 !freie Vernetzung
esize,mw*rad/ed
amesh,6,9
```

Abb. 1.3-3 zeigt das Ergebnis.



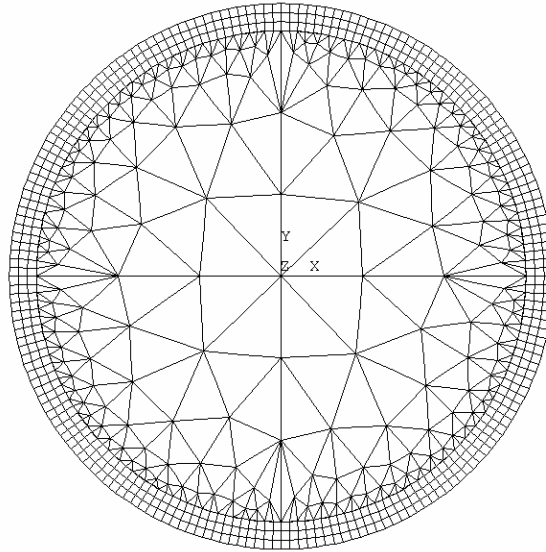


Abb. 1.3-3 vernetzter Querschnitt

### Rechteckiger Leiter

Für einen Leiter mit rechteckigem Querschnitt kann eine ähnliche Vorgehensweise gewählt werden. Hier soll die Nutzung der Symmetrie nach der Vernetzung angewendet werden.

```

!Vernetzung für Stromverdrängung
!Rechteck
!-----Parameter---
breit=12e-3    !Breite des Rechtecks
hoch=8e-3     !Höhe des Rechtecks
ed=0.5e-3     !Eindringtiefe
mw=ed/3       !Maschenweite

/prep7
et,1,plane53,1    !AZ,Volt

!----Geometrie-----
rectng,0,breit/2,0,hoch/2
rectng,0,breit/2-ed,0,hoch/2-ed
aovlap,all
!A4:Rand  A3:innen

```



Abb. 1.3-4 generierte Flächen

```
!-----Vernetzung-----
mshape,0,2d !Vierecke
mshkey,2    !
esize,mw
amesh,4
```

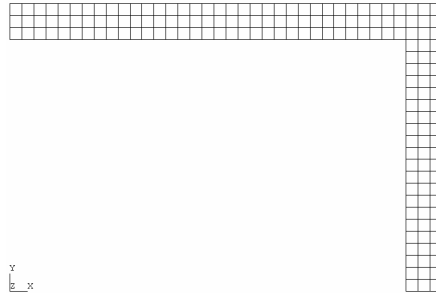


Abb. 1.3-5 Vernetzung der äußeren Schicht

```
!-----
mshape,1,2d !Dreiecke
mshkey,0    !freie Vernetzung
esize,mw*hoch/ed
amesh,3
```

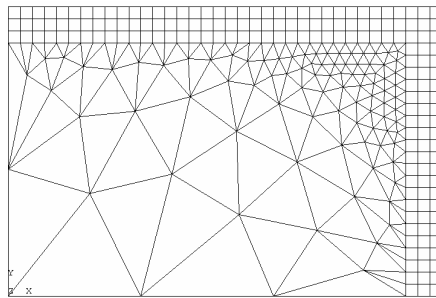


Abb. 1.3-6 Vernetzte Flächen

Nun werden symmetrische Elemente generiert und die dabei entstehenden Dopplungen entfernt.

```
!-----
arsym,x,all !symmetrisch zur Y-Achse
arsym,y,all !symmetrisch zur X-Achse
nummrg,node,mw/10
nummrg,kpoi,mw/10
eplot
```

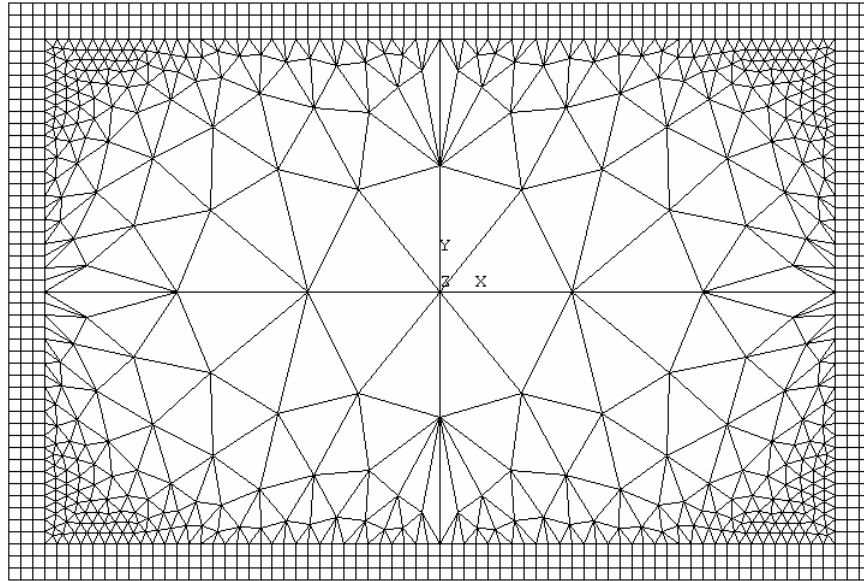


Abb. 1.3-7 Elemente nach Spiegelung