

Darstellungsparameter bearbeiten (Edit Display Parameters):

Damit kann die Sichtbarkeit von Ursprungsmarkierung, Max/Min-Feld und Kontrollpunkten festgelegt werden.

Verknüpfte Teile auflisten (List Associated Parts):

Listet alle Muster auf, die sich im aktuellen Teil und in der Sitzung befinden.

Musterfehler auflisten (List Pattern Errors):

Listet Fehler im Zusammenhang mit Mustern auf, z.B. wenn das im Part vorhandene Muster nicht im *Pattern-Verzeichnis* vorhanden ist.

Musterpunkt erzeugen (Create Pattern Point):

Erzeugt einen *Musterpunkt (Pattern Point)* bei der Erstellung des Musters. Muster können nur an diesen Punkten oder dem Nullpunkt „angefasst“ (selektiert) werden.

2.9 Expressions (Ausdrücke)

Stand: NX6

Tipp:

Ist die Umgebungsvariable UGII_DRAFT_EXPRESSIONS_OK = 1, so kann man auch im Modul Zeichnungserstellung (Drafting) auf Ausdrücke (Expressions) zugreifen.

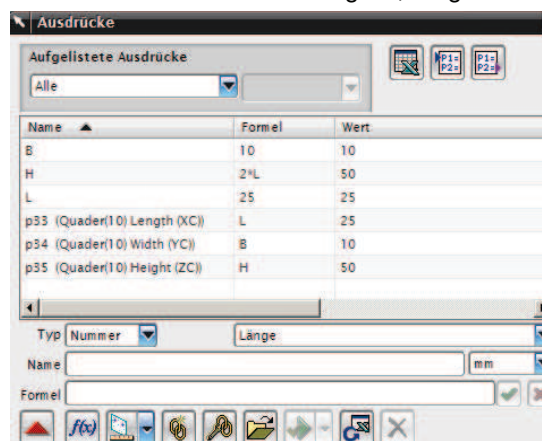
Werkzeug > Ausdrücke
(Tools > Expressions)

2.9.1 Allgemein

In den *Ausdrücken (Expressions)* werden Parameter von Formelementen und bestimmten anderen Objekten abgespeichert.

Es ist auch möglich, eigene Ausdrücke anzulegen, um z.B.

Rechenoperationen durchzuführen, NX als Taschenrechner zu benutzen oder Parameter voneinander abhängig zu machen.



2.9.2 Variablennamen

Variablennamen sind alphanumerische Zeichenketten, die allerdings mit einem Buchstaben beginnen müssen. Innerhalb des Namens sind auch Unterstriche erlaubt („_“).

Um mit Variablen rechnen zu können, müssen diese bereits vorhanden sein. Sie sollten also bereits mit einem Wert belegt worden sein.

Eigene Ausdrücke tauchen auch im *Teile-Navigator (Part Navigator)* unter *Anwenderausdrücke (User Expressions)* auf.

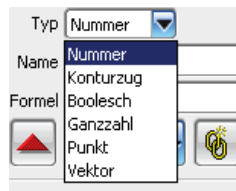
Achtung: Die Namen sind „case sensitive“, d.h., es wird zwischen Groß- und Kleinschreibung unterschieden!

Kommentare können mit „//“ an die Formel angehängt werden und sind bei komplexen Zusammenhängen sehr zu empfehlen.

Achtung!
Groß- und
Kleinschreibung bei
Variablen beachten!

2.9.3 Variablentypen

Seit NX6 sind neue Variablentypen verfügbar. Insbesondere *Punkt*, *Vektor* und *Konturzug (String)* sind hier zu erwähnen (siehe auch Beispiel mit Vektor).



2.9.4 Operatoren

Folgende Rechenoperatoren können genutzt werden:

Operator	Bedeutung
+	Addition
-	Subtraktion
*	Multiplikation
/	Division
%	Modulo
^	Potenz
=	Zuweisung

Vergleichs- und logische Operatoren:

<i>Operator</i>	<i>Bedeutung</i>
>	größer
<	kleiner
>=	größer gleich
<=	kleiner gleich
==	gleich
!=	nicht gleich (ungleich)
!	nicht
& oder &&	logisches UND
 oder 	logisches ODER

Operatoren unterliegen einer bestimmten Abarbeitungsreihenfolge. Die nächste Tabelle stellt dies dar:

<i>Operator</i>	<i>Bedeutung</i>
^	Von rechts nach links
- (negatives Vorzeichen) !	von rechts nach links
* / %	von links nach rechts
> < >= <=	von links nach rechts
== !=	Von links nach rechts
&&	von links nach rechts
 	von links nach rechts
=	von rechts nach links

2.9.5 Bedingungen

Ausdrücke können auch Bedingungen (Fallunterscheidungen) enthalten, die sich beliebig verschachteln lassen. So haben wir vor kurzem eine If-else-Verschachtelung mit 30 Klammern am Ende der Formel realisiert.

Name ▲	Formel	Wert
B	10	10
H	2*L	50
L	25	25
p33 (Quader(10) Length (XC))	L	25
p34 (Quader(10) Width (YC))	if(L<=25)(B)else(2*B)	10
p35 (Quader(10) Height (ZC))	if (L<=25) (H) else (if (L<=50) (2*B) else (110))	50

Hinweis:
Richtig wertvoll werden die Bedingungen erst durch If-else-Verschachtelungen!

2.9.6 Auflistung wichtiger Befehle/Funktionen



Funktionen

Functions

Um die Arbeit mit den Expressions noch effizienter zu gestalten, stehen zusätzlich noch Funktionen in diversen Themengebieten zur Verfügung. Die genauere Dokumentation zu diesen Befehlen ist in der Online-Hilfe im Bereich KF (Knowledge Fusion) zu finden.

Mathematische Funktionen

Befehl/Funktion	Bedeutung, Verwendung	Rückgabotyp
cos ()	Cosinus, cos(Winkel)	Nummer
sin()	Sinus, sin(Winkel)	Nummer
tan()	Tangens, tan(Winkel)	Nummer
hypcos()	hyperbolischer Cosinus, hypcos(Winkel)	Nummer
hypsin()	hyperbolischer Sinus, hypsin(Winkel)	Nummer
hyptan()	hyperbolischer Tangens; hyptan(Winkel)	Nummer
arccos()	Arcus Cosinus im Bogenmaß, arccos(Winkel)	Nummer
arcsin()	Arcus Sinus im Bogenmaß, arcsin(Winkel)	Nummer
arctan()	Arcus Tangens im Bogenmaß, arctan(Winkel)	Nummer
arctan2()	Arcus Tangens im Bogenmaß, Rückgabewerte zwischen -180° und 180° (-Pi bis +Pi) arctan(Winkel1, Winkel2)	Nummer
NormalizeAngle()	Winkel, die größer als 360° sind, werden in den Bereich von 0-360° normiert. NormalizeAngle(Winkel)	Nummer
Radians()	wandelt einen Winkel in Grad in einen Winkel in Radiant um, Radians(WinkelGrad)	Nummer
ceiling()	Aufrunden auf den nächsten Ganzzahlwert, ceiling(x)	Integer
floor()	Abrunden auf den nächsten Ganzzahlwert, floor(x)	Integer
round()	rundet eine Zahl, round(x)	Nummer
log()	natürlicher Logarithmus, log(x)	Nummer
log10()	Zehner-Logarithmus, log10(x)	Nummer
max()	maximaler Wert aus mehreren Zahlen, max(Zahl1, Zahl2, Zahl3)	Nummer
minimum()	minimaler Wert aus mehreren Zahlen, minimum(Zahl1, Zahl2, Zahl3)	Nummer
abs()	Absolutwert, Betrag (Zahl), abs(Zahl)	Nummer
mod()	Modulo-Funktion, mod(Zähler, Nenner)	Nummer
pi()	gibt den Wert von Pi zurück, pi()	Nummer
sqrt()	Quadratwurzel, sqrt(x)	Nummer
RotateVector()	rotiert einen Vektor um eine Achse, RotateVector(rotierender Vekt. , Winkel , Rot.-Achse) RotateVector(Richtung , 45 , Vector(1,0,0))	Vektor

Funktionen zur String-Verarbeitung:

Befehl/Funktion	Bedeutung, Verwendung	Rückgabotyp
ASCII()	gibt den ASCII-Wert des ersten Buchstaben eines Strings zurück, ASCII(„Automatik“) ergibt 65	Integer
Char()	gibt das zugehörige Zeichen eines ASCII-Wertes zurück, Char(65) ergibt A	String
charReplace()	tauscht einzelne Zeichen in Strings aus, charReplace(„NX5“, „5“, „6“) ergibt NX6	String
compareString()	vergleicht Strings miteinander, compareString(„NX5“, NX5“) ergibt 0	Integer
format()	gibt einen formatierten String zurück, format(Formatoptionen, String) format(„%6.2f“, 12.1111) ergibt 12.11 Genauere Informationen sind in der UG/NX-Online-Hilfe zu finden.	String
MakeNumber()	wandelt eine numerische Zeichenfolge in eine Nummer, MakeNumber(String)	Nummer
replaceString()	ersetzt einen Teilstring durch einen anderen, replaceString(„UG/NX“, „UG“, „UNIGRAPHICS“) ergibt UNIGRAPHICS/NX	String
Stringlower()	wandelt alle Großbuchstaben in einem String in Kleinbuchstaben um, Stringlower(„HBB“) ergibt „hbb“	String
Stringupper()	wandelt alle Kleinbuchstaben in einem String in Großbuchstaben um, Stringupper(„hbb“) ergibt „HBB“	String
substring()	schneidet aus einem String einen Teilstring heraus und gibt diesen zurück, substring(„HBB Engineering GmbH“, 1,3) ergibt HBB	String

Zugriff auf externe Daten:

Befehl/Funktion	Bedeutung, Verwendung	Rückgabotyp
ug_cell_hlookup()	sucht in einer Excel-Tabelle nach einer bestimmten Spalte und gibt den Wert einer angegebenen Zeile zurück. z.B. ug_cell_hlookup(„C:\Daten\test.xls“, „Variante“, „A1:E11“, 5, 0) ug_cell_hlookup(Excel-Datei, Suchbegriff, Bereich, Zeile, Modus)	Any
ug_cell_vlookup()	sucht in einer Excel-Tabelle nach einer bestimmten Zeile und gibt den Wert einer angegebenen Spalte zurück. Siehe ug_cell_hlookup()	Any
ug_cell_read()	Liest den Wert einer Zelle in einer Excel-Tabelle aus, ug_cell_read(„C:\Daten\test.xls“, „A1“)	Any
ug_excel_read()	liest den Wert einer Zelle aus einer Excel-Tabelle aus, ug_cell_read(„C:\Daten\test.xls“, „A1“)	Nummer

Befehl/Funktion	Bedeutung, Verwendung	Rückgabotyp
dateTimeString()	gibt das Datum und die Uhrzeit aus, dateTimeString("localTime?", True) oder dateTimeString("localTime?", False) für GMT	String
getenv()	liest den Wert einer Umgebungsvariablen aus, getenv("UGII_PATDIR")	String
stringValue()	wandelt einen Wert in einen String um, stringValue(x)	String
ug_dateTimeDiff()	gibt die Differenz von Datum und Uhrzeit aus, ug_dateTimeDiff(Datum1, Datum2) Datum1/2 ist z.B. „Wed Jun 03 08:36:09 2009“ Ergebnis z.B. „0 days 1 hours 59 minutes ...“	String

Zugriff auf Teileattribute (Part-Attribute):

Befehl/Funktion	Bedeutung, Verwendung	Rückgabotyp
ug_askPartAttrValue()	fragt den Wert eines Teileattributs ab, z.B. ug_askPartAttrValue(„Attributstitel“) (Attribut muss beim Anlegen vorhanden sein!)	String
ug_setPartAttrValue()	setzt den Wert eines Teileattributs z.B. ug_setPartAttrValue(„ATTRIBUT“, „WERT“)	Ganzzahl

Zur Umrechnung von Einheiten stehen folgende Funktionen zur Verfügung:

Befehl/Funktion	Bedeutung, Verwendung
cm	cm(x) konvertiert x von Zentimetern in die Standardeinheit des Parts
ft	ft(x) konvertiert x von Fuß in die Standardeinheit des Parts
grd	grd(x) konvertiert x von Gradient in Grad
inch	inch(x) konvertiert x von Zoll in die Standardeinheit des Parts
km	km(x) konvertiert x von Kilometern in die Standardeinheit des Parts
mc	mc(x) konvertiert x von Mikrometern in die Standardeinheit des Parts
minute	minute(x) konvertiert x von Minuten in Grad
ml	ml(x) konvertiert x von tausendstel Zoll in die Standardeinheit des Parts
mm	mm(x) konvertiert x von Millimetern in die Standardeinheit des Parts
mtr	mtr(x) konvertiert x von Metern in die Standardeinheit des Parts
sec	sec(x) konvertiert x von Sekunden in Grad
yd	yd(x) konvertiert x von Yards in die Standardeinheit des Parts

Sonstige Befehle:

Der Vollständigkeit halber weisen wir darauf hin, dass es noch einige Funktionen in speziellen Themengebieten gibt, die eine spezielle Erläuterung benötigen und den Umfang dieses Buches sprengen würden.



Funktionen
Functions

Hierzu gehören: *beams* (Krafteinwirkung auf Balken), *fluid* (Druckberechnung in Rohren), *gears* (Getriebeberechnungen), *materials* (Materialberechnungen), *mechanics* (mechanische Berechnungen, z.B. Zentrifugalkraft), *o_rings* (Berechnung zu O-Ringen), *plate* (Blechberechnung), *spring* (Federberechnung), *units* (Masseberechnung mit Gravitation), *vibration* (Vibrationsberechnungen)

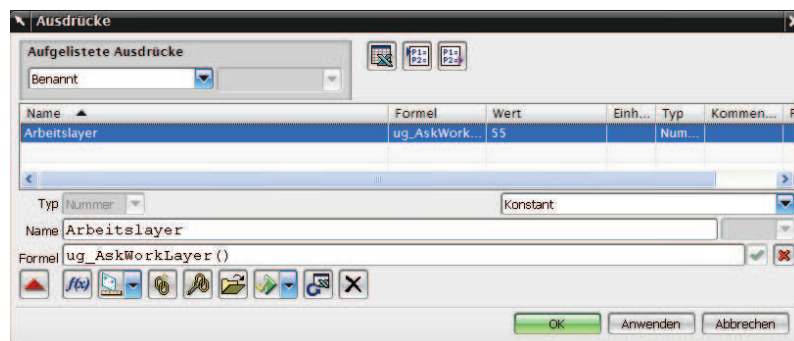
2.9.7 Knowledge Fusion in Expressions



Die im Menüpunkt *Funktionen* verfügbaren Befehle sind nur ein Teil der Möglichkeiten in den Expressions. Es ist auch möglich, andere Befehle aus dem Bereich KF zu verwenden. Einfache Beispiele hierfür sind:

`ug_AskWorkLayer()`, `ug_askCurrentWorkPart()`, `ug_askOSName()`, `RotateVector()`

Nachfolgend wird der aktuelle Arbeitslayer (hier = 55) in die Variable „Arbeitslayer“ geschrieben.

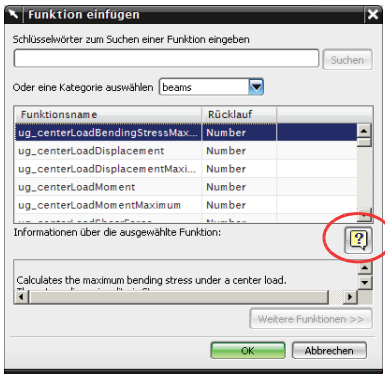


Damit sich diese Variable aktualisiert, wenn der Arbeitslayer wechselt, ist folgende Aktion nötig:

Werkzeuge > Aktualisieren > Aktualisieren für externe Änderung
(*Tools > Update > Update for External Change*)

2.9.8 Benutzung der Online-Hilfe zu den Funktionen

Die einzelnen Parameter der Funktionen können in der UNIGRAPHICS/NX-Hilfe nachgelesen werden:



- Aufruf des Funktionseditors (*Funktionen, Functions*) im Ausdrucksmenü
- *Kategorie (Category)* wählen ...
- Nach einem Klick auf das Fragezeichen wird die Hilfe zum jeweiligen KF-Befehl angezeigt.



ug_centerLoadBendingStressMaximum

Synopsis

Defun: `ug_centerLoadBendingStressMaximum`
 Number, #l,
 Number, #p,
 Number, #c,
 Number, #i)
 @(...) Number;

Description:
 Calculates the maximum bending stress under a center load.
 The return dimensionality is Stress.

$$\sigma_{max} = |M_{max}| \frac{c}{I}$$

Input Arguments:
 l - Length of Beam
 p - Load on Beam
 c - Distance from Neutral Axis to Extreme Fibers
 i - Moment of Inertia

Returns:
 stress - Returns the maximum bending stress

2.9.9 Beispiel: Werte aus Excel einlesen

In diesem Beispiel wird erläutert, wie man Werte aus einer beliebigen Excel-Datei einliest und aus diesen dann einen Vektor konstruiert.

- Zuerst werden die Vektorkomponenten I, J, K aus einer Excel-Datei eingelesen (siehe „ug_cell_read“ und siehe Tabelle oben „Zugriff auf externe Daten“).

	A	B	C
1	i	j	k
2	0,5	0,3	0,7

Name	Formel	Wert
I	ug_cell_read("C:\training\Vektoren.xls", "A2")	0.5
J	ug_cell_read("C:\training\Vektoren.xls", "B2")	0.3
K	ug_cell_read("C:\training\Vektoren.xls", "C2")	0.7
Vektor_1	Vector(I,J,K)	Vector(0.5,0.3,0.7)

- Mit den Werten I, J, K wird ein Ausdruck vom Typ Vektor erzeugt.

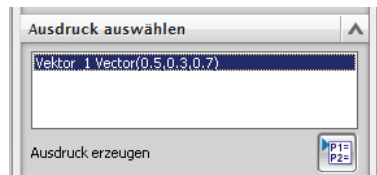
Typ: Vektor
 Name: Vektor_1
 Formel: Vector(I, J, K)

- Nun kann, basierend auf diesen Daten, eine *Bezugsachse* (*Datum Axis*) erzeugt werden.

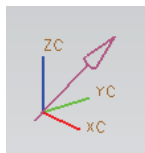


Der Startpunkt des Vektors kann hier beliebig angegeben werden, da er in diesem Beispiel nicht relevant ist.

Um den Vektor zu erzeugen, wählt man den *Vektor-Konstruktor* (*Vector Constructor*) aus. In diesem Menü kann dann „nach Ausdruck“ ausgewählt werden.



Nach dem Bestätigen entsteht nun eine *Bezugsachse*, die abhängig vom Excel-Sheet ist. Die externe Excel-Datei und das Part (die *Bezugsachse*) „kennen sich“ ab jetzt.



Natürlich könnten die I-J-K-Werte auch im aktuellen Part gehalten werden.