

**Fassung  
Februar 2004**

**Programm**

# **RF-COM**

**Programmierbare  
Schnittstelle  
für RFEM**

## **Programm- Beschreibung**

Alle Rechte, auch das der Übersetzung,  
vorbehalten. Ohne ausdrückliche Genehmigung  
der Ingenieur-Software Dlubal GmbH ist es nicht  
gestattet, diese Programm-Beschreibung oder  
Teile daraus auf jedwede Art zu vervielfältigen.

© Ingenieur-Software Dlubal GmbH  
Am Zellweg 2 D-93464 Tiefenbach

Tel.: +49 (0) 9673 9203-0  
Fax: +49 (0) 9673 1770  
E-Mail: [info@dlubal.com](mailto:info@dlubal.com)  
[www.dlubal.de](http://www.dlubal.de)



# Inhalt

	Inhalt	Seite		Inhalt	Seite
1.	<b>Einleitung</b>	<b>4</b>	3.3.9	Punkt-Zwangsverformung erzeugen	70
2.	<b>Voraussetzungen und Möglichkeiten der Schnittstelle</b>	<b>4</b>	3.3.10	Linien-Zwangsverformung erzeugen	71
2.1	Voraussetzungen	4	3.3.11	Imperfektion erzeugen	73
2.2	Möglichkeiten der COM-Schnittstelle	6	3.4	Steuerung der Berechnung	74
3.	<b>Arbeiten mit RF-COM, Befehlsreferenz</b>	<b>6</b>	3.4.1	Berechnungsparameter	74
3.1	Strukturdaten erzeugen	7	3.4.1.1	Berechnungsparameter setzen	74
3.1.1	Punkte erzeugen	7	3.4.2	FE-Netz	81
3.1.2	Linien erzeugen	10	3.4.2.1	FE-Netzparameter setzen	81
3.1.3	Materialien erzeugen	12	3.4.2.2	FE-Netz generieren	82
3.1.4	Flächen erzeugen	13	3.4.3	Starten der Berechnung	83
3.1.5	Öffnungen erzeugen	17	3.4.3.1	Gezielte Ergebnisse berechnen	83
3.1.6	Punktlager erzeugen	18	3.4.3.2	Alle Ergebnisse berechnen	83
3.1.7	Linienlager erzeugen	21	<b>A:</b>	<b>Literatur</b>	<b>85</b>
3.1.8	Flächenbettungen erzeugen	24	<b>B:</b>	<b>Wichtige VBA-Befehle</b>	<b>86</b>
3.1.9	Liniengelenke erzeugen	25	<b>C:</b>	<b>Index</b>	<b>88</b>
3.1.10	Stab-Querschnitte erzeugen	27			
3.1.11	Stabgelenke erzeugen	28			
3.1.12	Exzentrizitäten erzeugen	30			
3.1.13	Stabteilungen erzeugen	31			
3.1.14	Stäbe erzeugen	32			
3.1.15	Stabbettungen erzeugen	35			
3.1.16	Stab-Nichtlinearitäten erzeugen	37			
3.1.17	FE-Netzverdichtungen erzeugen	39			
3.2	Lastfälle, Lastfallgruppen, Lastfallkombinationen	41			
3.2.1	Lastfälle erzeugen	41			
3.2.2	Lastfallgruppen erzeugen	43			
3.2.3	Lastfallkombinationen erzeugen	45			
3.3	Belastungsdaten erzeugen	46			
3.3.1	Punktlasten erzeugen	46			
3.3.2	Stablasten erzeugen	47			
3.3.3	Linienlasten erzeugen	52			
3.3.4	Flächenlasten erzeugen	56			
3.3.5	Freie Punktlasten erzeugen	59			
3.3.6	Freie Linienlasten erzeugen	62			
3.3.7	Freie Rechtecklasten erzeugen	65			
3.3.8	Freie Polygonlasten erzeugen	68			

# 1. Einleitung

Das Programm RFEM bietet viele Möglichkeiten, einfach und schnell Strukturen und Belastungen einzugeben oder die Ergebnisse auszuwerten. Viele Ingenieure berechnen aber immer wieder Projekte, die sich sehr ähnlich sind und nur in bestimmten Parametern abweichen. Ein Generator, der solch eine Struktur auf Knopfdruck in RFEM erzeugen kann, wäre dafür eine große Hilfe. Eventuell müssen auch immer wieder spezielle Nachweise geführt werden, für die RFEM keine Zusatzmodule bietet. Mit Hilfe der COM-Schnittstelle von RFEM ist es leicht möglich, eigene Generatoren und Nachlaufmodule zu entwickeln. Über diese Schnittstelle kann auf fast alle Funktionen von RFEM von außen zugegriffen werden.

Die Schnittstelle setzt keine bestimmte Programmiersprache voraus. Wenn jemand C++ nicht mag, dann ist das also kein Problem. Prinzipiell kann jede Windowsprogrammiersprache verwendet werden, die die Möglichkeit bietet, über die COM-Schnittstelle von Windows mit anderen Anwendungen zu kommunizieren. Neben den „klassischen“ Programmiersprachen wie Visual C++, Visual Basic oder Delphi kann auch Visual Basic for Applications (VBA) von MS-Word oder Excel verwendet werden. Gerade das Zusammenspiel von Excel mit RFEM ist sehr interessant. Deswegen wird die Schnittstelle auch anhand von Beispielen von VBA für Excel erläutert.

Dieses Handbuch soll kein Programmierhandbuch sein. Die Beispiele sind deswegen bewusst einfach gehalten. Grundkenntnisse in der VBA-Programmierung sind aber trotzdem Voraussetzung.

Ein Einblick in die Hintergründe der COM-Technologie ist in [2] im Kapitel 3.4 zu finden

## 2. Voraussetzungen und Möglichkeiten der Schnittstelle

### 2.1 Voraussetzungen

Um RF-COM betreiben zu können, benötigen Sie folgende Dinge:

- eine installierte 3D-Vollversion von RFEM mit gültiger Lizenz und Dongle.
- eine Programmiersprache mit Unterstützung der COM Technologie (z.B. Visual Basic, Visual C++, Delphi, aber auch Excel, Word und Access liefern bereits standardmäßig VBA Editoren mit, die vollkommen ausreichen).
- Programmiergrundkenntnisse. Sie sollten schon einmal ein kleines Programm geschrieben haben und mit Variablen, Funktionen und Objekten umgehen können.

Vorbereitende Maßnahmen:

Bei der Installation von RFEM werden bereits alle programmtechnischen Maßnahmen für den Betrieb der COM-Schnittstelle getroffen. Hierbei wird die Datei RFEMPS.DLL in der Windows-Registry registriert. Sie benötigen hierfür Administrator-Rechte bei der Installation. Die Datei selbst befindet sich im RFEM-Programmverzeichnis. Im gleichen Verzeichnis befindet sich auch die mitgelieferte Objektbibliothek RFEM.TLB für den Befehlsatz von RF-COM. Diese Objektbibliothek muss im Editor der Programmierumgebung eingebunden bzw. es muss darauf verwiesen werden. Alle Vorgänge und Beschreibungen

zu RF-COM werden in diesem Handbuch anhand von MS-Excel erläutert. Zum Einbinden der Typelibrary in MS-Excel gehen Sie bitte wie folgt vor:

- Öffnen Sie in Excel den Visual-Basic-Editor über Extras → Makro → Visual Basic Editor

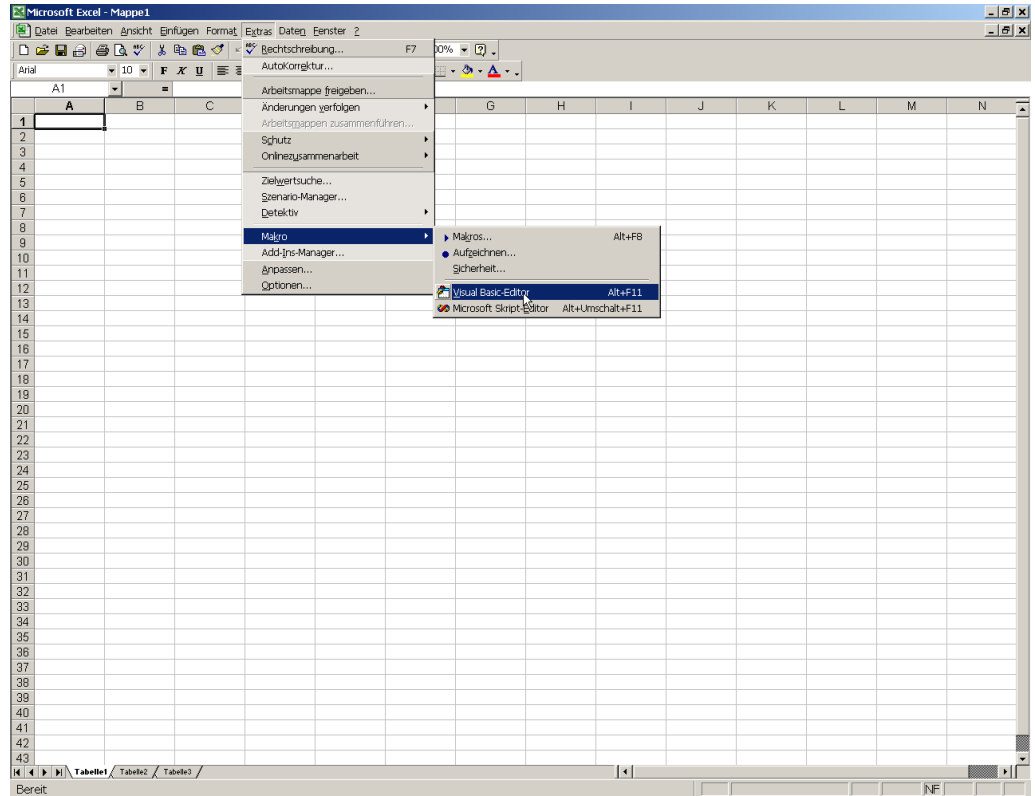


Bild 2.1

- Unter Extras → Verweise binden Sie über die Schaltfläche Durchsuchen die Datei RFEM.TLB (diese befindet sich im RFEM-Programmverzeichnis) in den Editor ein

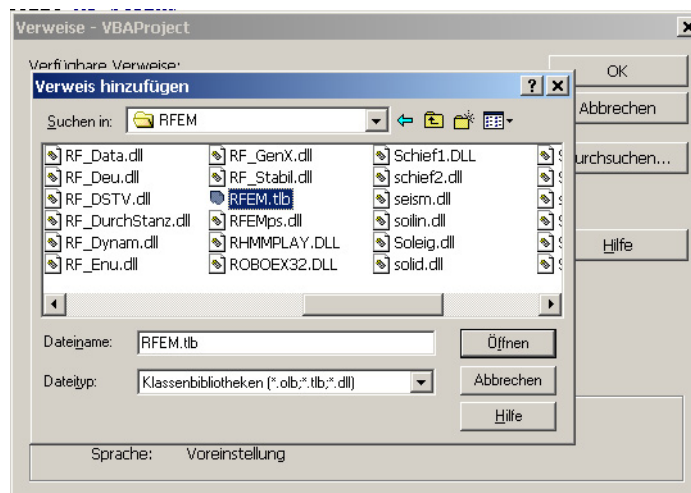


Bild 2.2

- Prüfen Sie, ob die RFEM-Objektbibliothek mit einem Häkchen versehen ist

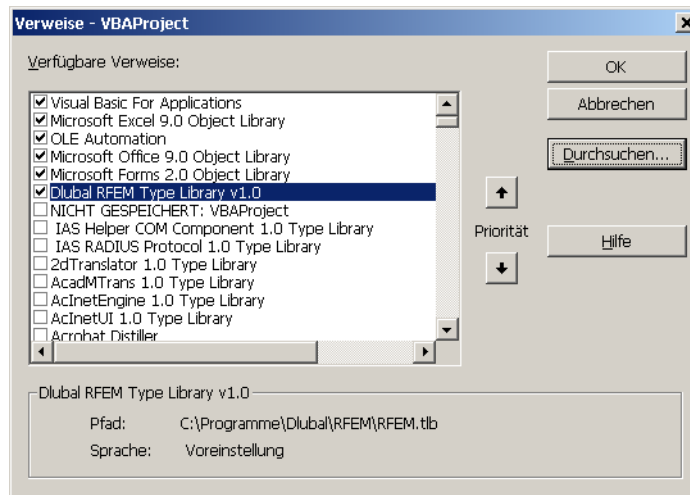


Bild 2.3

Bestätigen Sie mit OK und der Vorgang ist abgeschlossen.

Drücken Sie nun die F-2 Taste und Sie erhalten einen Überblick über alle installierten Objektbibliotheken und deren Befehle. Suchen Sie den Objektkatalog zu RFEM, um einen Überblick über die einzelnen Befehle zu erhalten.

## 2.2 Möglichkeiten der COM-Schnittstelle

Mit RF-COM können Sie Folgendes bewerkstelligen:

- Eingabe, Abfrage und Änderung sämtlicher RFEM-Eingabedaten, Einstellungen und Ergebnisse.
- Bearbeitung von Lastfällen, Lasten, Lastfallgruppen und Lastfallkombinationen mitsamt aller zugehörigen Parametern
- Generieren des FE-Netzes
- Start der Berechnung sowie Änderung der zugehörigen Berechnungsparameter
- Belegen und Freigeben der RF-COM-Lizenz.

RF-COM arbeitet mit Basis SI-Einheiten m, N, und °. Alle Eingaben sind entsprechend vorzugeben, Ergebnisse entsprechend umzurechnen.

## 3. Arbeiten mit RF-COM, Befehlsreferenz

Der beste Weg, um sich in RF-COM einzuarbeiten besteht darin, sich mit dem Befehlssatz anhand von Beispielen vertraut zu machen. Hier soll die Schnittstelle am Beispiel von Visual Basic for Applications für Excel erläutert werden.

Alle nachfolgend erläuterten Quelltexte befinden sich in der Datei Beispiel2.xls. Diese Datei ist im Ordner des Demo-Projektes zu finden, also standardmäßig in c:\programme\dlubal\projekte\demo.

## 3.1 Strukturdaten erzeugen

### 3.1.1 Punkte erzeugen

In einer bestehenden RFEM-Struktur sollen Punkte erzeugt werden. Die Punkte werden im Excel-Tabellenblatt 1.1 definiert und mit dem Makro CommandButton1\_Click() an RFEM übertragen.

Der Ablauf zum Erzeugen von Strukturdaten ist immer ähnlich. In den weiteren Beispielen zur Erzeugung von Strukturdaten ändern sich immer nur wenige Befehle. Deshalb dieses erste Beispiel etwas ausführlicher behandelt.

Der Quelltext ist hier mit Zeilennummern versehen, damit die Erläuterung vereinfacht wird. Im VBA-Editor von Excel werden keine Zeilennummern angezeigt.

```

1:   Dim RFApp As RFEM.IrfApplication
2:   Dim RFPos As RFEM.Position
3:   Dim RFTop As RFEM.IrfTopology
4:   Dim RFLoad As RFEM.IrfLoadCase
5:   Dim RFRes As RFEM.IrfResults
6:
7:   Sub CommandButton1_Click()
8:       Dim intAnzahlZeilen As Integer
9:       Dim Punkt() As RF_NODE
10:      Dim strZelle As String
11:
12:      Set RFPos = GetObject(, "RFEM.Position")
13:      Set RFApp = RFPos.rfGetApplication
14:      RFApp.rfLockLicence
15:      Set RFTop = RFPos.rfGetTopology
16:      intAnzahlZeilen = Tabelle1.Cells(2, 3)
17:      ReDim Punkt(intAnzahlZeilen)
18:      intOffset = 4
19:      For i = 0 To intAnzahlZeilen - 1
20:          Punkt(i).iNo = i + 1
21:          'Spalte A auslesen, Koordinatensystem
22:          strZelle = Tabelle1.Cells(i + intOffset, 1)
23:          Select Case strZelle
24:              Case "Kartesisch"
25:                  Punkt(i).csType = CS_CARTESIAN
26:              Case "X-Zylindrisch"
27:                  Punkt(i).csType = CS_XCYLINDRICAL
28:              Case "Y-Zylindrisch"
29:                  Punkt(i).csType = CS_YCYLINDRICAL
30:              Case "Z-Zylindrisch"
31:                  Punkt(i).csType = CS_ZCYLINDRICAL
32:              Case "Kugelförmig"
33:                  Punkt(i).csType = CS_SPHERICAL
34:              Case "Polar" 'nur bei 2-D
35:                  Punkt(i).csType = CS_POLAR
36:              Case Else 'default Kartesisch
37:                  Punkt(i).csType = CS_CARTESIAN
38:          End Select
39:          'Spalte B auslesen, Bezugspunkt
40:          Punkt(i).iRefNodeNo = Tabelle1.Cells(i + intOffset, 2)
41:          'Spalte C auslesen, X
42:          Punkt(i).x = Tabelle1.Cells(i + intOffset, 3)
43:          'Spalte D auslesen, Y
44:          Punkt(i).y = Tabelle1.Cells(i + intOffset, 4)
45:          'Spalte E auslesen, Z
46:          Punkt(i).z = Tabelle1.Cells(i + intOffset, 5)
47:          'Spalte F auslesen, Kommentar
48:          Punkt(i).strRemark = Tabelle1.Cells(i + intOffset, 6)
49:      Next i
50:      'Übergabe an RFEM
51:      RFTop.rfPrepareModification
52:      RFTop.rfSetNodeArr intAnzahlZeilen, Punkt(0)
53:      RFTop.rfFinishModification
54:      RFApp.rfUnlockLicence
55:      'Freigabe der Objekte
56:      Set RFPos = Nothing

```

```
57:         Set RFTop = Nothing
58:         Set RFApp = Nothing
59:     End Sub
```

In den Zeilen 1: bis 5: werden globale Objekte definiert. Diese Objekte stehen im gesamten Modul zur Verfügung. Das Objekt RFApp stellt ein Interface zu den Applikationsdaten, RFTop zu den Topologiedaten, RFLoad zu den Lastfalldaten und RFRes zu den Ergebnissen her. Die Klassen sind in der Type Library von RFEM definiert.

Ab der Zeile 6 beginnt die eigentliche Prozedur zur Erzeugung der Punkte. Die Prozedur sub CommandButton1\_Click() ist als Event Handler der Schaltfläche [Übernahme] auf dem Tabellenblatt 1.1 zugeordnet, das heißt, wenn auf diese Schaltfläche geklickt wird, dann wird diese Prozedur ausgeführt.

Im Tabellenblatt 1.1 ist die Tabelle 1.1 von RFEM nachgebildet. Die Prozedur CommandButton1\_Click() soll diese Tabelle auslesen und nach RFEM übertragen.

In den Zeilen 7 bis 9 werden einige lokale Variablen definiert, die nur innerhalb dieser Prozedur gelten. In intAnzahlZeilen wird die Anzahl der Zeilen gespeichert, die aus den Excel-Tabellenblatt ausgelesen werden soll.

In der Zeile 8 ist das Feld Punkt() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.1 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_NODE. Dies ist ein vordefinierter Typ.

Dabei handelt es sich um einen so genannten Recorddatentyp. In der Literatur wird dieser Konstrukt auch oft als Strukturvariable bezeichnet. Es handelt sich dabei auch einer Struktur aus mehreren anderen Datentypen. Das können einfache oder auch zusammengesetzte Typen sein.

**Beispiel in Basic:**

```
60:     Type TPerson
61:         Vorname           as string
62:         Nachname          as string
63:     End Type
64:     ...
65:     Dim Ansprechpartner as TPerson
66:     ...
67:     Ansprechpartner.Vorname="Franz"
68:     Ansprechpartner.Nachname="Mustermann"
69:     ...
```

In den Zeilen 60: bis 63: wird der benutzerdefinierte Datentyp definiert. In der Zeile 65: wird eine Variable vom Typ TPerson erzeugt. Wie die einzelnen „Untervariablen“ angesprochen werden, ist in den Zeilen 67: und 68: gezeigt.

**Das gleich Beispiel in C++:**

```
70:     class TPerson{
71:     public:
72:         Sting Vorname;
73:         String Nachname;
74:     };
75:     ...
76:     TPerson Ansprechpartner
77:     ...
78:     Ansprechpartner.Vorname="Franz";
79:     Ansprechpartner.Nachname="Mustermann";
80:     ...
```

RF\_NODE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
csType	Art des Koordinatensystems	Benutzerdefinierter Aufzählungstyp RF_CS_TYPE
iNo	Nummer des Punktes	Long

iRefNodeNo	Nummer des Referenzpunktes	Long
strRemark	Kommentar	String
x	x-Koordinate [m]	Single
y	y-Koordinate [m]	Single
z	z- Koordinate [m]	Single

Tabelle 3.1

Der benutzerdefinierte Aufzählungstyp RF\_CS\_TYPE kann die Werte von 0 bis 5 annehmen. Zur besseren Lesbarkeit sind die Zahlenwerte Konstanten zugeordnet. Prinzipiell ist es egal, ob man die Zahlenwerte oder die Konstanten im Quelltext verwendet. Um die Lesbarkeit des Quelltextes zu erhalten, sollten aber auf jeden Fall die Konstanten verwendet werden, auch wenn dabei etwas mehr zu schreiben ist.

Zahlenwert	Konstante	Bedeutung
0	CS_CARTESIAN	Kartesisches Koordinatensystem
1	CS_XCYLINDRICAL	X-Zylindrisches Koordinatensystem
2	CS_YCYLINDRICAL	Y- Zylindrisches Koordinatensystem
3	CS_ZCYLINDRICAL	Z- Zylindrisches Koordinatensystem
4	CS_SPHERICAL	Kugelförmiges Koordinatensystem
5	CS_POLAR	Polares Koordinatensystem

Tabelle 3.2

Die Variable strZelle dient als Zwischenspeicher für den Inhalt von Tabellenzellen.

In den Zeilen 11 bis 12 werden die Objekte RFPos und RFApp initialisiert.

In der Zeile 13 wird eine Methode von RFApp aufgerufen, die eine Lizenz für RF-COM belegt. Die Lizenzverwaltung muss vom Programmierer selber übernommen werden. Diese besteht aus zwei Zeilen. In der Zeile 13 wird die Lizenz belegt und in der Zeile 53 wieder freigegeben. Daraus wird ersichtlich, dass die Lizenz nur für die Laufzeit der Prozedur benötigt wird, also nicht während man mit der Excel-Tabelle arbeitet. Wichtig in diesem Zusammenhang ist noch, dass die Methode rfLockLicenc erst aufgerufen werden kann, nachdem in der Zeile 12 das Objekt RFApp initialisiert wurde.

In der Zeile 14 wird das Objekt RFTop initialisiert.

In der Zeile 15 wird festgelegt, wie viele Zeile aus dem Excel-Tabellenblatt 1.1 nach RFEM übertragen werden sollen. Die Anzahl wird in der Zelle C2 eingegeben. In der Zeile 15 wird diese Zelle ausgelesen in der Variablen intAnzahlZeilen abgespeichert. Damit ist auch die genaue Größe des Feldes Punkt() bekannt. Diese Größe wird in der Zeile 16 mit ReDim neu festgelegt.

In der Zeile 17 wird der Hilfsvariablen intOffset der Wert 4 zugewiesen. Die Tabellenzeilen im Tabellenblatt 1.1 sollen ab der 4. Zeile eingelesen werden.

In der folgenden For-Schleife werden die Tabellenzeilen zeilenweise eingelesen. Die Variable intOffset wird immer zur Zahlvariablen i der Schleife hinzuaddiert und daraus ergibt sich die Zeilennummer im Excel-Tabellenblatt.

In der Zeile 19 wird im Element iNo die Zeilennummer gespeichert. Die Laufvariable i wird um 1 erhöht, weil die For-Schleife von 0 an zählt.

In den Zeilen 20 bis 37 wird das Element csType belegt. Der Wert könnte auch direkt aus der Zelle eingelesen werden. Dann müssten aber in der Zelle Zahlenwerte stehen. Diese

Zahlenwerte hätten dann die Bedeutung, wie in Tabelle 3.2 beschrieben. Diese Vorgehensweise ist aber schlecht lesbar und damit fehleranfällig.

Deswegen sollen in der Tabellenzelle verständliche Einträge wie „Kartesisch“ stehen. Der Inhalt der Zelle wird mittels der Cells-Eigenschaft (siehe S. 87) in die Variable strZelle übertragen. Mit der Select-Case-Anweisung (siehe S. 86) wird die Variable strZelle ausgewertet und das Element csType mit den entsprechenden Konstanten nach Tabelle 3.2 belegt.

In den Zeilen 39 bis 47 werden die restlichen Elemente des Records mittels der Cells-Eigenschaft direkt eingelesen.

Jetzt sind alle Daten, die an RFEM übergeben werden sollen, im Feld Punkt() gespeichert. In den Zeilen 50 bis 52 erfolgt die Übergabe.

Mit der Methode rfPrepareModification wird unter anderem die grafischen Oberfläche von RFEM gesperrt.

Die Methode rfSetNodeArr übergibt das Feld Punkt() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetNodeArr mitgeteilt, wie viele Zeilen aus dem Feld Punkt() übertragen werden sollen.

Mit der Methode rfFinishModification wird die grafischen Oberfläche von RFEM wieder freigegeben.

In den Zeilen 55 bis 57 wird der Speicherplatz der benutzten Objekte wieder freigegeben.

### 3.1.2 Linien erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Linien erzeugt werden. Voraussetzung dafür ist, dass schon die Punkte für die Linien existieren. Die Linien werden im Excel-Tabellenblatt 1.2 definiert und mit dem Makro CommandButton2\_Click() an RFEM übertragen.

```

81: Sub CommandButton2_Click()
82:     Dim intAnzahlZeilen As Integer
83:     Dim Linie() As RF_LINE
84:     Dim strZelle As String
85:
86:     Set RFPos = GetObject(, "RFEM.Position")
87:     Set RFApp = RFPos.rfGetApplication
88:     RFApp.rfLockLicence
89:     Set RFTop = RFPos.rfGetTopology
90:     intAnzahlZeilen = Tabelle2.Cells(2, 3)
91:     ReDim Linie(intAnzahlZeilen)
92:     intOffset = 4
93:     For i = 0 To intAnzahlZeilen - 1
94:         Linie(i).iNo = i + 1
95:         'Spalte A auslesen, Liniertyp
96:         strZelle = Tabelle2.Cells(i + intOffset, 1)
97:         Select Case strZelle
98:             Case "Polylinie"
99:                 Linie(i).lineType = LT_POLYLINE
100:            Case "Bogen"
101:                Linie(i).lineType = LT_ARC
102:            Case "Kreis"
103:                Linie(i).lineType = LT_CIRCLE
104:            Case "Spline"
105:                Linie(i).lineType = LT_SPLINE_CUB
106:            Case "B-Spline"
107:                Linie(i).lineType = LT_SPLINE_B
108:            Case "Bezier-Spline"
109:                Linie(i).lineType = LT_SPLINE_BEZ
110:            Case Else 'default Polylinie
111:                Linie(i).lineType = LT_POLYLINE

```

```

112:         End Select
113:         'Spalte B auslesen, Punktliste
114:         Linie(i).strNodeList = Tabelle2.Cells(i + intOffset, 2)
115:         'Spalte D auslesen, Kommentar
116:         Linie(i).strRemark = Tabelle2.Cells(i + intOffset, 4)
117:     Next i
118:     'Übergabe an RFEM
119:     RFTop.rfPrepareModification
120:     RFTop.rfSetLineArr intAnzahlZeilen, Linie(0)
121:     RFTop.rfFinishModification
122:     RFApp.rfUnlockLicence
123:     'Freigabe der Objekte
124:     Set RFPos = Nothing
125:     Set RFTop = Nothing
126:     Set RFApp = Nothing
127: End Sub
    
```

In der Zeile 83: ist das Feld Linie() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.2 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LINE.

RF\_LINE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
iNo	Nummer der Linie	Long
iNodeCount	Anzahl der Linien, kann nur gelesen werden	Long
length	Länge der Linie, kann nur gelesen werden [m]	Double
lineType	Linientyp	Benutzerdefinierter Aufzählungstyp RF_LINE_TYPE
strNodeList	Liste der Punkte Wenn Punkte in der Liste enthalten sind, die nicht existieren, dann wird bei der Übergabe in Zeile 120: ein Laufzeitfehler ausgegeben.	String
strRemark	Kommentar	String

Tabelle 3.3

Im benutzerdefinierten Aufzählungstyp RF\_LINE\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
1	LT_POLYLINE	Polylinie
2	LT_SPLINE_CUB	Kubische Spline-Linie
3	LT_CIRCLE	Kreis
4	LT_ARC	Kreisbogen
7	LT_SPLINE_B	B-Spline-Linie
8	LT_SPLINE_BEZ	Bezier-Spline-Linie

Tabelle 3.4

Die Methode rfSetLineArr übergibt das Feld Linie() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetLineArr mitgeteilt, wie viele Zeilen aus dem Feld Linie() übertragen werden sollen.

### 3.1.3 Materialien erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Materialien erzeugt werden. Die Materialien werden im Excel-Tabellenblatt 1.3 definiert und mit dem Makro `CommandButton3_Click()` an RFEM übertragen.

```

128: Sub CommandButton3_Click()
129:     Dim intAnzahlZeilen As Integer
130:     Dim Material() As RF_MATERIAL
131:
132:     Set RFPos = GetObject(, "RFEM.Position")
133:     Set RFApp = RFPos.rfGetApplication
134:     RFApp.rfLockLicence
135:     Set RFTop = RFPos.rfGetTopology
136:     intAnzahlZeilen = Tabelle3.Cells(2, 3)
137:     ReDim Material(intAnzahlZeilen)
138:     intOffset = 4
139:     For i = 0 To intAnzahlZeilen - 1
140:         Material(i).iNo = i + 1
141:         'Spalte A auslesen, Materialname
142:         Material(i).strDescription = Tabelle3.Cells(i + intOffset, 1)
143:         'Spalte B auslesen, E
144:         Material(i).fE = Tabelle3.Cells(i + intOffset, 2)
145:         'Spalte C auslesen, G
146:         Material(i).fG = Tabelle3.Cells(i + intOffset, 3)
147:         'Spalte D auslesen, µ
148:         Material(i).fMi = Tabelle3.Cells(i + intOffset, 4)
149:         'Spalte E auslesen, gamma
150:         Material(i).fRo = Tabelle3.Cells(i + intOffset, 5)
151:         'Spalte F auslesen, alpha
152:         Material(i).fAlpha = Tabelle3.Cells(i + intOffset, 6)
153:         'Spalte G auslesen, Kommentar
154:         Material(i).strRemark = Tabelle3.Cells(i + intOffset, 7)
155:     Next i
156:     'Übergabe an RFEM
157:     RFTop.rfPrepareModification
158:     RFTop.rfSetMaterialArr intAnzahlZeilen, Material(0)
159:     RFTop.rfFinishModification
160:     RFApp.rfUnlockLicence
161:     'Freigabe der Objekte
162:     Set RFPos = Nothing
163:     Set RFTop = Nothing
164:     Set RFApp = Nothing
165: End Sub
    
```

In der Zeile 130: ist das Feld `Material()` definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.3 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ `RF_MATERIAL`.

`RF_MATERIAL` ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
fAlpha	Wärmedehnzahl	Single
fE	E-Modul [N/m <sup>2</sup> ]	Single
fG	G-Modul [N/m <sup>2</sup> ]	Single
fMi	Querdehnzahl	Single
fRo	Spezifisches Gewicht [N/m <sup>3</sup> ]	Single
iNo	Materialnummer	Long
strDescription	Materialname	String
strRemark	Kommentar	String

Tabelle 3.5

Wenn sich in strDescription ein Materialname befindet, der in der Materialdatenbank von RFEM vorhanden ist, dann brauchen keine weiteren Materialkennwerte übergeben werden. Es werden dann die Kennwerte aus der RFEM-Bibliothek eingelesen.

Die Methode rfSetMaterialArr übergibt das Feld Material() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetMaterialArr mitgeteilt, wie viele Zeilen aus dem Feld Material() übertragen werden sollen.

### 3.1.4 Flächen erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Flächen erzeugt werden. Voraussetzungen sind, dass bereits die Begrenzungslinien der Flächen und die Materialdaten definiert sind. Die Flächen werden im Excel-Tabellenblatt 1.4 definiert und mit dem Makro CommandButton4\_Click() an RFEM übertragen.

```

166: Sub CommandButton4_Click()
167: Dim intAnzahlZeilen As Integer
168: Dim Flaeche() As RF_SURFACE
169:
170: Set RFPos = GetObject(, "RFEM.Position")
171: Set RFApp = RFPos.rfGetApplication
172: RFApp.rfLockLicence
173: Set RFTop = RFPos.rfGetTopology
174: intAnzahlZeilen = Tabelle4.Cells(2, 3)
175: ReDim Flaeche(intAnzahlZeilen)
176: intOffset = 4
177: For i = 0 To intAnzahlZeilen - 1
178:   Flaeche(i).iNo = i + 1
179:   'Spalte A auslesen, Flächentyp
180:   strZelle = Tabelle4.Cells(i + intOffset, 1)
181:   Select Case strZelle
182:     Case "Eben"
183:       Flaeche(i).surfaceType = ST_LINEAR
184:     Case "Quadrangle"
185:       Flaeche(i).surfaceType = ST_QUAD
186:     Case "B-Spline"
187:       Flaeche(i).surfaceType = ST_BSPLINE
188:     Case "Rotationsfläche"
189:       Flaeche(i).surfaceType = ST_ROT
190:     Case "Rohr"
191:       Flaeche(i).surfaceType = ST_PIPELINE
192:     Case Else 'default Eben
193:       Flaeche(i).surfaceType = ST_LINEAR
194:   End Select
195:   'Spalte B auslesen, Liste der Begrenzungslinien
196:   Flaeche(i).strBLineList = Tabelle4.Cells(i + intOffset, 2)
197:   'Spalte C auslesen, Materialnummer
198:   Flaeche(i).iMaterialNo = Tabelle4.Cells(i + intOffset, 3)
199:   'Spalte D auslesen, Dickentyp
200:   strZelle = Tabelle4.Cells(i + intOffset, 4)
201:   Select Case strZelle
202:     Case "Konstant"
203:       Flaeche(i).thickType = TT_CONST
204:     Case "Linear"
205:       Flaeche(i).thickType = TT_LINEAR
206:     Case "Orthotrop"
207:       Flaeche(i).thickType = TT_ORTO
208:     Case Else 'default Eben
209:       Flaeche(i).thickType = TT_CONST
210:   End Select
211:   'Spalte E auslesen, Dicke in [m]
212:   Flaeche(i).thickConst = Tabelle4.Cells(i + intOffset, 5)
213: Next i
214: 'Übergabe an RFEM

```

```

215:      RFTop.rfPrepareModification
216:      RFTop.rfSetSurfaceArr intAnzahlZeilen, Flaeche(0)
217:      RFTop.rfFinishModification
218:      RFApp.rfUnlockLicence
219:      'Freigabe der Objekte
220:      Set RFPos = Nothing
221:      Set RFTop = Nothing
222:      Set RFApp = Nothing
223:      End Sub
    
```

In der Zeile 168: ist das Feld Flaeche() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.4 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_SURFACE.

RF\_SURFACE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
dArea	Flächeninhalt, kann nur gelesen werden [m <sup>2</sup> ]	Double
iBLineCount	Anzahl der Begrenzungslinien, kann nur gelesen werden	Long
iHoleCount	Anzahl der Öffnungen, kann nur gelesen werden	Long
iILineCount	Anzahl der integrierten Linien, kann nur gelesen werden	Long
iINodeCount	Anzahl der integrierten Punkte, kann nur gelesen werden	Long
iMaterialNo	Materialnummer Wenn eine nicht existierende Materialnummer übergeben wird, dann wird ein Laufzeitfehler ausgegeben.	Long
iNo	Nummer der Fläche	Long
strBLineList	Liste der Begrenzungslinien Wenn eine nicht existierende Liniennummer übergeben wird, dann wird ein Laufzeitfehler ausgegeben.	String
strHoleList	Liste der integrierten Öffnungen, kann nur gelesen werden	String
strILineList	Liste der integrierten Linien, kann nur gelesen werden	String
strINodeList	Liste der integrierten Punkte, kann nur gelesen werden	String
strRemark	Kommentar	String
surfaceType	Typ der Fläche	Benutzerdefinierter Aufzählungstyp RF_SURFACE_TYPE
thickConst	Dicke der Fläche [m]	Single
thickLin	Lineare Dicke	Benutzerdefinierter Typ RF_THICK_LIN

thickOrto	Orthotrope Dicke	Benutzerdefinierter Typ RF_THICK_ORTO
thickType	Dickentyp	Benutzerdefinierter Aufzählungstyp RF_THICK_TYPE

Tabelle 3.6

Im benutzerdefinierten Aufzählungstyp RF\_SURFACE\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
2	ST_QUAD	Quadrangle-Fläche
3	ST_BSPLINE	B-Spline-Fläche
4	ST_ROT	Rotationsfläche
5	ST_PIPELINE	Rohr
6	ST_LINEAR	Ebene Fläche

Tabelle 3.7

RF\_THICK\_LIN ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
iNo	Nummer der linearen Dicke	Long
iNode1No	Nummer des Punktes 1	Long
iNode2No	Nummer des Punktes 3	Long
iNode3No	Nummer des Punktes 3	Long
strRemark	Kommentar	String
thick1	Dicke am Punkt 1 [m]	Single
thick2	Dicke am Punkt 2 [m]	Single
thick3	Dicke am Punkt 3 [m]	Single

Tabelle 3.8

RF\_THICK\_ORTO ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
fAngle	Winkel der Orthotropie [°]	Single
fThickX	Dicke in X'-Richtung [m]	Single
fThickY	Dicke in X'-Richtung [m]	Single
fTorsSolid	Drillsteifigkeit [%]	Single
iMaterialX	Materialnummer in X'-Richtung Wenn eine nicht existierende Materialnummer übergeben wird, dann wird ein Laufzeitfehler ausgegeben.	Long
iMaterialY	Materialnummer in Y'-Richtung Wenn eine nicht existierende Materialnummer übergeben wird,	Long

	dann wird ein Laufzeitfehler ausgegeben.	
iNo	Nummer der orthotropen Dicke	Long
mem	Membran-Steifigkeitsmatrix	Benutzerdefinierter Typ RF_MEM
orto	Biege-Steifigkeitsmatrix	Benutzerdefinierter Typ RF_ORTO
strRemark	Kommentar	String
thickBy	Art, wie die orthotrope Dicke definiert ist	Benutzerdefinierter Typ RF_ORTO_THICK_BY

Tabelle 3.9

Im benutzerdefinierten Aufzählungstyp RF\_THICK\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
1	TT_CONST	Konstante Dicke
2	TT_LINEAR	Linear veränderliche Dicke
3	TT_ORTO	Orthotrope Dicke

Tabelle 3.10

RF\_MEM ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
fMemD11	Koeffizient d11	Single
fMemD12	Koeffizient d12	Single
fMemD22	Koeffizient d22	Single
fMemD33	Koeffizient d33	Single

Tabelle 3.11

RF\_ORTO ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
fOrtoD11	Koeffizient D11	Single
fOrtoD12	Koeffizient D12	Single
fOrtoD22	Koeffizient D22	Single
fOrtoD33	Koeffizient D33	Single
fOrtoD44	Koeffizient D44	Single
fOrtoD55	Koeffizient D55	Single

Tabelle 3.12

Im benutzerdefinierten Aufzählungstyp RF\_ORTO\_THICK\_BY sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	BY_THICK	Definition über Dicken
1	BY_MATRIX	Definition über die Koeffizienten der Steifigkeitsmatrix

Tabelle 3.13

Die Methode rfSetSurfaceArr übergibt das Feld Flaeche() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetSurfaceArr mitgeteilt, wie viele Zeilen aus dem Feld Flaeche() übertragen werden sollen.

### 3.1.5 Öffnungen erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Öffnungen erzeugt werden. Voraussetzungen sind, dass bereits die Begrenzungslinien der Öffnungen definiert sind. Die Öffnungen werden im Excel-Tabellenblatt 1.5 definiert und mit dem Makro CommandButton5\_Click() an RFEM übertragen.

```

224: Sub CommandButton5_Click()
225:   Dim intAnzahlZeilen As Integer
226:   Dim Oeffnungen() As RF_HOLE
227:
228:   Set RFPos = GetObject(, "RFEM.Position")
229:   Set RFApp = RFPos.rfGetApplication
230:   RFApp.rfLockLicence
231:   Set RFTop = RFPos.rfGetTopology
232:   intAnzahlZeilen = Tabelle21.Cells(2, 3)
233:   'Die Größe des Feldes Punkt wird neu festgelegt
234:   ReDim Oeffnungen(intAnzahlZeilen)
235:   'Schleife zum Auslesen der Zellen
236:   intOffset = 4
237:   For i = 0 To intAnzahlZeilen - 1
238:     Oeffnungen(i).iNo = i + 1
239:     'Spalte A, Linienliste
240:     Oeffnungen(i).strBLineList = Tabelle21.Cells(i + intOffset, 1)
241:     'Spalte B, Kommentar
242:     Oeffnungen(i).strRemark = Tabelle21.Cells(i + intOffset, 2)
243:   Next i
244:   RFTop.rfPrepareModification
245:   RFTop.rfSetHoleArr intAnzahlZeilen, Oeffnungen(0)
246:   RFTop.rfFinishModification
247:   RFApp.rfUnlockLicence
248:   'Freigabe der Objekte
249:   Set RFPos = Nothing
250:   Set RFTop = Nothing
251:   Set RFApp = Nothing
252: End Sub
    
```

In der Zeile 226: ist das Feld Oeffnung() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.5 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_HOLE.

RF\_HOLE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
dArea	Flächeninhalt der Öffnung, kann nur gelesen werden [m²]	Double
iBLineCount	Anzahl der Begrenzungslinien, kann nur gelesen werden	Long
iNo	Nummer der Öffnung	Long
inSurfaceNo	Nummer der Fläche, in der sich die Öffnung befindet Bei ebenen Flächen muss die Flächennummer nicht übergeben werden.	Long

strBLineList	Liste der Begrenzungslinien	String
strRemark	Kommentar	String

Tabelle 3.14

Die Methode rfSetHoleArr übergibt das Feld Oeffnung() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetHoleArr mitgeteilt, wie viele Zeilen aus dem Feld Oeffnung() übertragen werden sollen.

### 3.1.6 Punktlager erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Punktlager erzeugt werden. Voraussetzungen sind, dass bereits die Punkte definiert sind. Die Punktlager werden im Excel-Tabellenblatt 1.6 definiert und mit dem Makro CommandButton7\_Click() an RFEM übertragen.

```

253: Sub CommandButton7_Click()
254:   Dim intAnzahlZeilen As Integer
255:   Dim Punktlager() As RF_SUPPORT_NODE
256:
257:   Set RFPos = GetObject(, "RFEM.Position")
258:   Set RFApp = RFPos.rfGetApplication
259:   RFApp.rfLockLicence
260:   Set RFTop = RFPos.rfGetTopology
261:   intAnzahlZeilen = Tabelle7.Cells(2, 3)
262:   'Die Größe des Feldes Punkt wird neu festgelegt
263:   ReDim Punktlager(intAnzahlZeilen)
264:   'Schleife zum Auslesen der Zellen
265:   intOffset = 4
266:   For i = 0 To intAnzahlZeilen - 1
267:     Punktlager(i).iNo = i + 1
268:     'Spalte A auslesen, Punktlage
269:     Punktlager(i).strNodeList = Tabelle7.Cells(i + intOffset, 1)
270:     'Spalte B, Stützentyp
271:     Punktlager(i).isColumn = False
272:     'Spalte D auslesen, Lagerdrehung
273:     strZelle = Tabelle7.Cells(i + intOffset, 4)
274:     Select Case strZelle
275:       Case "XYZ"
276:         Punktlager(i).Order = ORDER_XYZ
277:       Case "XZY"
278:         Punktlager(i).Order = ORDER_XZY
279:       Case "YXZ"
280:         Punktlager(i).Order = ORDER_YXZ
281:       Case "YZX"
282:         Punktlager(i).Order = ORDER_YZX
283:       Case "ZXY"
284:         Punktlager(i).Order = ORDER_ZXY
285:       Case "ZYX"
286:         Punktlager(i).Order = ORDER_ZYX
287:       Case Else 'default XYZ
288:         Punktlager(i).Order = ORDER_XYZ
289:     End Select
290:     'Spalte E auslesen, um X'
291:     Punktlager(i).fRotX = Tabelle7.Cells(i + intOffset, 5)
292:     'Spalte F auslesen, um Y'
293:     Punktlager(i).fRotY = Tabelle7.Cells(i + intOffset, 6)
294:     'Spalte G auslesen, um Z'
295:     Punktlager(i).fRotZ = Tabelle7.Cells(i + intOffset, 7)
296:     'Spalte H auslesen, Stützung X
297:     Punktlager(i).fAlongX = Tabelle7.Cells(i + intOffset, 8)
298:     'Spalte I auslesen, Stützung Y
299:     Punktlager(i).fAlongY = Tabelle7.Cells(i + intOffset, 9)
300:     'Spalte J auslesen, Stützung Z
301:     Punktlager(i).fAlongZ = Tabelle7.Cells(i + intOffset, 10)
302:     'Spalte K auslesen, Stützung um X
303:     Punktlager(i).fAroundX = Tabelle7.Cells(i + intOffset, 11)

```

```

304:         'Spalte L auslesen, Stützung um Y
305:         Punktlager(i).fAroundY = Tabelle7.Cells(i + intOffset, 12)
306:         'Spalte M auslesen, Stützung um Z
307:         Punktlager(i).fAroundZ = Tabelle7.Cells(i + intOffset, 13)
308:         'Spalte N wird vorläufig fest gesetzt auf kein Ausfall
309:         Punktlager(i).exclusionX = ET_NONE
310:         Punktlager(i).exclusionY = ET_NONE
311:         Punktlager(i).exclusionZ = ET_NONE
312:     Next i
313:     'Übergabe an RFEM
314:     RFTop.rfPrepareModification
315:     RFTop.rfSetSuppNodeArr intAnzahlZeilen, Punktlager(0)
316:     RFTop.rfFinishModification
317:     RFApp.rfUnlockLicence
318:     'Freigabe der Objekte
319:     Set RFPos = Nothing
320:     Set RFTop = Nothing
321:     Set RFApp = Nothing
322: End Sub
    
```

In der Zeile 255: ist das Feld Punktlager() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.6 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_SUPPORT\_NODE.

RF\_SUPPORT\_NODE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
column	Stütze	Benutzerdefinierter Typ RF_COLUMN
exclusionX	Ausschluss in X-Richtung	Benutzerdefinierter Aufzählungstyp RF_EXCLUSION_TYPE
exclusionY	Ausschluss in Y-Richtung	Benutzerdefinierter Aufzählungstyp RF_EXCLUSION_TYPE
exclusionZ	Ausschluss in Z-Richtung	Benutzerdefinierter Aufzählungstyp RF_EXCLUSION_TYPE
fAlongX	Stützung in X, Feder in X Wenn der Wert -1 übergeben wird, dann wird ein starres Auflager gesetzt. Bei positiven Zahlenwerten werden diese als Feder übernommen. [N/m]	Single
fAlongY	Stützung in Y, Feder in Y	Single
fAlongZ	Stützung in Z, Feder in Z	Single
fAroundX	Einspannung um X, Feder um X Wenn der Wert -1 übergeben wird, dann wird eine starre Einspannung gesetzt. Bei Positiven Zahlenwerten werden diese als Feder übernommen. [Nm/rad]	Single
fAroundY	Einspannung um Y, Feder um Y	Single
fAroundZ	Einspannung um Z, Feder um Z	Single
fRotX	Drehung des Auflagers um die X-Achse [°]	Single
fRotY	Drehung des Auflagers um die Y-Achse [°]	Single

	Achse [°]	
fRotZ	Drehung des Auflagers um die Z-Achse [°]	Single
iNo	Nummer des Auflagers	Long
isColumn	Stütze vorhanden, ja, nein	Boolean
order	Reihenfolge der Stützendrehung	Benutzerdefinierter Aufzählungstyp RF_ROT_ORDER
strNodeList	Liste der gelagerten Punkte	String
strRemark	Kommentar	String

Tabelle 3.15

RF\_COLUMN ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bCalcCol	Soll die Stütze in der Berechnung berücksichtigt werden, ja,nein	Boolean
downSupport	Lagerung der Stütze am Fußpunkt	Benutzerdefinierter Aufzählungstyp RF_COLUMN_SUPP
fHeight	Stützhöhe [m]	Single
fSizeX	Abmessung in x-Richtung, Durchmesser [m]	Single
fSizeY	Abmessung in x-Richtung [m]	Single
fSuppElast	Einspanngrad am Fußpunkt [%]	Single
iMaterialNo	Materialnummer	Long
type	Stützentyp	Benutzerdefinierter Aufzählungstyp RF_COLUMN_TYPE

Tabelle 3.16

Im benutzerdefinierten Aufzählungstyp RF\_EXCLUSION\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	ET_NONE	Kein Ausschluss
1	ET_NOTENS	Ausschluss bei Zug
2	ET_NOPRES	Ausschluss bei Druck

Tabelle 3.17

Im benutzerdefinierten Aufzählungstyp RF\_ROT\_ORDER sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	ORDER_ZYX	Reihenfolge ZYX
1	ORDER_ZXY	Reihenfolge ZXY
2	ORDER_YZX	Reihenfolge YZX
3	ORDER_YXZ	Reihenfolge YXZ

4	ORDER_XZY	Reihenfolge XZY
5	ORDER_XYZ	Reihenfolge XYZ

Tabelle 3.18

Im benutzerdefinierten Aufzählungstyp RF\_COLUMN\_SUPP sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	COLUMN_SUPP_RELEASE	Gelenkig
1	COLUMN_SUPP_ELAST	Elastisch, der Einspanngrad ist in RF_COLUMN definiert.
2	COLUMN_SUPP_RIGID	Eingespannt

Tabelle 3.19

Im benutzerdefinierten Aufzählungstyp RF\_COLUMN\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	COLUMN_RECT	Rechteckig
1	COLUMN_CIRC	Rund

Tabelle 3.20

Die Methode rfSetSuppNodeArr übergibt das Feld Punktlager() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetSuppNodeArr mitgeteilt, wie viele Zeilen aus dem Feld Punktlager() übertragen werden sollen.

### 3.1.7 Linienlager erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Linienlager erzeugt werden. Voraussetzungen sind, dass bereits die Linien definiert sind. Die Linienlager werden im Excel-Tabellenblatt 1.7 definiert und mit dem Makro CommandButton7\_Click() an RFEM übertragen.

```

323: Sub CommandButton7_Click()
324: Dim intAnzahlZeilen As Integer
325: Dim Linienlager() As RF_SUPPORT_LINE
326:
327: Set RFPos = GetObject(, "RFEM.Position")
328: Set RFApp = RFPos.rfGetApplication
329: RFApp.rfLockLicence
330: Set RFTop = RFPos.rfGetTopology
331: intAnzahlZeilen = Tabelle13.Cells(2, 3)
332: 'Die Größe des Feldes Punkt wird neu festgelegt
333: ReDim Linienlager(intAnzahlZeilen)
334: 'Schleife zum Auslesen der Zellen
335: intOffset = 4
336: For i = 0 To intAnzahlZeilen - 1
337: Linienlager(i).iNo = i + 1
338: 'Spalte A, Liste der gelagerten Linien
339: Linienlager(i).strLineList = Tabelle13.Cells(i + intOffset, 1)
340: 'Spalte B, Bezugsachse
341: strZelle = Tabelle13.Cells(i + intOffset, 2)
342: Select Case strZelle
343: Case "Global"
344: Linienlager(i).isLocal = False
345: Case "Lokal"
346: Linienlager(i).isLocal = True

```

```

347:             Case Else 'default Global
348:                 Linienlager(i).isLocal = False
349:             End Select
350:             'Spalte C, Stützung X
351:             Linienlager(i).fAlongX = Tabelle13.Cells(i + intOffset, 3)
352:             'Spalte D, Stützung Y
353:             Linienlager(i).fAlongY = Tabelle13.Cells(i + intOffset, 4)
354:             'Spalte E, Stützung Z
355:             Linienlager(i).fAlongZ = Tabelle13.Cells(i + intOffset, 5)
356:             'Spalte F, Einspannung um X
357:             Linienlager(i).fAroundX = Tabelle13.Cells(i + intOffset, 6)
358:             'Spalte G, Einspannung um Y
359:             Linienlager(i).fAroundY = Tabelle13.Cells(i + intOffset, 7)
360:             'Spalte H, Einspannung um Z
361:             Linienlager(i).fAroundZ = Tabelle13.Cells(i + intOffset, 8)
362:             'Spalte I, Drehung
363:             Linienlager(i).fRotAngle = Tabelle13.Cells(i + intOffset, 9)
364:             'Spalte J, Ausfall
365:             strZelle = Tabelle13.Cells(i + intOffset, 10)
366:             strErsterWert = Left(strZelle, 1)
367:             strZweiterWert = Left(strZelle, 3)
368:             strZweiterWert = Right(strZweiterWert, 1)
369:             strDritterWert = Left(strZelle, 5)
370:             strDritterWert = Right(strDritterWert, 1)
371:             Select Case strErsterWert
372:                 Case "0"
373:                     Linienlager(i).exclusionX = ET_NONE
374:                 Case "1"
375:                     Linienlager(i).exclusionX = ET_NOTENS
376:                 Case "2"
377:                     Linienlager(i).exclusionX = ET_NOPRES
378:                 Case Else 'default Kein Ausfall
379:                     Linienlager(i).exclusionX = ET_NONE
380:             End Select
381:             Select Case strZweiterWert
382:                 Case "0"
383:                     Linienlager(i).exclusionY = ET_NONE
384:                 Case "1"
385:                     Linienlager(i).exclusionY = ET_NOTENS
386:                 Case "2"
387:                     Linienlager(i).exclusionY = ET_NOPRES
388:                 Case Else 'default Global
389:                     Linienlager(i).exclusionY = ET_NONE
390:             End Select
391:             Select Case strDritterWert
392:                 Case "0"
393:                     Linienlager(i).exclusionZ = ET_NONE
394:                 Case "1"
395:                     Linienlager(i).exclusionZ = ET_NOTENS
396:                 Case "2"
397:                     Linienlager(i).exclusionZ = ET_NOPRES
398:                 Case Else 'default Global
399:                     Linienlager(i).exclusionZ = ET_NONE
400:             End Select
401:             'Spalte K, Kommentar
402:             Linienlager(i).strRemark = Tabelle13.Cells(i + intOffset, 11)
403:             Next i
404:             'Übergabe an RFEM
405:             RFTop.rfPrepareModification
406:             RFTop.rfSetSuppLineArr intAnzahlZeilen, Linienlager(0)
407:             RFTop.rfFinishModification
408:             RFApp.rfUnlockLicence
409:             'Freigabe der Objekte
410:             Set RFPos = Nothing
411:             Set RFTop = Nothing
412:             Set RFApp = Nothing
413:         End Sub

```

In der Zeile 325: ist das Feld Linienlager() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.7 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_SUPPORT\_LINE.

RF\_SUPPORT\_LINE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
exclusionX	Ausschluss in X-Richtung	Benutzerdefinierter Aufzählungstyp RF_EXCLUSION_TYPE
exclusionY	Ausschluss in Y-Richtung	Benutzerdefinierter Aufzählungstyp RF_EXCLUSION_TYPE
exclusionZ	Ausschluss in Z-Richtung	Benutzerdefinierter Aufzählungstyp RF_EXCLUSION_TYPE
fAlongX	Stützung in X, Feder in X Wenn der Wert -1 übergeben wird, dann wird ein starres Auflager gesetzt. Bei positiven Zahlenwerten werden diese als Feder übernommen. [N/m <sup>2</sup> ]	Single
fAlongY	Stützung in Y, Feder in Y	Single
fAlongZ	Stützung in Z, Feder in Z	Single
fAroundX	Einspannung um X, Feder um X Wenn der Wert -1 übergeben wird, dann wird eine starre Einspannung gesetzt. Bei Positiven Zahlenwerten werden diese als Feder übernommen. [Nm/rad]	Single
fAroundY	Einspannung um Y, Feder um Y	Single
fAroundZ	Einspannung um Z, Feder um Z	Single
fRotAngle	Auflagerdrehung	Single
iNo	Nummer des Linienlagers	Long
isLocal	Lokales Linienlager, ja, nein	Boolean
strLineList	Linienliste	String
strRemark	Kommentar	String

Tabelle 3.21

Im benutzerdefinierten Aufzählungstyp RF\_EXCLUSION\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	ET_NONE	Kein Ausschluss
1	ET_NOTENS	Ausschluss bei Zug
2	ET_NOPRES	Ausschluss bei Druck

Tabelle 3.22

Die Methode rfSetSuppLineArr übergibt das Feld Linienlager() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetSuppLineArr mitgeteilt, wie viele Zeilen aus dem Feld Linienlager () übertragen werden sollen.

### 3.1.8 Flächenbettungen erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Flächenbettungen erzeugt werden.

Voraussetzungen sind, dass bereits die Flächen definiert sind. Die Flächenbettungen werden im Excel-Tabellenblatt 1.8 definiert und mit dem Makro CommandButton14\_Click() an RFEM übertragen.

```

414: Sub CommandButton14_Click()
415:   Dim intAnzahlZeilen As Integer
416:   Dim Flaechenbettung() As RF_FOUND_SURFACE
417:
418:   Set RFPos = GetObject(, "RFEM.Position")
419:   Set RFApp = RFPos.rfGetApplication
420:   RFApp.rfLockLicence
421:   Set RFTop = RFPos.rfGetTopology
422:   intAnzahlZeilen = Tabelle14.Cells(2, 3)
423:   ReDim Flaechenbettung(intAnzahlZeilen)
424:   intOffset = 4
425:   For i = 0 To intAnzahlZeilen - 1
426:     Flaechenbettung(i).iNo = i + 1
427:     'Spalte A, Liste der Flächen
428:     Flaechenbettung(i).strSurfaceList = Tabelle14.Cells(i +
intOffset, 1)
429:     'Spalte B, C-1,x
430:     Flaechenbettung(i).fC1x = Tabelle14.Cells(i + intOffset, 2)
431:     'Spalte C, C-1,y
432:     Flaechenbettung(i).fC1y = Tabelle14.Cells(i + intOffset, 3)
433:     'Spalte D, C-1,z
434:     Flaechenbettung(i).fC1z = Tabelle14.Cells(i + intOffset, 4)
435:     'Spalte E, C-2,x
436:     Flaechenbettung(i).fC2x = Tabelle14.Cells(i + intOffset, 5)
437:     'Spalte F, C-2,y
438:     Flaechenbettung(i).fC2y = Tabelle14.Cells(i + intOffset, 6)
439:     'Spalte G,Ausfall
440:     strZelle = Tabelle14.Cells(i + intOffset, 7)
441:     Select Case strZelle
442:       Case "Kein"
443:         Flaechenbettung(i).exclusion = ET_NONE
444:       Case "bei Zug"
445:         Flaechenbettung(i).exclusion = ET_NOTENS
446:       Case "bei Druck"
447:         Flaechenbettung(i).exclusion = ET_NOPRES
448:       Case Else 'default Kein Ausfall
449:         Flaechenbettung(i).exclusion = ET_NONE
450:     End Select
451:     'Spalte H, Kommentar
452:     Flaechenbettung(i).strRemark = Tabelle14.Cells(i + intOffset,
8)
453:   Next i
454:   'Übergabe an RFEM
455:   RFTop.rfPrepareModification
456:   RFTop.rfSetFoundSurfArr intAnzahlZeilen, Flaechenbettung(0)
457:   RFTop.rfFinishModification
458:   RFApp.rfUnlockLicence
459:   'Freigabe der Objekte
460:   Set RFPos = Nothing
461:   Set RFTop = Nothing
462:   Set RFApp = Nothing
463: End Sub

```

In der Zeile 416: ist das Feld Flaechenbettung() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.8 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_FOUND\_SURFACE.

RF\_FOUND\_SURFACE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
exclusion	Ausschluss in z-Richtung	Benutzerdefinierter Aufzählungstyp RF_EXCLUSION_TYPE
fC1x	Bettungsmodul in x-Richtung	Single
fC1y	Bettungsmodul in y-Richtung	Single
fC1z	Bettungsmodul in z-Richtung	Single
iNo	Nummer der Bettung	Long
strRemark	Kommentar	String
strSurfaceList	Liste der gebetteten Flächen	String

Tabelle 3.23

Im benutzerdefinierten Aufzählungstyp RF\_EXCLUSION\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	ET_NONE	Kein Ausschluss
1	ET_NOTENS	Ausschluss bei Zug
2	ET_NOPRES	Ausschluss bei Druck

Tabelle 3.24

Die Methode rfSetFoundSurfArr übergibt das Feld Flaechenbettung() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetFoundSurfArr mitgeteilt, wie viele Zeilen aus dem Feld Flaechenbettung () übertragen werden sollen.

### 3.1.9 Liniengelenke erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Liniengelenke erzeugt werden. Voraussetzungen sind, dass bereits die Linien und die Flächen definiert sind. Die Liniengelenke werden im Excel-Tabellenblatt 1.9 definiert und mit dem Makro CommandButton15\_Click() an RFEM übertragen.

```

464: Sub CommandButton15_Click()
465:   Dim intAnzahlZeilen As Integer
466:   Dim Liniengelenk() As RF_RELEASE_SURFACE
467:
468:   Set RFPos = GetObject(, "RFEM.Position")
469:   Set RFApp = RFPos.rfGetApplication
470:   RFApp.rfLockLicence
471:   Set RFTop = RFPos.rfGetTopology
472:   intAnzahlZeilen = Tabelle15.Cells(2, 3)
473:   ReDim Liniengelenk(intAnzahlZeilen)
474:   intOffset = 4
475:   For i = 0 To intAnzahlZeilen - 1
476:     Liniengelenk(i).iNo = i + 1
477:     'Spalte A, Linie-Nummer
478:     Liniengelenk(i).iLineNo = Tabelle15.Cells(i + intOffset, 1)
479:     'Spalte B, Fläche-Nummer
480:     Liniengelenk(i).iSurfaceNo = Tabelle15.Cells(i + intOffset, 2)
481:     'Spalte C, N
482:     Liniengelenk(i).n = Tabelle15.Cells(i + intOffset, 3)
483:     'Spalte D, V-2
484:     Liniengelenk(i).q2 = Tabelle15.Cells(i + intOffset, 4)
485:     'Spalte E, V-3

```

```

486:         Liniengelenk(i).q3 = Tabelle15.Cells(i + intOffset, 5)
487:         'Spalte F, M-T
488:         Liniengelenk(i).mt = Tabelle15.Cells(i + intOffset, 8)
489:         'Spalte G, M-2
490:         Liniengelenk(i).m2 = Tabelle15.Cells(i + intOffset, 7)
491:         'Spalte H, M-3
492:         Liniengelenk(i).m3 = Tabelle15.Cells(i + intOffset, 8)
493:         'Spalte I, Kommentar
494:         Liniengelenk(i).strRemark = Tabelle15.Cells(i + intOffset, 9)
495:     Next i
496:     'Übergabe an RFEM
497:     RFTop.rfPrepareModification
498:     RFTop.rfSetReleaseSurfArr intAnzahlZeilen, Liniengelenk(0)
499:     RFTop.rfFinishModification
500:     RFApp.rfUnlockLicence
501:     'Freigabe der Objekte
502:     Set RFPos = Nothing
503:     Set RFTop = Nothing
504:     Set RFApp = Nothing
505: End Sub
    
```

In der Zeile 466: ist das Feld Liniengelenk() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.9 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_RELEASE\_SURFACE.

RF\_RELEASE\_SURFACE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
iLineNo	Nummer der Linien mit dem Gelenk	Long
iNo	Nummer des Liniengelenks	Long
iSide	Seite des Liniengelenks	Benutzerdefinierter Aufzählungstyp RF_RELEASE_SIDE
iSurfaceNo	Flächennummer	Long
m2	Gelenk um die Linienachse 2, Feder um die Linienachse 2 Wenn der Wert -1 übergeben wird, dann wird dieser Freiheitsgrad als starr angesetzt, bei 0 gelenkig. Bei positiven Zahlenwerten werden diese als Feder übernommen. [N/rad/m]	Single
m3	Gelenk um die Linienachse 3, Feder um die Linienachse 3	Single
mt	Gelenk um die Linienachse 1, Feder um die Linienachse 1	Single
n	Gelenk in Richtung der Linienachse 1, Feder in Richtung der Linienachse 1 Wenn der Wert -1 übergeben wird, dann wird dieser Freiheitsgrad als starr angesetzt, bei 0 gelenkig. Bei positiven Zahlenwerten werden diese als Feder übernommen. [N/m <sup>2</sup> ]	Single
q2	Gelenk in Richtung der Linienachse 1, Feder in Richtung	Single

	der Liniennachse 1	
q3	Gelenk in Richtung der Liniennachse 1, Feder in Richtung der Liniennachse 1	Single
strRemark	Kommentar	String

Tabelle 3.25

Im benutzerdefinierten Aufzählungstyp RF\_RELEASE\_SIDE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	RS_NONE	Keine Definition der Seite
1	RS_LEFT	Linke Seite
2	RS_RIGHT	Rechte Seite

Tabelle 3.26

Die Methode rfSetReleaseSurfArr übergibt das Feld Liniengelenk() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetReleaseSurfArr mitgeteilt, wie viele Zeilen aus dem Feld Liniengelenk () übertragen werden sollen.

### 3.1.10 Stab-Querschnitte erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Stabquerschnitte erzeugt werden. Voraussetzungen sind, dass bereits die Materialien definiert sind. Die Stabquerschnitte werden im Excel-Tabellenblatt 1.12 definiert und mit dem Makro CommandButton8\_Click() an RFEM übertragen.

```

506: Sub CommandButton8_Click()
507:     Dim intAnzahlZeilen As Integer
508:     Dim Querschnitt() As RF_CROSS_SECTION
509:
510:     Set RFPos = GetObject(, "RFEM.Position")
511:     Set RFApp = RFPos.rfGetApplication
512:     RFApp.rfLockLicence
513:     Set RFTop = RFPos.rfGetTopology
514:     intAnzahlZeilen = Tabelle8.Cells(2, 3)
515:     ReDim Querschnitt(intAnzahlZeilen)
516:     intOffset = 4
517:     For i = 0 To intAnzahlZeilen - 1
518:         Querschnitt(i).iNo = i + 1
519:         'Spalte A auslesen, Materialnummer
520:         Querschnitt(i).iMaterialNo = Tabelle8.Cells(i + intOffset, 1)
521:         'Spalte B, Bezeichnung
522:         Querschnitt(i).strName = Tabelle8.Cells(i + intOffset, 2)
523:         'Spalte C, I-T
524:         Querschnitt(i).fIT = Tabelle8.Cells(i + intOffset, 3)
525:         'Spalte D, I-2
526:         Querschnitt(i).fI2 = Tabelle8.Cells(i + intOffset, 4)
527:         'Spalte E, I-3
528:         Querschnitt(i).fI3 = Tabelle8.Cells(i + intOffset, 5)
529:         'Spalte F, A-1
530:         Querschnitt(i).fA1 = Tabelle8.Cells(i + intOffset, 6)
531:         'Spalte G, A-2
532:         Querschnitt(i).fA2 = Tabelle8.Cells(i + intOffset, 7)
533:         'Spalte H, A-3
534:         Querschnitt(i).fA3 = Tabelle8.Cells(i + intOffset, 8)
535:         'Spalte F, Alpha
536:         Querschnitt(i).fAlpha = Tabelle8.Cells(i + intOffset, 9)
    
```

```

537:      Next i
538:      'Übergabe an RFEM
539:      RFTop.rfPrepareModification
540:      RFTop.rfSetCrossSectionArr intAnzahlZeilen, Querschnitt(0)
541:      RFTop.rfFinishModification
542:      RFApp.rfUnlockLicence
543:      'Freigabe der Objekte
544:      Set RFPos = Nothing
545:      Set RFTop = Nothing
546:      Set RFApp = Nothing
547:  End Sub
    
```

In der Zeile 508: ist das Feld Querschnitt() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.12 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_CROSS\_SECTION.

RF\_CROSS\_SECTION ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
fA1	Querschnittsfläche A-1 [m <sup>2</sup> ]	Single
fA2	Querschnittsfläche A-2 [m <sup>2</sup> ]	Single
fA3	Querschnittsfläche A-3 [m <sup>2</sup> ]	Single
fAlpha	Querschnittsdrehung [°]	Single
fI2	Flächenträgheitsmoment I-2	Single
fI3	Flächenträgheitsmoment I-3	Single
fIT	Flächenträgheitsmoment I-T	Single
iMaterialNo	Materialnummer	Single
iNo	Nummer des Querschnitts	Long
strName	Querschnittsname	String
strRemark	Kommentar	String

Tabelle 3.27

Wenn in strName ein Profilname aus der RFEM-Querschnittsbibliothek übergeben wird, dann können die Felder für Querschnittsflächen und Flächenträgheitsmomente leer bleiben. Es werden dann die Querschnittswerte aus der Bibliothek eingelesen.

Die Methode rfSetCrossSectionArr übergibt das Feld Querschnitt() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetCrossSectionArr mitgeteilt, wie viele Zeilen aus dem Feld Querschnitt () übertragen werden sollen.

### 3.1.11 Stabgelenke erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Stabgelenke erzeugt werden. Die Stabgelenke werden im Excel-Tabellenblatt 1.13 definiert und mit dem Makro CommandButton10\_Click() an RFEM übertragen.

```

548:  Sub CommandButton10_Click()
549:      Dim intAnzahlZeilen As Integer
550:      Dim Gelenk() As RF_RELEASE_MEMBER
551:
552:      Set RFPos = GetObject(, "RFEM.Position")
553:      Set RFApp = RFPos.rfGetApplication
554:      RFApp.rfLockLicence
555:      Set RFTop = RFPos.rfGetTopology
    
```

```

556:         intAnzahlZeilen = Tabelle10.Cells(2, 3)
557:         ReDim Gelenk(intAnzahlZeilen)
558:         intOffset = 4
559:         For i = 0 To intAnzahlZeilen - 1
560:             Gelenk(i).iNo = i + 1
561:             'Spalte A auslesen, N
562:             Gelenk(i).n = Tabelle10.Cells(i + intOffset, 1)
563:             'Spalte B, V-2
564:             Gelenk(i).q2 = Tabelle10.Cells(i + intOffset, 2)
565:             'Spalte C, V-3
566:             Gelenk(i).q3 = Tabelle10.Cells(i + intOffset, 3)
567:             'Spalte D, M-T
568:             Gelenk(i).mt = Tabelle10.Cells(i + intOffset, 4)
569:             'Spalte E, M-2
570:             Gelenk(i).m2 = Tabelle10.Cells(i + intOffset, 5)
571:             'Spalte F, M-3
572:             Gelenk(i).m3 = Tabelle10.Cells(i + intOffset, 6)
573:             'Spalte G, Kommentar
574:             Gelenk(i).strRemark = Tabelle10.Cells(i + intOffset, 7)
575:         Next i
576:         'Übergabe an RFEM
577:         RFTop.rfPrepareModification
578:         RFTop.rfSetReleaseMemberArr intAnzahlZeilen, Gelenk(0)
579:         RFTop.rfFinishModification
580:         RFApp.rfUnlockLicence
581:         'Freigabe der Objekte
582:         Set RFPos = Nothing
583:         Set RFTop = Nothing
584:         Set RFApp = Nothing
585:     End Sub

```

In der Zeile 550: ist das Feld Gelenk() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.13 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_RELEASE\_MEMBER.

RF\_RELEASE\_MEMBER ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
iNo	Nummer des Gelenks	Long
m2	Gelenk um die Stabachse 2, Feder um die Stabachse 2 Wenn der Wert -1 übergeben wird, dann wird dieser Freiheitsgrad als starr angesetzt, bei 0 gelenkig. Bei positiven Zahlenwerten werden diese als Feder übernommen. [Nm/rad]	Single
m3	Gelenk um die Stabachse 2, Feder um die Stabachse 2	Single
mt	Gelenk um die Stabachse 1, Feder um die Stabachse 1	Single
n	Gelenk in Richtung der Stabachse 1, Feder in Richtung der Stabachse 1 Wenn der Wert -1 übergeben wird, dann wird dieser Freiheitsgrad als starr angesetzt, bei 0 gelenkig. Bei positiven Zahlenwerten werden diese als Feder übernommen. [N/m]	Single
q2	Gelenk in Richtung der Stabachse 2, Feder in Richtung der	Single

	Stabachse 2	
q3	Gelenk in Richtung der Stabachse 3, Feder in Richtung der Stabachse 3	Single
strRemark	Kommentar	String

Tabelle 3.28

Die Methode `rfSetReleaseMemberArr` übergibt das Feld `Gelenk()` an RFEM. Mit der Variable `intAnzahlZeilen` wird `rfSetReleaseMemberArr` mitgeteilt, wie viele Zeilen aus dem Feld `Gelenk()` übertragen werden sollen.

### 3.1.12 Exzentrizitäten erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Stab-Exzentrizitäten erzeugt werden. Die Exzentrizitäten werden im Excel-Tabellenblatt 1.14 definiert und mit dem Makro `CommandButton11_Click()` an RFEM übertragen.

```

586: Sub CommandButton11_Click()
587: Dim intAnzahlZeilen As Integer
588: Dim Exzentrizitaet() As RF_ECCENTRICITY
589:
590: Set RFPos = GetObject(, "RFEM.Position")
591: Set RFApp = RFPos.rfGetApplication
592: RFApp.rfLockLicence
593: Set RFTop = RFPos.rfGetTopology
594: intAnzahlZeilen = Tabelle11.Cells(2, 3)
595: ReDim Exzentrizitaet(intAnzahlZeilen)
596: intOffset = 4
597: For i = 0 To intAnzahlZeilen - 1
598:   Exzentrizitaet(i).iNo = i + 1
599:   'Spalte A auslesen, Typ
600:   strZelle = Tabelle11.Cells(i + intOffset, 1)
601:   Select Case strZelle
602:     Case "Global"
603:       Exzentrizitaet(i).isLocal = False
604:     Case "Lokal"
605:       Exzentrizitaet(i).isLocal = True
606:     Case Else 'default Global
607:       Exzentrizitaet(i).isLocal = False
608:   End Select
609:   'Spalte B, e-x Anfang
610:   Exzentrizitaet(i).fE1x = Tabelle11.Cells(i + intOffset, 2)
611:   'Spalte C, e-y Anfang
612:   Exzentrizitaet(i).fE1y = Tabelle11.Cells(i + intOffset, 3)
613:   'Spalte D, e-z Anfang
614:   Exzentrizitaet(i).fE1z = Tabelle11.Cells(i + intOffset, 4)
615:   'Spalte E, e-x Ende
616:   Exzentrizitaet(i).fE2x = Tabelle11.Cells(i + intOffset, 5)
617:   'Spalte F, e-y Ende
618:   Exzentrizitaet(i).fE2y = Tabelle11.Cells(i + intOffset, 6)
619:   'Spalte G, e-z Ende
620:   Exzentrizitaet(i).fE2z = Tabelle11.Cells(i + intOffset, 7)
621:   'Spalte H, Kommentar
622:   Exzentrizitaet(i).strRemark = Tabelle11.Cells(i + intOffset,
8)
623: Next i
624:   'Übergabe an RFEM
625:   RFTop.rfPrepareModification
626:   RFTop.rfSetEccentricityArr intAnzahlZeilen, Exzentrizitaet(0)
627:   RFTop.rfFinishModification
628:   RFApp.rfUnlockLicence
629:   'Freigabe der Objekte
630:   Set RFPos = Nothing

```

```
631:      Set RFTop = Nothing
632:      Set RFApp = Nothing
633:      End Sub
```

In der Zeile 588: ist das Feld Exzentrizitaet () definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.14 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_ECCENTRICITY.

RF\_ECCENTRICITY ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
fE1x	Exzentrizität in X-Richtung am Anfangsknoten des Stabes	Single
fE1y	Exzentrizität in Y-Richtung am Anfangsknoten des Stabes	Single
fE1z	Exzentrizität in Z-Richtung am Anfangsknoten des Stabes	Single
fE2x	Exzentrizität in X-Richtung am Endknoten des Stabes	Single
fE2y	Exzentrizität in Y-Richtung am Endknoten des Stabes	Single
fE2z	Exzentrizität in Z-Richtung am Endknoten des Stabes	Single
iNo	Nummer der Exzentrizität	Single
isLocal	Lokale Exzentrizität, ja, nein	Boolean
strRemark	Kommentar	String

Tabelle 3.29

Die Methode rfSetEccentricityArr übergibt das Feld Exzentrizitaet() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetEccentricityArr mitgeteilt, wie viele Zeilen aus dem Feld Exzentrizitaet() übertragen werden sollen.

### 3.1.13 Stabteilungen erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Stabteilungen erzeugt werden. Die Stabteilungen werden im Excel-Tabellenblatt 1.15 definiert und mit dem Makro CommandButton12\_Click() an RFEM übertragen.

```
634:      Sub CommandButton12_Click()
635:          Dim intAnzahlZeilen As Integer
636:          Dim Stabteilung() As RF_MEMBER_PART
637:
638:          Set RFPos = GetObject(, "RFEM.Position")
639:          Set RFApp = RFPos.rfGetApplication
640:          RFApp.rfLockLicence
641:          Set RFTop = RFPos.rfGetTopology
642:          intAnzahlZeilen = Tabelle12.Cells(2, 3)
643:          ReDim Stabteilung(intAnzahlZeilen)
644:          intOffset = 4
645:          For i = 0 To intAnzahlZeilen - 1
646:              Stabteilung(i).iNo = i + 1
647:              'Spalte A, Anzahl der Punkte
648:              Stabteilung(i).iCount = Tabelle12.Cells(i + intOffset, 1)
649:              'Spalte B, x-1
```

```

650:      Stabteilung(i).distanceArr(0) = Tabelle12.Cells(i + intOffset,
        2)
651:      'Spalte C, x-2
652:      Stabteilung(i).distanceArr(1) = Tabelle12.Cells(i + intOffset,
        3)
653:      'Spalte D, x-3
654:      Stabteilung(i).distanceArr(2) = Tabelle12.Cells(i + intOffset,
        4)
655:      'Spalte E, x-4
656:      Stabteilung(i).distanceArr(3) = Tabelle12.Cells(i + intOffset,
        5)
657:      'Spalte F, x-5
658:      Stabteilung(i).distanceArr(4) = Tabelle12.Cells(i + intOffset,
        6)
659:      'Spalte G, x-6
660:      Stabteilung(i).distanceArr(5) = Tabelle12.Cells(i + intOffset,
        7)
661:      'Spalte H, x-7
662:      Stabteilung(i).distanceArr(6) = Tabelle12.Cells(i + intOffset,
        8)
663:      'Spalte I, x-8
664:      Stabteilung(i).distanceArr(7) = Tabelle12.Cells(i + intOffset,
        9)
665:      'Spalte J, x-9
666:      Stabteilung(i).distanceArr(8) = Tabelle12.Cells(i + intOffset,
        10)
667:      'Spalte K, Kommentar
668:      Stabteilung(i).strRemark = Tabelle12.Cells(i + intOffset, 11)
669:      Next i
670:      'Übergabe an RFEM
671:      RFTop.rfPrepareModification
672:      RFTop.rfSetMemberPartArr intAnzahlZeilen, Stabteilung(0)
673:      RFTop.rfFinishModification
674:      RFApp.rfUnlockLicence
675:      'Freigabe der Objekte
676:      Set RFPos = Nothing
677:      Set RFTop = Nothing
678:      Set RFApp = Nothing
679:      End Sub
    
```

In der Zeile 636: ist das Feld Stabteilung() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.15 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_MEMBER\_PART.

RF\_MEMBER\_PART ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
distanceArr(0 to 8)	Feld mit 9 Elementen für den relativen Abstand der Stabteilung von Stabanfang	Single
iCount	Anzahl der Teilungen	Long
iNo	Nummer der Stabteilung	Long
strRemark	Kommentar	String

Tabelle 3.30

Die Methode rfSetMemberPartArr übergibt das Feld Stabteilung() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetMemberPartArr mitgeteilt, wie viele Zeilen aus dem Feld Stabteilung() übertragen werden sollen.

### 3.1.14 Stäbe erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Stäbe erzeugt werden. Voraussetzungen sind, dass bereits die Stab-Querschnitte definiert sind. Die Stäbe werden im Excel-Tabellenblatt 1.16 definiert und mit dem Makro CommandButton9\_Click() an RFEM übertragen.

```

680: Sub CommandButton9_Click()
681: Dim intAnzahlZeilen As Integer
682: Dim Stab() As RF_MEMBER
683:
684: Set RFPos = GetObject(, "RFEM.Position")
685: Set RFApp = RFPos.rfGetApplication
686: RFApp.rfLockLicence
687: Set RFTop = RFPos.rfGetTopology
688: intAnzahlZeilen = Tabelle9.Cells(2, 3)
689: ReDim Stab(intAnzahlZeilen)
690: intOffset = 4
691: For i = 0 To intAnzahlZeilen - 1
692:   Stab(i).iNo = i + 1
693:   'Spalte A auslesen, Liniennummer
694:   Stab(i).iCurveNo = Tabelle9.Cells(i + intOffset, 1)
695:   'Spalte B, Stabtyp
696:   strZelle = Tabelle9.Cells(i + intOffset, 2)
697:   Select Case strZelle
698:     Case "Balkenstab"
699:       Stab(i).Type = ET_BEAM
700:     Case "Seilstab"
701:       Stab(i).Type = ET_CABLE
702:     Case "Druckstab"
703:       Stab(i).Type = ET_COMPRESSION
704:     Case "Nullstab"
705:       Stab(i).Type = ET_DUMMY
706:     Case "Knickstab"
707:       Stab(i).Type = ET_EULER
708:     Case "Rippe"
709:       Stab(i).Type = ET_RIB
710:     Case "Kopplung Gelenk-Gelenk"
711:       Stab(i).Type = ET_RIGID_TYPE_HH
712:     Case "Kopplung Gelenk-Fest"
713:       Stab(i).Type = ET_RIGID_TYPE_HR
714:     Case "Kopplung Fest-Gelenk"
715:       Stab(i).Type = ET_RIGID_TYPE_RH
716:     Case "Kopplung Fest-Fest"
717:       Stab(i).Type = ET_RIGID_TYPE_RR
718:     Case "Zugstab"
719:       Stab(i).Type = ET_TENSION
720:     Case "Fachwerkstab"
721:       Stab(i).Type = ET_TRUSS_FALSE
722:     Case "Fachwerkstab nur N"
723:       Stab(i).Type = ET_TRUSS_TRUE
724:     Case Else 'default Balken
725:       Stab(i).Type = ET_BEAM
726:   End Select
727:   'Spalte C, Querschnitt Anfang
728:   Stab(i).iBeginCrossSectionNo = Tabelle9.Cells(i + intOffset, 3)
729:   'Spalte D, Querschnitt Ende
730:   Stab(i).iEndCrossSectionNo = Tabelle9.Cells(i + intOffset, 4)
731:   'Spalte F, Drehungstyp fest auf Winkel
732:   Stab(i).rotType = BY_ANGLE
733:   'Spalte G, Drehwinkel
734:   Stab(i).fRotation = Tabelle9.Cells(i + intOffset, 7)
735:   'Spalte H, Gelenk Anfang
736:   Stab(i).iBeginReleaseNo = Tabelle9.Cells(i + intOffset, 8)
737:   'Spalte I, Gelenk Ende
738:   Stab(i).iEndReleaseNo = Tabelle9.Cells(i + intOffset, 9)
739:   'Spalte J, Exzentrizität
740:   Stab(i).iEccentricityNo = Tabelle9.Cells(i + intOffset, 10)
741:   'Spalte K, Teilung
742:   Stab(i).iPartitionNo = Tabelle9.Cells(i + intOffset, 11)
743:   'Spalte L, Kommentar
744:   Stab(i).strRemark = Tabelle9.Cells(i + intOffset, 12)
745: Next i
746: 'Übergabe an RFEM
747: RFTop.rfPrepareModification
748: RFTop.rfSetMemberArr intAnzahlZeilen, Stab()
749: RFTop.rfFinishModification

```

```

750:      RFApp.rfUnlockLicence
751:      'Freigabe der Objekte
752:      Set RFPos = Nothing
753:      Set RFTop = Nothing
754:      Set RFApp = Nothing
755:      End Sub
    
```

In der Zeile 682: ist das Feld Stab() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.16 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_MEMBER.

RF\_MEMBER ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
fRotation	Stabdrehwinkel [°]	Single
iBeginCrossSectionNo	Querschnittsnummer am Stabanfang	Long
iBeginReleaseNo	Nummer des Stabendgelenktyps am Stabanfang	Long
iCurveNo	Nummer der Linien, auf der der Stab definiert werden soll	Long
iEccentricityNo	Nummer Exzentrizität	Long
iEndCrossSectionNo	Querschnittsnummer am Stabende	Long
iEndReleaseNo	Nummer des Stabendgelenktyps am Stabende	Long
iFoundNo	Nummer der elastischen Stabbettung	Long
iNo	Nummer des Stabes	Long
iPartitionNo	Nummer der Stabteilung	Long
iRibNo	Nummer der Rippe	Long
rotNode	Hilfspunkt für die Stabdrehung	Benutzerdefinierter Typ RF_MEMBER_ROTATION_NODE
rotType	Typ der Stabdrehung	Benutzerdefinierter Aufzählungstyp RF_MEMBER_ROTATION_TYPE
strRemark	Kommentar	String
type	Stabtyp	Benutzerdefinierter Aufzählungstyp RF_MEMBER_TYPE

Tabelle 3.31

RF\_MEMBER\_ROTATION\_NODE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
axis	Drehungstyp	Benutzerdefinierter Aufzählungstyp RF_NODE_ROTATION_AXIS
x	X-Koordinate des Hilfspunktes	Single
y	Y-Koordinate des Hilfspunktes	Single
z	Z-Koordinate des Hilfspunktes	Single

Tabelle 3.32

Im benutzerdefinierten Aufzählungstyp RF\_MEMBER\_ROTATION\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	BY_ANGLE	Drehung durch Drehwinkel
1	BY_NODE	Drehung durch Punkt

Tabelle 3.33

Im benutzerdefinierten Aufzählungstyp RF\_MEMBER\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	MT_BEAM	Balken
1	MT_TRUSS_FALSE	Fachwerkstab
2	MT_TRUSS_TRUE	Fachwerkstab (nur N)
3	MT_TENSION	Zugstab
4	MT_COMPRESSION	Druckstab
5	MT_EULER	Knickstab
6	MT_CABLE	Seil
7	MT_DUMMY	Nullstab
8	MT_RIGID_TYPE_RR	Starre Kopplung fest - fest
9	MT_RIGID_TYPE_RH	Starre Kopplung fest - gelenkig
10	MT_RIGID_TYPE_HH	Starre Kopplung gelenkig - gelenkig
11	MT_RIGID_TYPE_HR	Starre Kopplung gelenkig - fest
12	MT_RIB	Rippe

Tabelle 3.34

Im benutzerdefinierten Aufzählungstyp RF\_NODE\_ROTATION\_AXIS sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	AXIS_2	Ausrichtung der Stabachse 2 auf den Drehpunkt
1	AXIS_3	Ausrichtung der Stabachse 3 auf den Drehpunkt

Tabelle 3.35

Die Methode rfSetMemberArr übergibt das Feld Stab() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetMemberArr mitgeteilt, wie viele Zeilen aus dem Feld Stab() übertragen werden sollen.

### 3.1.15 Stabbettungen erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Stabbettungen erzeugt werden. Die Stabbettungen werden im Excel-Tabellenblatt 1.17 definiert und mit dem Makro `CommandButton16_Click()` an RFEM übertragen.

```

756: Sub CommandButton16_Click()
757:   Dim intAnzahlZeilen As Integer
758:   Dim Stabbettung() As RF_FOUND_MEMBER
759:
760:   Set RFPos = GetObject(, "RFEM.Position")
761:   Set RFApp = RFPos.rfGetApplication
762:   RFApp.rfLockLicence
763:   Set RFTop = RFPos.rfGetTopology
764:   intAnzahlZeilen = Tabelle16.Cells(2, 3)
765:   ReDim Stabbettung(intAnzahlZeilen)
766:   intOffset = 4
767:   For i = 0 To intAnzahlZeilen - 1
768:     Stabbettung(i).iNo = i + 1
769:     'Spalte A, Stäbe
770:     Stabbettung(i).strMemberList = Tabelle16.Cells(i+intOffset, 1)
771:     'Spalte B, C-1,1
772:     Stabbettung(i).fC11 = Tabelle16.Cells(i + intOffset, 2)
773:     'Spalte C, C-1,2
774:     Stabbettung(i).fC12 = Tabelle16.Cells(i + intOffset, 3)
775:     'Spalte D, C-1,3
776:     Stabbettung(i).fC13 = Tabelle16.Cells(i + intOffset, 4)
777:     'Spalte E, C-2,1
778:     Stabbettung(i).fC21 = Tabelle16.Cells(i + intOffset, 5)
779:     'Spalte F, C-2,2
780:     Stabbettung(i).fC22 = Tabelle16.Cells(i + intOffset, 6)
781:     'Spalte G, C-2,2
782:     Stabbettung(i).fC23 = Tabelle16.Cells(i + intOffset, 7)
783:     'Spalte H, C-Phi
784:     Stabbettung(i).fCfi = Tabelle16.Cells(i + intOffset, 8)
785:     'Spalte I, Ausfall
786:     strZelle = Tabelle16.Cells(i + intOffset, 9)
787:     Select Case strZelle
788:       Case "Kein"
789:         Stabbettung(i).exclusion = ET_NONE
790:       Case "bei Zug"
791:         Stabbettung(i).exclusion = ET_NOTENS
792:       Case "bei Druck"
793:         Stabbettung(i).exclusion = ET_NOPRES
794:       Case Else 'default Kein Ausfall
795:         Stabbettung(i).exclusion = ET_NONE
796:     End Select
797:     'Spalte J, Kommentar
798:     Stabbettung(i).strRemark = Tabelle16.Cells(i + intOffset, 10)
799:   Next i
800:   'Übergabe an RFEM
801:   RFTop.rfPrepareModification
802:   RFTop.rfSetFoundMemberArr intAnzahlZeilen, Stabbettung(0)
803:   RFTop.rfFinishModification
804:   RFApp.rfUnlockLicence
805:   'Freigabe der Objekte
806:   Set RFPos = Nothing
807:   Set RFTop = Nothing
808:   Set RFApp = Nothing
809: End Sub

```

In der Zeile 758: ist das Feld `Stabbettung()` definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.17 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ `RF_FOUND_MEMBER`.

`RF_FOUND_MEMBER` ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
exclusion	Ausfall der Bettung	Benutzerdefinierter Aufzählungstyp <code>RF_EXCLUSION_TYPE</code>
fC11	Bettungskonstante C-1,1 [N/m <sup>2</sup> ]	Single
fC12	Bettungskonstante C-1,2 [N/m <sup>2</sup> ]	Single

fC13	Bettungskonstante C-1,3 [N/m <sup>2</sup> ]	Single
fC21	Bettungskonstante C-2,1 [N]	Single
fC22	Bettungskonstante C-2,2 [N]	Single
fC23	Bettungskonstante C-2,3 [N]	Single
fCfi	Bettungskonstante C-Phi [N/rad/m]	Single
iNo	Nummer der Bettung	Long
strMemberList	Liste der gebetteten Stäbe	String
strRemark	Kommentar	String

Tabelle 3.36

Im benutzerdefinierten Aufzählungstyp RF\_EXCLUSION\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	ET_NONE	Kein Ausfall
1	ET_NOPRES	Ausfall bei Druck
2	ET_NOTENS	Ausfall bei Zug

Tabelle 3.37

Die Methode rfSetFoundMemberArr übergibt das Feld Stabbettung() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetFoundMemberArr mitgeteilt, wie viele Zeilen aus dem Feld Stabbettung() übertragen werden sollen.

### 3.1.16 Stab-Nichtlinearitäten erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Stab-Nichtlinearitäten erzeugt werden. Voraussetzungen sind, dass bereits die Stäbe definiert sind. Die Stab-Nichtlinearitäten werden im Excel-Tabellenblatt 1.19 definiert und mit dem Makro CommandButton17\_Click() an RFEM übertragen.

```

810: Sub CommandButton17_Click()
811:     Dim intAnzahlZeilen As Integer
812:     Dim StabNichtlinearitaet() As RF_NL
813:
814:     Set RFPos = GetObject(, "RFEM.Position")
815:     Set RFApp = RFPos.rfGetApplication
816:     RFApp.rfLockLicence
817:     Set RFTop = RFPos.rfGetTopology
818:     intAnzahlZeilen = Tabelle17.Cells(2, 3)
819:     ReDim StabNichtlinearitaet(intAnzahlZeilen)
820:     intOffset = 4
821:     For i = 0 To intAnzahlZeilen - 1
822:         StabNichtlinearitaet(i).iNo = i + 1
823:         'Spalte A, Stabliste
824:         StabNichtlinearitaet(i).strMemberList = Tabelle17.Cells(i +
intOffset, 1)
825:         'Spalte B, Typ
826:         strZelle = Tabelle17.Cells(i + intOffset, 2)
827:         Select Case strZelle
828:             Case "Ausfall bei Zug"
829:                 StabNichtlinearitaet(i).Type = NL_ETENS
830:             Case "Ausfall bei Druck"

```

```

831:             StabNichtlinearitaet(i).Type = NL_EPRESS
832:         Case "Ausfall bei Zug mit Schlupf"
833:             StabNichtlinearitaet(i).Type = NL_ETENS_GAP
834:         Case "Ausfall bei Druck mit Schlupf"
835:             StabNichtlinearitaet(i).Type = NL_EPRESS_GAP
836:         Case "Schlupf"
837:             StabNichtlinearitaet(i).Type = NL_GAP_SIDES
838:         Case "Reißen bei Zug"
839:             StabNichtlinearitaet(i).Type = NL_ETENS_S
840:         Case "Fließen bei Zug"
841:             StabNichtlinearitaet(i).Type = NL_ETENS_P
842:         Case "Reißen bei Druck"
843:             StabNichtlinearitaet(i).Type = NL_EPRESS_S
844:         Case "Fließen bei Druck"
845:             StabNichtlinearitaet(i).Type = NL_EPRESS_P
846:         Case "Reißen"
847:             StabNichtlinearitaet(i).Type = NL_LIM_S
848:         Case "Fließen"
849:             StabNichtlinearitaet(i).Type = NL_LIM_P
850:         Case "Plastisches Gelenk"
851:             StabNichtlinearitaet(i).Type = NL_PLAST_FORCE
852:         Case Else 'default kein Ausfall
853:             StabNichtlinearitaet(i).Type = NL_NO
854:     End Select
855:     'Spalte C, Parameter1
856:     StabNichtlinearitaet(i).fParam1 = Tabelle17.Cells(i +
intOffset, 3)
857:     'Spalte D, Parameter2
858:     StabNichtlinearitaet(i).fParam2 = Tabelle17.Cells(i +
intOffset, 4)
859:     'Spalte E, Parameter3
860:     StabNichtlinearitaet(i).fParam3 = Tabelle17.Cells(i +
intOffset, 5)
861:     'Spalte F, Parameter4
862:     StabNichtlinearitaet(i).fParam4 = Tabelle17.Cells(i +
intOffset, 6)
863:     'Spalte G, Parameter5
864:     StabNichtlinearitaet(i).fParam5 = Tabelle17.Cells(i +
intOffset, 7)
865:     'Spalte H, Parameter6
866:     StabNichtlinearitaet(i).fParam6 = Tabelle17.Cells(i +
intOffset, 8)
867:     'Spalte I, Kommentar
868:     StabNichtlinearitaet(i).strRemark = Tabelle17.Cells(i +
intOffset, 9)
869:     Next i
870:     'Übergabe an RFEM
871:     RFTop.rfPrepareModification
872:     RFTop.rfSetNLarr intAnzahlZeilen, StabNichtlinearitaet(0)
873:     RFTop.rfFinishModification
874:     RFApp.rfUnlockLicence
875:     'Freigabe der Objekte
876:     Set RFPos = Nothing
877:     Set RFTop = Nothing
878:     Set RFApp = Nothing
879: End Sub

```

In der Zeile 812: ist das Feld StabNichtlinearitaet() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.19 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_NL.

RF\_NL ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
fParam1	Parameter 1, die Bedeutung des Parameters ist vom Typ der Nichtlinearität abhängig	Single
fParam2	Parameter 2	Single
fParam3	Parameter 3	Single

fParam4	Parameter 4	Single
fParam5	Parameter 5	Single
fParam6	Parameter 6	Single
iNo	Nummer der Nichtlinearität	Long
strMemberList	Stabliste	String
strRemark	Kommentar	String
type	Typ der Nichtlinearität	Benutzerdefinierter Aufzählungstyp RF_NL_TYPE

Tabelle 3.38

Im benutzerdefinierten Aufzählungstyp RF\_NL\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	NL_NO	Kein Ausfall
1	NL_ETENS	Ausfall bei Zug
2	NL_EPRESS	Ausfall bei Druck
3	NL_ETENS_GAP	Ausfall bei Zug mit Schlupf
4	NL_EPRESS_GAP	Ausfall bei Druck mit Schlupf
5	NL_GAP_SIDES	Schlupf
6	NL_ETENS_S	Reißen bei Zug
7	NL_ETENS_P	Fließen bei Zug
8	NL_EPRESS_S	Reißen bei Druck
9	NL_EPRESS_P	Fließen bei Druck
10	NL_LIM_S	Reißen
11	NL_LIM_P	Fließen
12	NL_PLAST_RELEASE	Plastisches Gelenk

Tabelle 3.39

Die Methode rfSetNLArr übergibt das Feld StabNichtlinearitaet() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetNLArr mitgeteilt, wie viele Zeilen aus dem Feld StabNichtlinearitaet() übertragen werden sollen.

### 3.1.17 FE-Netzverdichtungen erzeugen

Die grundsätzliche Vorgehensweise bei der Programmierung des Makros wurde in 3.1.1 Punkte erzeugen (siehe S. 7) ausführlich erläutert. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen FE-Netzverdichtungen erzeugt werden. Voraussetzungen sind, dass bereits Punkte, Linien oder Flächen definiert sind. Die FE-Netzverdichtungen werden im Excel-Tabellenblatt 1.20 definiert und mit dem Makro CommandButton34\_Click() an RFEM übertragen.

```

880: Sub CommandButton34_Click()
881:     Dim intAnzahlZeilen As Integer
882:     Dim Verdichtung() As RF_MESH
883:     Dim strZelle As String
884: 
```

```

885:      Set RFPos = GetObject(, "RFEM.Position")
886:      Set RFApp = RFPos.rfGetApplication
887:      RFApp.rfLockLicence
888:      Set RFTop = RFPos.rfGetTopology
889:      intAnzahlZeilen = Tabelle34.Cells(2, 3)
890:      ReDim Verdichtung(intAnzahlZeilen)
891:      intOffset = 4
892:      For i = 0 To intAnzahlZeilen - 1
893:          Verdichtung(i).iNo = i + 1
894:          'Spalte A auslesen, Koordinatensystem
895:          strZelle = Tabelle34.Cells(i + intOffset, 1)
896:          Select Case strZelle
897:              Case "Punkte"
898:                  Verdichtung(i).Type = MT_NODE
899:                  Verdichtung(i).fSizeE = Tabelle34.Cells(i + intOffset,
900: 3)
901:                  Verdichtung(i).fSizeI = Tabelle34.Cells(i + intOffset,
902: 4)
903:                  Verdichtung(i).fSizeO = Tabelle34.Cells(i + intOffset,
904: 5)
905:              Case "Linien durch FE-Länge"
906:                  Verdichtung(i).Type = MT_LINE_LENGTH
907:                  Verdichtung(i).fSizeE = Tabelle34.Cells(i + intOffset,
908: 3)
909:              Case "Linien durch Teilung"
910:                  Verdichtung(i).Type = MT_LINE_DIVISION
911:                  Verdichtung(i).iNum = Tabelle34.Cells(i + intOffset,
912: 6)
913:              Case "Fläche"
914:                  Verdichtung(i).Type = MT_SURFACE
915:                  Verdichtung(i).fSizeE = Tabelle34.Cells(i + intOffset,
916: 3)
917:              Case Else 'default Punkt
918:                  Verdichtung(i).Type = MT_NODE
919:                  Verdichtung(i).fSizeE = Tabelle34.Cells(i + intOffset,
920: 3)
921:                  Verdichtung(i).fSizeI = Tabelle34.Cells(i + intOffset,
922: 4)
923:                  Verdichtung(i).fSizeO = Tabelle34.Cells(i + intOffset,
924: 5)
925:          End Select
926:          'Spalte B auslesen, Liste
927:          Verdichtung(i).strObjectList = Tabelle34.Cells(i + intOffset,
928: 2)
929:          'Spalte G auslesen, Kommentar
930:          Verdichtung(i).strRemark = Tabelle34.Cells(i + intOffset, 7)
931:      Next i
932:      'Übergabe an RFEM
933:      RFTop.rfPrepareModification
934:      RFTop.rfSetMeshArr intAnzahlZeilen, Verdichtung(0)
935:      RFTop.rfFinishModification
936:      RFApp.rfUnlockLicence
937:      'Freigabe der Objekte
938:      Set RFPos = Nothing
939:      Set RFTop = Nothing
940:      Set RFApp = Nothing
941:  End Sub

```

In der Zeile 882: ist das Feld Verdichtung() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 1.20 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_MESH.

RF\_MESH ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
fSizeE	Radius der Verdichtung, wenn eine Punkterdichtung definiert wird oder die angestrebte FE-Netzlänge, wenn eine Verdichtung auf einer Fläche oder entlang einer Linie definiert wird	Single

fSizeI	Angestrebte FE-Netzlänge bei einer Punktverdichtung im Zentrum	Single
fSizeO	Angestrebte FE-Netzlänge bei einer Punktverdichtung am Rand	Single
iNo	Nummer der FE-Netzverdichtung	Long
iNum	Anzahl der Teilungen, wenn eine FE-Netzverdichtung an einer Linie durch Teilung definiert wird	Long
strObjectList	Liste der Punkte, Linien oder Flächen	String
strRemark	Kommentar	String
type	Typ der FE-Netzverdichtung	Benutzerdefinierter Aufzählungstyp RF_MESH_TYPE

Tabelle 3.40

Im benutzerdefinierten Aufzählungstyp RF\_MESH\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	MT_NODE	FE-Netzverdichtung an einem Punkt
1	MT_LINE_LENGTH	FE-Netzverdichtung an einer Linie definiert durch eine FE-Netzlänge
2	MT_LINE_DIVISION	FE-Netzverdichtung an einer Linie definiert durch eine Teilung
3	MT_SURFACE	FE-Netzverdichtung auf einer Fläche

Tabelle 3.41

Die Methode rfSetMeshArr übergibt das Feld Verdichtung() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetMeshArr mitgeteilt, wie viele Zeilen aus dem Feld Verdichtung() übertragen werden sollen.

## 3.2 Lastfälle, Lastfallgruppen, Lastfallkombinationen

### 3.2.1 Lastfälle erzeugen

Das Anlegen von Lastfällen ähnelt sehr dem Anlegen von Strukturdaten. Im Kapitel 3.1.1 (S. 7) wurde das ausführlich dargestellt. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Lastfälle erzeugt werden. Die Lastfälle werden im Excel-Tabellenblatt Lastfälle definiert und mit dem Makro CommandButton18\_Click() an RFEM übertragen.

```

932: Sub CommandButton18_Click()
933:     Dim intAnzahlZeilen As Integer
934:     Dim Lastfall() As RF_LOAD_CASE
935:
936:     Set RFPos = GetObject(, "RFEM.Position")
937:     Set RFApp = RFPos.rfGetApplication
938:     RFApp.rfLockLicence
    
```

```

939:      Set RFTop = RFPos.rfGetTopology
940:      intAnzahlZeilen = Tabelle18.Cells(2, 3)
941:      ReDim Lastfall(intAnzahlZeilen)
942:      intOffset = 4
943:      For i = 0 To intAnzahlZeilen - 1
944:          'Spalte A, Lastfallnummer
945:          Lastfall(i).iNo = Tabelle18.Cells(i + intOffset, 1)
946:          'Spalte B, Bezeichnung
947:          Lastfall(i).strDescription = Tabelle18.Cells(i + intOffset, 2)
948:          'Spalte C, LF-Faktor
949:          Lastfall(i).fLCFactor = Tabelle18.Cells(i + intOffset, 3)
950:          'Spalte D, Typ
951:          strZelle = Tabelle18.Cells(i + intOffset, 4)
952:          Select Case strZelle
953:              Case "Ständig"
954:                  Lastfall(i).Type = LCT_PERMANENT
955:              Case "Veränderlich"
956:                  Lastfall(i).Type = LCT_VARIABLE
957:              Case "Außergewöhnlich"
958:                  Lastfall(i).Type = LCT_EXCEPTION
959:              Case "Entlastend ständig"
960:                  Lastfall(i).Type = LCT_RELIEVE_PERMANENT
961:              Case "Entlastend veränderlich"
962:                  Lastfall(i).Type = LCT_RELIEVE_VARIABLE
963:              Case "Imperfektion"
964:                  Lastfall(i).Type = LCT_IMPERFECTION
965:              Case Else 'default Veränderlich
966:                  Lastfall(i).Type = LCT_VARIABLE
967:          End Select
968:          'Spalte E, Eigengewicht berücksichtigen
969:          strZelle = Tabelle18.Cells(i + intOffset, 5)
970:          Select Case strZelle
971:              Case "Ja"
972:                  Lastfall(i).bDWRespect = True
973:              Case "Nein"
974:                  Lastfall(i).bDWRespect = False
975:              Case Else 'default Nein
976:                  Lastfall(i).bDWRespect = False
977:          End Select
978:          'Spalte F, Richtung
979:          strZelle = Tabelle18.Cells(i + intOffset, 6)
980:          Select Case strZelle
981:              Case "X"
982:                  Lastfall(i).DWDDir = LD_X
983:              Case "Y"
984:                  Lastfall(i).DWDDir = LD_Y
985:              Case "Z"
986:                  Lastfall(i).DWDDir = LD_Z
987:              Case Else 'default Nein
988:                  Lastfall(i).DWDDir = LD_X
989:          End Select
990:          'Spalte G, Faktor
991:          Lastfall(i).fDWFactor = Tabelle18.Cells(i + intOffset, 7)
992:          'Spalte G, Kommentar
993:          Lastfall(i).strRemark = Tabelle18.Cells(i + intOffset, 8)
994:      Next i
995:      'Übergabe an RFEM
996:      RFPos.rfGetLoads.rfPrepareModification
997:      RFPos.rfGetLoads.rfSetLoadCaseArr intAnzahlZeilen, Lastfall(0)
998:      RFPos.rfGetLoads.rfFinishModification
999:      RFApp.rfUnlockLicence
1000:      'Freigabe der Objekte
1001:      Set RFPos = Nothing
1002:      Set RFTop = Nothing
1003:      Set RFApp = Nothing
1004:  End Sub

```

In der Zeile 934: ist das Feld Lastfall() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt Lastfälle gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_CASE.

RF\_LOAD\_CASE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bDWRespect	Eigengewicht berücksichtigen, ja, nein	Boolean
DWDir	Richtung des Eigengewichtes	Benutzerdefinierter Aufzählungstyp RF_LOAD_CASE_DIRECTION
fDWFactor	Faktor des Eigengewichtes	Single
fLCFactor	Lastfall-Faktor	Single
iNo	Lastfallnummer	Long
strDescription	Beschreibung	String
strRemark	Kommentar	String
type	Überlagerungsart	Benutzerdefinierter Aufzählungstyp RF_LOAD_CASE_TYPE

Tabelle 3.42

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_CASE\_DIRECTION sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LCD_X	Richtung X
1	LCD_Y	Richtung Y
2	LCD_Z	Richtung Z

Tabelle 3.43

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_CASE\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LCT_PERMANENT	Ständig
1	LCT_VARIABLE	Veränderlich
2	LCT_EXCEPTION	Außergewöhnlich
3	LCT_IMPERFECTION	Imperfektion
4	LCT_RELIEVE_PERMANENT	Entlastend ständig
5	LCT_RELIEVE_VARIABLE	Entlastend veränderlich

Tabelle 3.44

Die Methode rfSetLoadCaseArr übergibt das Feld Lastfall() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetLoadCaseArr mitgeteilt, wie viele Zeilen aus dem Feld Lastfall() übertragen werden sollen.

### 3.2.2 Lastfallgruppen erzeugen

Das Anlegen von Lastfallgruppen ähnelt sehr dem Anlegen von Strukturdaten. Im Kapitel 3.1.1 (S. 7) wurde das ausführlich dargestellt. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Lastfallgruppen erzeugt werden. Die Lastfallgruppen werden im Excel-Tabellenblatt Lastfallgruppen definiert und mit dem Makro CommandButton19\_Click() an RFEM übertragen.

```

1005: Sub CommandButton19_Click()
1006:   Dim intAnzahlZeilen As Integer
1007:   Dim Lastfallgruppe() As RF_LOAD_GROUP
1008:
1009:   Set RFPos = GetObject(, "RFEM.Position")
1010:   Set RFApp = RFPos.rfGetApplication
1011:   RFApp.rfLockLicence
1012:   Set RFTop = RFPos.rfGetTopology
1013:   intAnzahlZeilen = Tabelle19.Cells(2, 3)
1014:   ReDim Lastfallgruppe(intAnzahlZeilen)
1015:   intOffset = 4
1016:   For i = 0 To intAnzahlZeilen - 1
1017:     'Spalte A, Lastfallgruppen-Nummer
1018:     Lastfallgruppe(i).iNo = Tabelle19.Cells(i + intOffset, 1)
1019:     'Spalte B, Bezeichnung
1020:     Lastfallgruppe(i).strDescription = Tabelle19.Cells(i +
intOffset, 2)
1021:     'Spalte C, LG-Faktor
1022:     Lastfallgruppe(i).fLGFactor = Tabelle19.Cells(i + intOffset,
3)
1023:     'Spalte D, Definition
1024:     Lastfallgruppe(i).strDefinition = Tabelle19.Cells(i +
intOffset, 4)
1025:     'Spalte E, Ergebnisse durch LG-Faktor dividieren
1026:     strZelle = Tabelle19.Cells(i + intOffset, 5)
1027:     Select Case strZelle
1028:       Case "Ja"
1029:         Lastfallgruppe(i).bDividing = True
1030:       Case "Nein"
1031:         Lastfallgruppe(i).bDividing = False
1032:       Case Else 'default Nein
1033:         Lastfallgruppe(i).bDividing = False
1034:     End Select
1035:     'Spalte F, Entlastende Wirkung der Zugkräfte berücksichtigen
1036:     strZelle = Tabelle19.Cells(i + intOffset, 6)
1037:     Select Case strZelle
1038:       Case "Ja"
1039:         Lastfallgruppe(i).bTension = True
1040:       Case "Nein"
1041:         Lastfallgruppe(i).bTension = False
1042:       Case Else 'default Nein
1043:         Lastfallgruppe(i).bTension = True
1044:     End Select
1045:     'Spalte G, Abminderungsfaktor für die Steifigkeit
1046:     Lastfallgruppe(i).fGamma_M = Tabelle19.Cells(i + intOffset, 7)
1047:   Next i
1048:   'Übergabe an RFEM
1049:   RFPos.rfGetLoads.rfPrepareModification
1050:   RFPos.rfGetLoads.rfSetLoadGroupArr intAnzahlZeilen,
Lastfallgruppe(0)
1051:   RFPos.rfGetLoads.rfFinishModification
1052:   RFApp.rfUnlockLicence
1053:   'Freigabe der Objekte
1054:   Set RFPos = Nothing
1055:   Set RFTop = Nothing
1056:   Set RFApp = Nothing
1057: End Sub

```

In der Zeile 1007: ist das Feld Lastfallgruppe() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt Lastfälle gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_GROUP.

RF\_LOAD\_GROUP ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bDividing	Ergebnis durch den Lastgruppenfaktor teilen, ja, nein	Boolean

bTension	Entlastende Wirkung von Zugkräften berücksichtigen, ja, nein	Boolean
fGamma_M	Abminderungsfaktor der Steifigkeit	Single
fLGFactor	LG-Faktor	Single
iMethod	Theorie I., II. Oder III. Ordnung	Long
iNo	Nummer der Lastfallgruppe	Long
strDefinition	Definition	String
strDescription	Beschreibung	String

Tabelle 3.45

Die Methode rfSetLoadGroupArr übergibt das Feld Lastfallgruppe() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetLoadGroupArr mitgeteilt, wie viele Zeilen aus dem Feld Lastfallgruppe() übertragen werden sollen.

### 3.2.3 Lastfallkombinationen erzeugen

Das Anlegen von Lastfallkombinationen ähnelt sehr dem Anlegen von Strukturdaten. Im Kapitel 3.1.1 (S. 7) wurde das ausführlich dargestellt. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Lastfallkombinationen erzeugt werden. Die Lastfallkombinationen werden im Excel-Tabellenblatt Lastfallkombinationen definiert und mit dem Makro CommandButton20\_Click() an RFEM übertragen.

```

1058: Sub CommandButton20_Click()
1059:     Dim intAnzahlZeilen As Integer
1060:     Dim Lastfallkombination() As RF_LOAD_COMBINATION
1061:
1062:     Set RFPos = GetObject(, "RFEM.Position")
1063:     Set RFApp = RFPos.rfGetApplication
1064:     RFApp.rfLockLicence
1065:     Set RFTop = RFPos.rfGetTopology
1066:     intAnzahlZeilen = Tabelle20.Cells(2, 3)
1067:     ReDim Lastfallkombination(intAnzahlZeilen)
1068:     intOffset = 4
1069:     For i = 0 To intAnzahlZeilen - 1
1070:         'Spalte A, Lastfallgruppen-Nummer
1071:         Lastfallkombination(i).iNo = Tabelle20.Cells(i + intOffset, 1)
1072:         'Spalte B, Bezeichnung
1073:         Lastfallkombination(i).strDescription = Tabelle20.Cells(i +
intOffset, 2)
1074:         'Spalte C, Definition
1075:         Lastfallkombination(i).strDefinition = Tabelle20.Cells(i +
intOffset, 3)
1076:     Next i
1077:     'Übergabe an RFEM
1078:     RFPos.rfGetLoads.rfPrepareModification
1079:     RFPos.rfGetLoads.rfSetLoadCombinationArr intAnzahlZeilen,
Lastfallkombination(0)
1080:     RFPos.rfGetLoads.rfFinishModification
1081:     RFApp.rfUnlockLicence
1082:     'Freigabe der Objekte
1083:     Set RFPos = Nothing
1084:     Set RFTop = Nothing
1085:     Set RFApp = Nothing
1086: End Sub

```

In der Zeile 1060: ist das Feld Lastfallkombination() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt Lastfallkombination gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_COMBINATION.

RF\_LOAD\_COMBINATION ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
iNo	Nummer der Lastfallkombination	Long
strDefinition	Definition	String
strDescription	Beschreibung	String

Tabelle 3.46

Die Methode rfSetLoadCombinationArr übergibt das Feld Lastfallkombination() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetLoadCombinationArr mitgeteilt, wie viele Zeilen aus dem Feld Lastfallkombination() übertragen werden sollen.

## 3.3 Belastungsdaten erzeugen

### 3.3.1 Punktlasten erzeugen

Das Anlegen von Lasten ähnelt sehr dem Anlegen von Strukturdaten. Im Kapitel 3.1.1 (S. 7) wurde das ausführlich dargestellt. Hier wird nur noch auf die Besonderheiten eingegangen.

In einer bestehenden RFEM-Struktur sollen Punktlasten erzeugt werden. Die Punktlasten werden im Excel-Tabellenblatt 2.1 definiert und mit dem Makro CommandButton22\_Click() an RFEM übertragen.

```

1087: Sub CommandButton22_Click()
1088:   Dim intAnzahlZeilen As Integer
1089:   Dim Punktlast() As RF_LOAD_NODAL_FORCE
1090:
1091:   Set RFPos = GetObject(, "RFEM.Position")
1092:   Set RFApp = RFPos.rfGetApplication
1093:   RFApp.rfLockLicence
1094:   intLastfallNr = Tabelle22.Cells(2, 6)
1095:   Set RFLoad = RFPos.rfGetLoads.rfGetLoadCase(intLastfallNr - 1,
AT_INDEX)
1096:   Set RFTop = RFPos.rfGetTopology
1097:   intAnzahlZeilen = Tabelle22.Cells(2, 3)
1098:   ReDim Punktlast(intAnzahlZeilen)
1099:   intOffset = 4
1100:   For i = 0 To intAnzahlZeilen - 1
1101:     Punktlast(i).iNo = i + 1
1102:     'Spalte A, Punktliste
1103:     Punktlast(i).strNodeList = Tabelle22.Cells(i + intOffset, 1)
1104:     'Spalte B, P-X
1105:     Punktlast(i).force.x = Tabelle22.Cells(i + intOffset, 2)
1106:     'Spalte C, P-Y
1107:     Punktlast(i).force.y = Tabelle22.Cells(i + intOffset, 3)
1108:     'Spalte D, P-Z
1109:     Punktlast(i).force.z = Tabelle22.Cells(i + intOffset, 4)
1110:     'Spalte E, M-X
1111:     Punktlast(i).moment.x = Tabelle22.Cells(i + intOffset, 5)
1112:     'Spalte F, M-Y
1113:     Punktlast(i).moment.y = Tabelle22.Cells(i + intOffset, 6)
1114:     'Spalte G, M-Z
1115:     Punktlast(i).moment.z = Tabelle22.Cells(i + intOffset, 7)
1116:     'Spalte H, Kommentar
1117:     Punktlast(i).strRemark = Tabelle22.Cells(i + intOffset, 8)
1118:   Next i
1119:   'Übergabe an RFEM
1120:   RFLoad.rfPrepareModification
1121:   RFLoad.rfSetNodalForceArr intAnzahlZeilen, Punktlast(0)
1122:   RFLoad.rfFinishModification
1123:   RFApp.rfUnlockLicence
1124:   'Freigabe der Objekte
1125:   Set RFPos = Nothing
1126:   Set RFTop = Nothing

```

```
1127:      Set RFApp = Nothing
1128:      Set RFLoad = Nothing
1129: End Sub
```

In der Zeile 1089: ist das Feld Punktlast() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 2.1 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_NODAL\_FORCE.

RF\_LOAD\_NODAL\_FORCE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
force	Größe der Kraft	Benutzerdefinierter Typ RF_POINT
iNo	Nummer der Punktlast	Long
moment	Größe des Momentes	Benutzerdefinierter Typ RF_POINT
strNodeList	Liste der belasteten Punkte	String
strRemark	Kommentar	String

Tabelle 3.47

RF\_POINT ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
x	Kraft in X-Richtung [N], Moment um X [Nm]	Single
y	Kraft in Y-Richtung [N], Moment um Y [Nm]	Single
z	Kraft in Z-Richtung [N], Moment um Z [Nm]	Single

Tabelle 3.48

Eine Besonderheit ist, dass Lasten einem Lastfall zugeordnet werden müssen. In der Tabellenzelle F2 wird die Nummer des Lastfalls definiert. In der Zeile 1094: wird diese Zelle ausgelesen und in der Variablen intLastfallNr gespeichert. In der Zeile 1095: wird das Objekt RFLoad erzeugt. Mit der Methode rfGetLoadCase wird ihm die entsprechende Lastfallnummer mitgeteilt. Von der Lastfallnummer wird 1 abgezogen, weil die Nummerierung intern bei Null beginnt.

Die Methode rfSetNodalForceArr übergibt das Feld Punktlast() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetNodalForceArr mitgeteilt, wie viele Zeilen aus dem Feld Punktlast() übertragen werden sollen.

### 3.3.2 Stablasten erzeugen

Eine ausführlichere Erläuterung zum Erzeugen von Belastungsdaten ist im Kapitel 3.3.1 (S. 46) zu finden.

In einer bestehenden RFEM-Struktur sollen Stablasten erzeugt werden. Voraussetzungen sind, dass mindestens ein Lastfall und die zu belastenden Stäbe existieren. Die Stablasten werden im Excel-Tabelleblatt 2.2 definiert und mit dem Makro CommandButton23\_Click() an RFEM übertragen.

```
1130: Sub CommandButton23_Click()
1131:     Dim intAnzahlZeilen As Integer
1132:     Dim Stablast() As RF_LOAD_MEMBER
1133:
1134:     Set RFPos = GetObject(, "RFEM.Position")
1135:     Set RFApp = RFPos.rfGetApplication
1136:     RFApp.rfLockLicence
1137:     intLastfallNr = Tabelle23.Cells(2, 6)
```

```

1138:      Set RFLoad = RFPos.rfGetLoads.rfGetLoadCase(intLastfallNr - 1,
AT_INDEX)
1139:      Set RFTop = RFPos.rfGetTopology
1140:      intAnzahlZeilen = Tabelle23.Cells(2, 3)
1141:      ReDim Stablast(intAnzahlZeilen)
1142:      intOffset = 4
1143:      For i = 0 To intAnzahlZeilen - 1
1144:          Stablast(i).iNo = i + 1
1145:          'Spalte A, Stabliste
1146:          Stablast(i).strMemberList = Tabelle23.Cells(i + intOffset, 1)
1147:          'Spalte B, Typ, Spalte K, Höhe
1148:          strZelle = Tabelle23.Cells(i + intOffset, 2)
1149:          Select Case strZelle
1150:              Case "Kraft"
1151:                  Stablast(i).Type = LMT_FORCE
1152:              Case "Moment"
1153:                  Stablast(i).Type = LMT_MOMENT
1154:              Case "T -Änderung"
1155:                  Stablast(i).Type = LMT_THERM1
1156:              Case "T -Differenz"
1157:                  Stablast(i).Type = LMT_THERM2
1158:              Case "L -Änderung"
1159:                  Stablast(i).Type = LMT_DEFORM1
1160:              Case "Krümmung"
1161:                  Stablast(i).Type = LMT_DEFORM2
1162:              Case Else 'default Kraft
1163:                  Stablast(i).Type = LMT_FORCE
1164:          End Select
1165:          'Spalte C, Lastverlauf, Spalten E - J
1166:          strZelle = Tabelle23.Cells(i + intOffset, 3)
1167:          strZelle1 = Tabelle23.Cells(i + intOffset, 5) 'P-1
1168:          strZelle2 = Tabelle23.Cells(i + intOffset, 6) 'P-2
1169:          strZelle3 = Tabelle23.Cells(i + intOffset, 7) 'A
1170:          strZelle4 = Tabelle23.Cells(i + intOffset, 8) 'B
1171:          strZelle5 = Tabelle23.Cells(i + intOffset, 9) 'relativ
1172:          strZelle6 = Tabelle23.Cells(i + intOffset, 10) 'gesamte Länge
1173:          Select Case strZelle
1174:              Case "Einzellast"
1175:                  Stablast(i).Style = LS_SINGLE
1176:                  Stablast(i).styleSingle.fp1 = strZelle1
1177:                  Stablast(i).styleSingle.fA = strZelle3
1178:                  Select Case strZelle5
1179:                      Case "Ja"
1180:                          Stablast(i).styleSingle.bRelative = True
1181:                      Case "Nein"
1182:                          Stablast(i).styleSingle.bRelative = False
1183:                      Case Else 'default Nein
1184:                          Stablast(i).styleSingle.bRelative = False
1185:                  End Select
1186:              Case "Linienlast"
1187:                  Stablast(i).Style = LS_CONSTANT
1188:                  Stablast(i).styleConstant.fp1 = strZelle1
1189:              Case "Trapezlast"
1190:                  Stablast(i).Style = LS_TRAPEZ
1191:                  Stablast(i).styleTrapez.fp1 = strZelle1
1192:                  Stablast(i).styleTrapez.fp2 = strZelle2
1193:                  Stablast(i).styleTrapez.fA = strZelle3
1194:                  Stablast(i).styleTrapez.fB = strZelle4
1195:                  Select Case strZelle5
1196:                      Case "Ja"
1197:                          Stablast(i).styleTrapez.bRelative = True
1198:                      Case "Nein"
1199:                          Stablast(i).styleTrapez.bRelative = False
1200:                      Case Else 'default Nein
1201:                          Stablast(i).styleTrapez.bRelative = False
1202:                  End Select
1203:              Case "Stablast"
1204:                  Select Case strZelle6
1205:                      Case "Ja"
1206:                          Stablast(i).styleTrapez.bEntire = True
1207:                      Case "Nein"
1208:                          Stablast(i).styleTrapez.bEntire = False
1209:                      Case Else 'default Nein
1210:                          Stablast(i).styleTrapez.bEntire = False
1211:                  End Select

```

```

1212:                Case Else 'default Linienlast
1213:                    Stablast(i).Style = LS_CONSTANT
1214:                    Stablast(i).styleConstant.fP1 = strZelle1
1215:                End Select
1216:                'Spalte D, Lastrichtung
1217:                strZelle = Tabelle23.Cells(i + intOffset, 4)
1218:                Select Case strZelle
1219:                    Case "1"
1220:                        Stablast(i).direction = LD_LOCAL_X
1221:                    Case "2"
1222:                        Stablast(i).direction = LD_LOCAL_Y
1223:                    Case "3"
1224:                        Stablast(i).direction = LD_LOCAL_Z
1225:                    Case "GX"
1226:                        Stablast(i).direction = LD_LENGTH_X
1227:                    Case "GY"
1228:                        Stablast(i).direction = LD_LENGTH_Y
1229:                    Case "GZ"
1230:                        Stablast(i).direction = LD_LENGTH_Z
1231:                    Case "X"
1232:                        Stablast(i).direction = LD_PROJECT_X
1233:                    Case "Y"
1234:                        Stablast(i).direction = LD_PROJECT_Y
1235:                    Case "Z"
1236:                        Stablast(i).direction = LD_PROJECT_Z
1237:                    Case Else 'default Z
1238:                        Stablast(i).direction = LD_PROJECT_Z
1239:                End Select
1240:                'Spalte L, Kommentar
1241:                Stablast(i).strRemark = Tabelle23.Cells(i + intOffset, 12)
1242:            Next i
1243:            'Übergabe an RFEM
1244:            RFLoad.rfPrepareModification
1245:            RFLoad.rfSetMemberArr intAnzahlZeilen, Stablast(0)
1246:            RFLoad.rfFinishModification
1247:            RFApp.rfUnlockLicence
1248:            'Freigabe der Objekte
1249:            Set RFPos = Nothing
1250:            Set RFTop = Nothing
1251:            Set RFApp = Nothing
1252:            Set RFLoad = Nothing
1253:        End Sub

```

In der Zeile 1132: ist das Feld Stablast() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 2.2 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_MEMBER.

RF\_LOAD\_MEMBER ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
direction	Lastrichtung	Benutzerdefinierter Aufzählungstyp RF_LOAD_DIRECTION
fH	Höhe der Querschnitt, nur für die Temperaturdifferenz erforderlich	Single
iNo	Nummer der Stablast	Long
strMemberList	Liste der belasteten Stäbe	String
strRemark	Kommentar	String
style	Lastverlauf	Benutzerdefinierter Aufzählungstyp RF_LOAD_STYLE
styleConstant	Lastgröße bei konstantem Verlauf	Benutzerdefinierter Typ RF_LOAD_STYLE_CONSTANT
styleSingle	Lastgrößen bei Einzellast	Benutzerdefinierter Typ RF_LOAD_STYLE_SINGLE

		RF_LOAD_STYLE_SINGLE
styleTrapez	Lastgrößen bei Trapezlast	Benutzerdefinierter Typ RF_LOAD_STYLE_TRAPEZ
type	Lasttyp	Benutzerdefinierter Aufzählungstyp RF_LOAD_MEMBER_TYPE

Tabelle 3.49

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_DIRECTION sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LD_GLOBAL_X	Last in globaler X-Richtung
1	LD_GLOBAL_Y	Last in globaler Y-Richtung
2	LD_GLOBAL_Z	Last in globaler Z-Richtung
3	LD_LOCAL_1	Last in Richtung der Stabachse 1
4	LD_LOCAL_2	Last in Richtung der Stabachse 2
5	LD_LOCAL_3	Last in Richtung der Stabachse 3
6	LD_PROJECT_X	Last auf die projizierte Länge in globaler X-Richtung
7	LD_PROJECT_Y	Last auf die projizierte Länge in globaler Y-Richtung
8	LD_PROJECT_Z	Last auf die projizierte Länge in globaler Z-Richtung

Tabelle 3.50

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_STYLE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LS_SINGLE	Einzellast
1	LS_CONSTANT	Konstanter Lastverlauf
2	LS_TRAPEZ	Trapezlast

Tabelle 3.51

RF\_LOAD\_STYLE\_CONSTANT ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
fP1	Linienlast [N/m]	Single

Tabelle 3.52

RF\_LOAD\_STYLE\_SINGLE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bRelative	Relative Länge, ja , nein	Boolean
fA	Abstand der Last vom Stabanfang Wenn bRelative= TRUE, dann muss ein Zahlenwert zwischen 0 und 1 angegeben werden. =	Single

	entspricht Stabanfang, 1 dem Stabende. Wenn bRelative= FALSE, dann wird der Abstand in [m] angegeben.	
fP1	Größe der Einzellast	Single

Tabelle 3.53

RF\_LOAD\_STYLE\_TRAPEZ ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bEntire	Last über die gesamte Stablänge, ja, nein	Boolean
bRelative	Relative Länge, ja , nein	Boolean
fA	Abstand des Lastbeginns vom Stabanfang Wenn bRelative= TRUE, dann muss ein Zahlenwert zwischen 0 und 1 angegeben werden. = entspricht Stabanfang, 1 dem Stabende. Wenn bRelative= FALSE, dann wird der Abstand in [m] angegeben.	Single
fB	Abstand des Lastendes vom Stabanfang	Single
fP1	Größe der Trapezlast am Anfang	Single
fP2	Größe der Trapezlast am Ende	Single

Tabelle 3.54

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_MEMBER\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LLT_FORCE	Kraft
1	LLT_MOMENT	Moment
2	LLT_THERM1	Temperaturänderung
3	LMT_THERM2	Temperaturdifferenz
4	LMT_DEFORM1	Längenänderung
5	LMT_DEFORM2	Krümmung

Tabelle 3.55

Die Methode rfSetMemberArr übergibt das Feld Stablast() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetMemberArr mitgeteilt, wie viele Zeilen aus dem Feld Stablast() übertragen werden sollen.

### 3.3.3 Linienlasten erzeugen

Eine ausführlichere Erläuterung zum Erzeugen von Belastungsdaten ist im Kapitel 3.3.1 (S. 46) zu finden.

In einer bestehenden RFEM-Struktur sollen Linienlasten erzeugt werden. Voraussetzungen sind, dass mindestens ein Lastfall und die zu belastenden Linien existieren. Die Linienlasten werden im Excel-Tabellenblatt 2.3 definiert und mit dem Makro CommandButton24\_Click() an RFEM übertragen.

```

1254: Sub CommandButton24_Click()
1255:     Dim intAnzahlZeilen As Integer
1256:     Dim Linienlast() As RF_LOAD_LINE_FORCE
1257:
1258:     Set RFPos = GetObject(, "RFEM.Position")
1259:     Set RFApp = RFPos.rfGetApplication
1260:     RFApp.rfLockLicence
1261:     intLastfallNr = Tabelle24.Cells(2, 6)
1262:     Set RFLoad = RFPos.rfGetLoads.rfGetLoadCase(intLastfallNr - 1,
AT_INDEX)
1263:     Set RFTop = RFPos.rfGetTopology
1264:     intAnzahlZeilen = Tabelle24.Cells(2, 3)
1265:     ReDim Linienlast(intAnzahlZeilen)
1266:     intOffset = 4
1267:     For i = 0 To intAnzahlZeilen - 1
1268:         Linienlast(i).iNo = i + 1
1269:         'Spalte A, Linienliste
1270:         Linienlast(i).strLineList = Tabelle24.Cells(i + intOffset, 1)
1271:         'Spalte B, Typ
1272:         strZelle = Tabelle24.Cells(i + intOffset, 2)
1273:         Select Case strZelle
1274:             Case "Kraft"
1275:                 Linienlast(i).Type = LT_FORCE
1276:             Case "Moment"
1277:                 Linienlast(i).Type = LT_MOMENT
1278:             Case Else 'default Kraft
1279:                 Linienlast(i).Type = LT_FORCE
1280:         End Select
1281:         'Spalte C, Lastverlauf, Spalten E - J
1282:         strZelle = Tabelle24.Cells(i + intOffset, 3)
1283:         strZelle1 = Tabelle24.Cells(i + intOffset, 5) 'P-1
1284:         strZelle2 = Tabelle24.Cells(i + intOffset, 6) 'P-2
1285:         strZelle3 = Tabelle24.Cells(i + intOffset, 7) 'A
1286:         strZelle4 = Tabelle24.Cells(i + intOffset, 8) 'B
1287:         strZelle5 = Tabelle24.Cells(i + intOffset, 9) 'relativ
1288:         strZelle6 = Tabelle24.Cells(i + intOffset, 10) 'gesamte Länge
1289:         Select Case strZelle
1290:             Case "Einzellast"
1291:                 Linienlast(i).Style = LS_SINGLE
1292:                 Linienlast(i).styleSingle.fP1 = strZelle1
1293:                 Linienlast(i).styleSingle.fA = strZelle3
1294:                 Select Case strZelle5
1295:                     Case "Ja"
1296:                         Linienlast(i).styleSingle.bRelative = True
1297:                     Case "Nein"
1298:                         Linienlast(i).styleSingle.bRelative = False
1299:                     Case Else 'default Nein
1300:                         Linienlast(i).styleSingle.bRelative = False
1301:                 End Select
1302:             Case "Linienlast"
1303:                 Linienlast(i).Style = LS_CONSTANT
1304:                 Linienlast(i).styleConstant.fP1 = strZelle1
1305:             Case "Trapezlast"
1306:                 Linienlast(i).Style = LS_TRAPEZ
1307:                 Linienlast(i).styleTrapez.fP1 = strZelle1
1308:                 Linienlast(i).styleTrapez.fP2 = strZelle2
1309:                 Linienlast(i).styleTrapez.fA = strZelle3
1310:                 Linienlast(i).styleTrapez.fB = strZelle4
1311:                 Select Case strZelle5
1312:                     Case "Ja"
1313:                         Linienlast(i).styleTrapez.bRelative = True
1314:                     Case "Nein"
1315:                         Linienlast(i).styleTrapez.bRelative = False

```

```

1316:             Case Else 'default Nein
1317:                 Linienlast(i).styleTrapez.bRelative = False
1318:             End Select
1319:         Select Case strZelle6
1320:             Case "Ja"
1321:                 Linienlast(i).styleTrapez.bEntire = True
1322:             Case "Nein"
1323:                 Linienlast(i).styleTrapez.bEntire = False
1324:             Case Else 'default Nein
1325:                 Linienlast(i).styleTrapez.bEntire = False
1326:             End Select
1327:         Case Else 'default Linienlast
1328:             Linienlast(i).Style = LS_CONSTANT
1329:             Linienlast(i).styleConstant.fP1 = strZelle1
1330:         End Select
1331:         'Spalte D, Lastrichtung
1332:         strZelle = Tabelle24.Cells(i + intOffset, 4)
1333:         Select Case strZelle
1334:             Case "1"
1335:                 Linienlast(i).direction = LD_LOCAL_X
1336:             Case "2"
1337:                 Linienlast(i).direction = LD_LOCAL_Y
1338:             Case "3"
1339:                 Linienlast(i).direction = LD_LOCAL_Z
1340:             Case "GX"
1341:                 Linienlast(i).direction = LD_LENGTH_X
1342:             Case "GY"
1343:                 Linienlast(i).direction = LD_LENGTH_Y
1344:             Case "GZ"
1345:                 Linienlast(i).direction = LD_LENGTH_Z
1346:             Case "X"
1347:                 Linienlast(i).direction = LD_PROJECT_X
1348:             Case "Y"
1349:                 Linienlast(i).direction = LD_PROJECT_Y
1350:             Case "Z"
1351:                 Linienlast(i).direction = LD_PROJECT_Z
1352:             Case Else 'default Z
1353:                 Linienlast(i).direction = LD_PROJECT_Z
1354:         End Select
1355:         'Spalte L, Kommentar
1356:         Linienlast(i).strRemark = Tabelle24.Cells(i + intOffset, 11)
1357:     Next i
1358:     'Übergabe an RFEM
1359:     RFLoad.rfPrepareModification
1360:     RFLoad.rfSetLineForceArr intAnzahlZeilen, Linienlast(0)
1361:     RFLoad.rfFinishModification
1362:     RFApp.rfUnlockLicence
1363:     'Freigabe der Objekte
1364:     Set RFPos = Nothing
1365:     Set RFTop = Nothing
1366:     Set RFApp = Nothing
1367:     Set RFLoad = Nothing
1368: End Sub

```

In der Zeile 1256: ist das Feld Linienlast() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 2.3 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_LINE\_FORCE.

RF\_LOAD\_LINE\_FORCE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
direction	Lastrichtung	Benutzerdefinierter Aufzählungstyp RF_LOAD_DIRECTION
iNo	Nummer der Linienlast	Long
strLineList	Liste der belasteten Linien	String
strRemark	Kommentar	String

style	Lastverlauf	Benutzerdefinierter Aufzählungstyp RF_LOAD_STYLE
styleConstant	Lastgröße bei konstantem Verlauf	Benutzerdefinierter Typ RF_LOAD_STYLE_CONSTANT
styleSingle	Lastgrößen bei Einzellast	Benutzerdefinierter Typ RF_LOAD_STYLE_SINGLE
styleTrapez	Lastgrößen bei Trapezlast	Benutzerdefinierter Typ RF_LOAD_STYLE_TRAPEZ
type	Lasttyp	Benutzerdefinierter Aufzählungstyp RF_LOAD_LINE_TYPE

Tabelle 3.56

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_DIRECTION sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LD_GLOBAL_X	Last in globaler X-Richtung
1	LD_GLOBAL_Y	Last in globaler Y-Richtung
2	LD_GLOBAL_Z	Last in globaler Z-Richtung
3	LD_LOCAL_1	Last in Richtung der Stabachse 1
4	LD_LOCAL_2	Last in Richtung der Stabachse 2
5	LD_LOCAL_3	Last in Richtung der Stabachse 3
6	LD_PROJECT_X	Last auf die projizierte Länge in globaler X-Richtung
7	LD_PROJECT_Y	Last auf die projizierte Länge in globaler Y-Richtung
8	LD_PROJECT_Z	Last auf die projizierte Länge in globaler Z-Richtung

Tabelle 3.57

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_STYLE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LS_SINGLE	Einzellast
1	LS_CONSTANT	Konstanter Lastverlauf
2	LS_TRAPEZ	Trapezlast

Tabelle 3.58

RF\_LOAD\_STYLE\_CONSTANT ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
fP1	Linienlast [N/m]	Single

Tabelle 3.59

RF\_LOAD\_STYLE\_SINGLE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bRelative	Relative Länge, ja , nein	Boolean
fA	Abstand der Last vom Stabanfang Wenn bRelative= TRUE, dann muss ein Zahlenwert zwischen 0 und 1 angegeben werden. = entspricht Stabanfang, 1 dem Stabende.  Wenn bRelative= FALSE, dann wird der Abstand in [m] angegeben.	Single
fP1	Größe der Einzellast	Single

Tabelle 3.60

RF\_LOAD\_STYLE\_TRAPEZ ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bEntire	Last über die gesamte Stablänge, ja, nein	Boolean
bRelative	Relative Länge, ja , nein	Boolean
fA	Abstand des Lastbeginns vom Stabanfang Wenn bRelative= TRUE, dann muss ein Zahlenwert zwischen 0 und 1 angegeben werden. = entspricht Stabanfang, 1 dem Stabende.  Wenn bRelative= FALSE, dann wird der Abstand in [m] angegeben.	Single
fB	Abstand des Lastendes vom Stabanfang	Single
fP1	Größe der Trapezlast am Anfang	Single
fP2	Größe der Trapezlast am Ende	Single

Tabelle 3.61

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_LINE\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LLT_FORCE	Kraft
1	LLT_MOMENT	Moment

Tabelle 3.62

Die Methode rfSetLineArr übergibt das Feld Linienlast() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetLineArr mitgeteilt, wie viele Zeilen aus dem Feld Linienlast() übertragen werden sollen.

### 3.3.4 Flächenlasten erzeugen

Eine ausführlichere Erläuterung zum Erzeugen von Belastungsdaten ist im Kapitel 3.3.1 (S. 46) zu finden.

In einer bestehenden RFEM-Struktur sollen Flächenlasten erzeugt werden. Voraussetzungen sind, dass mindestens ein Lastfall und die zu belastenden Flächen existieren. Die Flächenlasten werden im Excel-Tabellenblatt 2.4 definiert und mit dem Makro CommandButton25\_Click() an RFEM übertragen.

```

1369: Sub CommandButton25_Click()
1370:   Dim intAnzahlZeilen As Integer
1371:   Dim Flaechenlast() As RF_LOAD_SURFACE
1372:
1373:   Set RFPos = GetObject(, "RFEM.Position")
1374:   Set RFApp = RFPos.rfGetApplication
1375:   RFApp.rfLockLicence
1376:   intLastfallNr = Tabelle25.Cells(2, 6)
1377:   Set RFLoad = RFPos.rfGetLoads.rfGetLoadCase(intLastfallNr - 1,
AT_INDEX)
1378:   Set RFTop = RFPos.rfGetTopology
1379:   intAnzahlZeilen = Tabelle25.Cells(2, 3)
1380:   ReDim Flaechenlast(intAnzahlZeilen)
1381:   intOffset = 4 'In der Zeile 4 soll mit dem Auslesen begonnen
werden
1382:   For i = 0 To intAnzahlZeilen - 1
1383:     Flaechenlast(i).iNo = i + 1
1384:     'Spalte A, Flächenliste
1385:     Flaechenlast(i).strSurfaceList = Tabelle25.Cells(i +
intOffset, 1)
1386:     'Spalte B, Typ
1387:     strZelle = Tabelle25.Cells(i + intOffset, 2)
1388:     Select Case strZelle
1389:       Case "Kraft"
1390:         Flaechenlast(i).Type = LST_FORCE
1391:       Case "T-Aenderung"
1392:         Flaechenlast(i).Type = LST_THERM1
1393:       Case "T-Differenz"
1394:         Flaechenlast(i).Type = LST_THERM2
1395:       Case "L-Aenderung"
1396:         Flaechenlast(i).Type = LST_DEFORM1
1397:       Case "Kruemmung"
1398:         Flaechenlast(i).Type = LST_DEFORM2
1399:       Case Else 'default Kraft
1400:         Flaechenlast(i).Type = LST_FORCE
1401:     End Select
1402:     'Spalte C, Lastverlauf, Spalten E - J
1403:     strZelle = Tabelle25.Cells(i + intOffset, 3)
1404:     Select Case strZelle
1405:       Case "Konstant"
1406:         Flaechenlast(i).Style = LSS_UNIFORM
1407:         Flaechenlast(i).fNode1Value = Tabelle25.Cells(i +
intOffset, 6)
1408:       Case "Schief"
1409:         Flaechenlast(i).Style = LSS_TRIANGLE
1410:         Flaechenlast(i).iNode1No = Tabelle25.Cells(i +
intOffset, 5)
1411:         Flaechenlast(i).fNode1Value = Tabelle25.Cells(i +
intOffset, 6)
1412:         Flaechenlast(i).iNode2No = Tabelle25.Cells(i +
intOffset, 7)
1413:         Flaechenlast(i).fNode2Value = Tabelle25.Cells(i +
intOffset, 8)
1414:         Flaechenlast(i).iNode3No = Tabelle25.Cells(i +
intOffset, 9)
1415:         Flaechenlast(i).fNode3Value = Tabelle25.Cells(i +
intOffset, 10)
1416:       Case "SchiefX"
1417:         Flaechenlast(i).Style = LSS_GRADIENT_X
1418:         Flaechenlast(i).iNode1No = Tabelle25.Cells(i +
intOffset, 5)
1419:         Flaechenlast(i).fNode1Value = Tabelle25.Cells(i +
intOffset, 6)

```

```

1420:           Flaechenlast(i).iNode2No = Tabelle25.Cells(i +
           intOffset, 7)
1421:           Flaechenlast(i).fNode2Value = Tabelle25.Cells(i +
           intOffset, 8)
1422:           Case "SchiefY"
1423:             Flaechenlast(i).Style = LSS_GRADIENT_Y
1424:             Flaechenlast(i).iNode1No = Tabelle25.Cells(i +
           intOffset, 5)
1425:             Flaechenlast(i).fNode1Value = Tabelle25.Cells(i +
           intOffset, 6)
1426:             Flaechenlast(i).iNode2No = Tabelle25.Cells(i +
           intOffset, 7)
1427:             Flaechenlast(i).fNode2Value = Tabelle25.Cells(i +
           intOffset, 8)
1428:           Case "SchiefZ"
1429:             Flaechenlast(i).Style = LSS_GRADIENT_Z
1430:             Flaechenlast(i).iNode1No = Tabelle25.Cells(i +
           intOffset, 5)
1431:             Flaechenlast(i).fNode1Value = Tabelle25.Cells(i +
           intOffset, 6)
1432:             Flaechenlast(i).iNode2No = Tabelle25.Cells(i +
           intOffset, 7)
1433:             Flaechenlast(i).fNode2Value = Tabelle25.Cells(i +
           intOffset, 8)
1434:           Case Else 'default Konstant
1435:             Flaechenlast(i).Style = LSS_UNIFORM
1436:             Flaechenlast(i).fNode1Value = Tabelle25.Cells(i +
           intOffset, 5)
1437:         End Select
1438:         'Spalte D, Lastrichtung
1439:         strZelle = Tabelle25.Cells(i + intOffset, 4)
1440:         Select Case strZelle
1441:           Case "Lx"
1442:             Flaechenlast(i).direction = LD_LOCAL_1
1443:           Case "Ly"
1444:             Flaechenlast(i).direction = LD_LOCAL_2
1445:           Case "Lz"
1446:             Flaechenlast(i).direction = LD_LOCAL_3
1447:           Case "GX"
1448:             Flaechenlast(i).direction = LD_GLOBAL_X
1449:           Case "GY"
1450:             Flaechenlast(i).direction = LD_GLOBAL_Y
1451:           Case "GZ"
1452:             Flaechenlast(i).direction = LD_GLOBAL_Z
1453:           Case "X"
1454:             Flaechenlast(i).direction = LD_PROJECT_X
1455:           Case "Y"
1456:             Flaechenlast(i).direction = LD_PROJECT_Y
1457:           Case "Z"
1458:             Flaechenlast(i).direction = LD_PROJECT_Z
1459:           Case Else 'default Z
1460:             Flaechenlast(i).direction = LD_PROJECT_Z
1461:         End Select
1462:         'Spalte L, Kommentar
1463:         Flaechenlast(i).strRemark = Tabelle25.Cells(i + intOffset, 11)
1464:     Next i
1465:     'Übergabe an RFEM
1466:     RFLoad.rfPrepareModification
1467:     RFLoad.rfSetSurfaceArr intAnzahlZeilen, Flaechenlast(0)
1468:     RFLoad.rfFinishModification
1469:     RFApp.rfUnlockLicence
1470:     'Freigabe der Objekte
1471:     Set RFPos = Nothing
1472:     Set RFTop = Nothing
1473:     Set RFApp = Nothing
1474:     Set RFLoad = Nothing
1475: End Sub

```

In der Zeile 1371: ist das Feld Flaechenlast() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 2.4 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_SURFACE.

RF\_LOAD\_SURFACE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
direction	Lastrichtung	Benutzerdefinierter Aufzählungstyp RF_LOAD_DIRECTION
fNode1Value	Belastung im Punkt 1	Single
fNode2Value	Belastung im Punkt 2	Single
fNode3Value	Belastung im Punkt 3	Single
iNo	Nummer der Flächenlast	Long
iNode1No	Punktnummer für den 1. Punkt	Long
iNode2No	Punktnummer für den 2. Punkt	Long
iNode3No	Punktnummer für den 3. Punkt	Long
strRemark	Kommentar	String
strSurfacelist	Liste der belasteten Flächen	
style	Lastverlauf	Benutzerdefinierter Aufzählungstyp RF_LOAD_SURFACE_STYLE
type	Lasttyp	Benutzerdefinierter Aufzählungstyp RF_LOAD_SURFACE_TYPE

Tabelle 3.63

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_DIRECTION sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LD_GLOBAL_X	Last in globaler X-Richtung
1	LD_GLOBAL_Y	Last in globaler Y-Richtung
2	LD_GLOBAL_Z	Last in globaler Z-Richtung
3	LD_LOCAL_1	Last in lokaler x-Richtung
4	LD_LOCAL_2	Last in lokaler y-Richtung
5	LD_LOCAL_3	Last in lokaler z-Richtung
6	LD_PROJECT_X	Last auf die projizierte Länge in globaler X-Richtung
7	LD_PROJECT_Y	Last auf die projizierte Länge in globaler Y-Richtung
8	LD_PROJECT_Z	Last auf die projizierte Länge in globaler Z-Richtung

Tabelle 3.64

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_SURFACE\_STYLE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LSS_UNIFORM	Gleichmäßig
1	LSS_TRIANGLE	Definition über drei Punkte

2	LSS_GRADIENT_X	Veränderlich in X
3	LSS_GRADIENT_Y	Veränderlich in Y
4	LSS_GRADIENT_Z	Veränderlich in Z

Tabelle 3.65

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_SURFACE\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LST_FORCE	Kraft
2	LST_THERM1	Temperaturänderung
3	LST_THERM2	Temperaturdifferenz
4	LST_DEFORM1	Längenänderung
5	LST_DEFORM2	Krümmung

Tabelle 3.66

Die Methode rfSetSurfaceArr übergibt das Feld Flaechenlast() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetSurfaceArr mitgeteilt, wie viele Zeilen aus dem Feld Flaechenlast() übertragen werden sollen.

### 3.3.5 Freie Punktlasten erzeugen

Eine ausführlichere Erläuterung zum Erzeugen von Belastungsdaten ist im Kapitel 3.3.1 (S. 46) zu finden.

In einer bestehenden RFEM-Struktur sollen freie Punktlasten erzeugt werden. Voraussetzungen sind, dass mindestens ein Lastfall und die zu belastenden Flächen existieren. Die freie Punktlasten werden im Excel-Tabellenblatt 2.5 definiert und mit dem Makro CommandButton26\_Click() an RFEM übertragen.

```

1476: Sub CommandButton26_Click()
1477:     Dim intAnzahlZeilen As Integer
1478:     Dim FreiePunktlast() As RF_LOAD_FREE_POINT
1479:
1480:     Set RFPos = GetObject(, "RFEM.Position")
1481:     Set RFApp = RFPos.rfGetApplication
1482:     RFApp.rfLockLicence
1483:     intLastfallNr = Tabelle26.Cells(2, 6)
1484:     Set RFLoad = RFPos.rfGetLoads.rfGetLoadCase(intLastfallNr - 1,
AT_INDEX)
1485:     Set RFTop = RFPos.rfGetTopology
1486:     intAnzahlZeilen = Tabelle26.Cells(2, 3)
1487:     ReDim FreiePunktlast(intAnzahlZeilen)
1488:     intOffset = 4 'In der Zeile 4 soll mit dem Auslesen begonnen
werden
1489:     For i = 0 To intAnzahlZeilen - 1
1490:         FreiePunktlast(i).iNo = i + 1
1491:         'Spalte A, Flächenliste
1492:         FreiePunktlast(i).strSurfaceList = Tabelle26.Cells(i +
intOffset, 1)
1493:         'Spalte B, Projektion
1494:         strZelle = Tabelle26.Cells(i + intOffset, 2)
1495:         Select Case strZelle
1496:             Case "XY"
1497:                 FreiePunktlast(i).plane = LFP_XY
1498:             Case "YZ"
1499:                 FreiePunktlast(i).plane = LFP_YZ
1500:             Case "XZ"
1501:                 FreiePunktlast(i).plane = LFP_XZ
1502:             Case Else 'default XY
1503:                 FreiePunktlast(i).plane = LFP_XY

```

```

1504:      End Select
1505:      'Spalte C, X-Koordinate
1506:      FreiePunktlast(i).pl.x = Tabelle26.Cells(i + intOffset, 3)
1507:      'Spalte D, Y-Koordinate
1508:      FreiePunktlast(i).pl.y = Tabelle26.Cells(i + intOffset, 4)
1509:      'Spalte E, Z-Koordinate
1510:      FreiePunktlast(i).pl.z = Tabelle26.Cells(i + intOffset, 5)
1511:      'Spalte E, Lasttyp
1512:      strZelle = Tabelle26.Cells(i + intOffset, 6)
1513:      Select Case strZelle
1514:      Case "Kraft"
1515:          FreiePunktlast(i).Type = FPT_FORCE
1516:      Case "Moment"
1517:          FreiePunktlast(i).Type = FPT_MOMENT
1518:      Case Else 'default Kraft
1519:          FreiePunktlast(i).Type = FPT_FORCE
1520:      End Select
1521:      'Spalte F, Lastrichtung
1522:      strZelle = Tabelle26.Cells(i + intOffset, 7)
1523:      Select Case strZelle
1524:      Case "Lx"
1525:          FreiePunktlast(i).direction = FPD_LOCAL_X
1526:      Case "Ly"
1527:          FreiePunktlast(i).direction = FPD_LOCAL_Y
1528:      Case "Lz"
1529:          FreiePunktlast(i).direction = FPD_LOCAL_Z
1530:      Case "X"
1531:          FreiePunktlast(i).direction = FPD_GLOBAL_X
1532:      Case "Y"
1533:          FreiePunktlast(i).direction = FPD_GLOBAL_Y
1534:      Case "Z"
1535:          FreiePunktlast(i).direction = FPD_GLOBAL_Z
1536:      Case Else 'default Z
1537:          FreiePunktlast(i).direction = FPD_GLOBAL_Z
1538:      End Select
1539:      'Spalte G, Last
1540:      FreiePunktlast(i).fLoadValue = Tabelle26.Cells(i + intOffset,
8)
1541:      'Spalte H, Kommentar
1542:      FreiePunktlast(i).strRemark = Tabelle26.Cells(i + intOffset,
9)
1543:      Next i
1544:      'Übergabe an RFEM
1545:      RFLoad.rfPrepareModification
1546:      RFLoad.rfSetFreePointArr intAnzahlZeilen, FreiePunktlast(0)
1547:      RFLoad.rfFinishModification
1548:      RFApp.rfUnlockLicence
1549:      'Freigabe der Objekte
1550:      Set RFPos = Nothing
1551:      Set RFTop = Nothing
1552:      Set RFApp = Nothing
1553:      Set RFLoad = Nothing
1554:  End Sub

```

In der Zeile 1478: ist das Feld FreiePunktlast() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 2.5 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_FREE\_POINT.

RF\_LOAD\_FREE\_POINT ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
direction	Lastrichtung	Benutzerdefinierter Aufzählungstyp RF_LOAD_FREE_POINT_DIRECTION
fLoadValue	Belastung	Single
iNo	Nummer der freien Punktlast	Long
p1	Koordinaten der Last	Benutzerdefinierter Typ RF_POINT
plane	Projektionsebene	Benutzerdefinierter Aufzählungstyp RF_LOAD_FREE_PLANE

		typ RF_LOAD_FREE_PLANE
strRemark	Kommentar	String
strSurfaceList	Liste der belasteten Flächen	String
type	Lasttyp	Benutzerdefinierter Aufzählungstyp RF_LOAD_FREE_POINT_TYPE

Tabelle 3.67

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_FREE\_POINT\_DIRECTION sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	FPD_LOCAL_X	Lokal in x
1	FPD_LOCAL_Y	Lokal in y
2	FPD_LOCAL_Z	Lokal in z
3	FPD_GLOBAL_X	Global in X
4	FPD_GLOBAL_Y	Global in Y
5	FPD_GLOBAL_Z	Global in Z

Tabelle 3.68

RF\_POINT ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
x	X-Koordinate (auf das globale Koordinatensystem bezogen)	Single
y	Y-Koordinate	Single
z	Z-Koordinate	Single

Tabelle 3.69

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_FREE\_PLANE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LFP_XY	Lastprojektionsebene XY
1	LFP_YZ	Lastprojektionsebene YZ
2	LFP_XZ	Lastprojektionsebene XZ

Tabelle 3.70

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_FREE\_POINT\_TYPEE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	FPT_FORCE	Kraft
1	FPT_MOMENT	Moment

Tabelle 3.71

Die Methode `rfSetFreePointArr` übergibt das Feld `FreiePunktlast()` an RFEM. Mit der Variable `intAnzahlZeilen` wird `rfSetFreePointArr` mitgeteilt, wie viele Zeilen aus dem Feld `FreiePunktlast()` übertragen werden sollen.

### 3.3.6 Freie Linienlasten erzeugen

Eine ausführlichere Erläuterung zum Erzeugen von Belastungsdaten ist im Kapitel 3.3.1 (S. 46) zu finden.

In einer bestehenden RFEM-Struktur sollen freie Linienlasten erzeugt werden.

Voraussetzungen sind, dass mindestens ein Lastfall und die zu belastenden Flächen existieren. Die freie Linienlasten werden im Excel-Tabellenblatt 2.6 definiert und mit dem Makro `CommandButton27_Click()` an RFEM übertragen.

```

1555: Sub CommandButton27_Click()
1556:     Dim intAnzahlZeilen As Integer
1557:     Dim FreieLinienlast() As RF_LOAD_FREE_LINE
1558:
1559:     Set RFPos = GetObject(, "RFEM.Position")
1560:     Set RFApp = RFPos.rfGetApplication
1561:     RFApp.rfLockLicence
1562:     intLastfallNr = Tabelle27.Cells(2, 6)
1563:     Set RFLoad = RFPos.rfGetLoads.rfGetLoadCase(intLastfallNr - 1,
AT_INDEX)
1564:     Set RFTop = RFPos.rfGetTopology
1565:     intAnzahlZeilen = Tabelle27.Cells(2, 3)
1566:     ReDim FreieLinienlast(intAnzahlZeilen)
1567:     intOffset = 4
1568:     For i = 0 To intAnzahlZeilen - 1
1569:         FreieLinienlast(i).iNo = i + 1
1570:         'Spalte A, Flächenliste
1571:         FreieLinienlast(i).strSurfaceList = Tabelle27.Cells(i +
intOffset, 1)
1572:         'Spalte B, Projektion
1573:         strZelle = Tabelle27.Cells(i + intOffset, 2)
1574:         Select Case strZelle
1575:             Case "XY"
1576:                 FreieLinienlast(i).plane = LFP_XY
1577:             Case "YZ"
1578:                 FreieLinienlast(i).plane = LFP_YZ
1579:             Case "XZ"
1580:                 FreieLinienlast(i).plane = LFP_XZ
1581:             Case Else 'default XY
1582:                 FreieLinienlast(i).plane = LFP_XY
1583:         End Select
1584:         'Spalte C, X-Koordinate Punkt 1
1585:         FreieLinienlast(i).p1.x = Tabelle27.Cells(i + intOffset, 3)
1586:         'Spalte D, Y-Koordinate Punkt 2
1587:         FreieLinienlast(i).p1.y = Tabelle27.Cells(i + intOffset, 4)
1588:         'Spalte E, Z-Koordinate Punkt 3
1589:         FreieLinienlast(i).p1.z = Tabelle27.Cells(i + intOffset, 5)
1590:         'Spalte F, X-Koordinate Punkt 1
1591:         FreieLinienlast(i).p2.x = Tabelle27.Cells(i + intOffset, 6)
1592:         'Spalte G, Y-Koordinate Punkt 2
1593:         FreieLinienlast(i).p2.y = Tabelle27.Cells(i + intOffset, 7)
1594:         'Spalte H, Z-Koordinate Punkt 3
1595:         FreieLinienlast(i).p2.z = Tabelle27.Cells(i + intOffset, 8)
1596:         'Spalte I, Lasttyp
1597:         strZelle = Tabelle27.Cells(i + intOffset, 9)
1598:         Select Case strZelle
1599:             Case "Konstant"
1600:                 FreieLinienlast(i).Type = FLT_CONSTANT
1601:             Case "Linear"
1602:                 FreieLinienlast(i).Type = FLT_LINEAR
1603:             Case Else 'default Konstant
1604:                 FreieLinienlast(i).Type = FLT_CONSTANT
1605:         End Select
1606:         'Spalte J, Lastrichtung
1607:         strZelle = Tabelle27.Cells(i + intOffset, 10)
1608:         Select Case strZelle
1609:             Case "Lx"
1610:                 FreieLinienlast(i).direction = LFD_LOCAL_X

```

```

1611:         Case "Ly"
1612:             FreieLinienlast(i).direction = LFD_LOCAL_Y
1613:         Case "Lz"
1614:             FreieLinienlast(i).direction = LFD_LOCAL_Z
1615:         Case "GX"
1616:             FreieLinienlast(i).direction = LFD_GLOBAL_X
1617:         Case "GY"
1618:             FreieLinienlast(i).direction = LFD_GLOBAL_Y
1619:         Case "GZ"
1620:             FreieLinienlast(i).direction = LFD_GLOBAL_Z
1621:         Case "PX"
1622:             FreieLinienlast(i).direction = LFD_PROJECT_X
1623:         Case "PY"
1624:             FreieLinienlast(i).direction = LFD_PROJECT_Y
1625:         Case "PZ"
1626:             FreieLinienlast(i).direction = LFD_PROJECT_Z
1627:         Case Else 'default GZ
1628:             FreieLinienlast(i).direction = LFD_GLOBAL_Z
1629:         End Select
1630:         'Spalte K, Last p-1
1631:         FreieLinienlast(i).fLoad1Value = Tabelle27.Cells(i +
intOffset, 11)
1632:         'Spalte L, Last p-2
1633:         FreieLinienlast(i).fLoad2Value = Tabelle27.Cells(i +
intOffset, 12)
1634:         'Spalte M, Kommentar
1635:         FreieLinienlast(i).strRemark = Tabelle27.Cells(i + intOffset,
13)
1636:     Next i
1637:     'Übergabe an RFEM
1638:     RFLoad.rfPrepareModification
1639:     RFLoad.rfSetFreeLineArr intAnzahlZeilen, FreieLinienlast(0)
1640:     RFLoad.rfFinishModification
1641:     RFApp.rfUnlockLicence
1642:     'Freigabe der Objekte
1643:     Set RFPos = Nothing
1644:     Set RFTop = Nothing
1645:     Set RFApp = Nothing
1646:     Set RFLoad = Nothing
1647: End Sub

```

In der Zeile 1557: ist das Feld FreieLinienlast() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 2.6 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_FREE\_LINE.

RF\_LOAD\_FREE\_LINE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
direction	Lastrichtung	Benutzerdefinierter Aufzählungstyp RF_LOAD_FREE_DIRECTION
fLoad1Value	Belastung 1	Single
fLoad2Value	Belastung 2	Single
iNo	Nummer der freien Linienlast	Long
p1	Koordinaten der Last am Anfangspunkt	Benutzerdefinierter Typ RF_POINT
p2	Koordinaten der Last am Endpunkt	Benutzerdefinierter Typ RF_POINT
plane	Projektionsebene	Benutzerdefinierter Aufzählungstyp RF_LOAD_FREE_PLANE
strRemark	Kommentar	String
strSurfaceList	Liste der belasteten Flächen	String
type	Lasttyp	Benutzerdefinierter Aufzählungstyp RF_LOAD_FREE_LINE_TYPE

		typ RF_LOAD_FREE_LINE_TYPE
--	--	----------------------------

Tabelle 3.72

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_FREE\_DIRECTION sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LFD_LOCAL_X	Lokal in x
1	LFD_LOCAL_Y	Lokal in y
2	LFD_LOCAL_Z	Lokal in z
3	LFD_GLOBAL_X	Global in X
4	LFD_GLOBAL_Y	Global in Y
5	LFD_GLOBAL_Z	Global in Z
6	LFD_PROJECT_X	Global in X bezogen auf die projizierte Länge
7	LFD_PROJECT_Y	Global in Y bezogen auf die projizierte Länge
8	LFD_PROJECT_Z	Global in Z bezogen auf die projizierte Länge

Tabelle 3.73

RF\_POINT ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
x	X-Koordinate (auf das globale Koordinatensystem bezogen)	Single
y	Y-Koordinate	Single
z	Z-Koordinate	Single

Tabelle 3.74

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_FREE\_PLAN sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LFP_XY	Lastprojektionsebene XY
1	LFP_YZ	Lastprojektionsebene YZ
2	LFP_XZ	Lastprojektionsebene XZ

Tabelle 3.75

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_FREE\_LINE\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	FLT_CONSTANT	Konstant
1	FLT_LINEAR	Linear veränderlich

Tabelle 3.76

Die Methode rfSetFreeLineArr übergibt das Feld FreieLinienlast() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetFreeLineArr mitgeteilt, wie viele Zeilen aus dem Feld FreieLinienlast() übertragen werden sollen.

### 3.3.7 Freie Rechtecklasten erzeugen

Eine ausführlichere Erläuterung zum Erzeugen von Belastungsdaten ist im Kapitel 3.3.1 (S. 46) zu finden.

In einer bestehenden RFEM-Struktur sollen freie Rechtecklasten erzeugt werden. Voraussetzungen sind, dass mindestens ein Lastfall und die zu belastenden Flächen existieren. Die freie Rechtecklasten werden im Excel-Tabellenblatt 2.7 definiert und mit dem Makro CommandButton28\_Click() an RFEM übertragen.

```

1648: Sub CommandButton28_Click()
1649:   Dim intAnzahlZeilen As Integer
1650:   Dim FreieRechtecklast() As RF_LOAD_FREE_RECTANGLE
1651:
1652:   Set RFPos = GetObject(, "RFEM.Position")
1653:   Set RFApp = RFPos.rfGetApplication
1654:   RFApp.rfLockLicence
1655:   intLastfallNr = Tabelle28.Cells(2, 6)
1656:   Set RFLoad = RFPos.rfGetLoads.rfGetLoadCase(intLastfallNr - 1,
AT_INDEX)
1657:   Set RFTop = RFPos.rfGetTopology
1658:   intAnzahlZeilen = Tabelle28.Cells(2, 3)
1659:   ReDim FreieRechtecklast(intAnzahlZeilen)
1660:   intOffset = 4 'In der Zeile 4 soll mit dem Auslesen begonnen
werden
1661:   For i = 0 To intAnzahlZeilen - 1
1662:     FreieRechtecklast(i).iNo = i + 1
1663:     'Spalte A, Flächenliste
1664:     FreieRechtecklast(i).strSurfaceList = Tabelle28.Cells(i +
intOffset, 1)
1665:     'Spalte B, Projektion
1666:     strZelle = Tabelle28.Cells(i + intOffset, 2)
1667:     Select Case strZelle
1668:       Case "XY"
1669:         FreieRechtecklast(i).plane = LFP_XY
1670:       Case "YZ"
1671:         FreieRechtecklast(i).plane = LFP_YZ
1672:       Case "XZ"
1673:         FreieRechtecklast(i).plane = LFP_XZ
1674:       Case Else 'default XY
1675:         FreieRechtecklast(i).plane = LFP_XY
1676:     End Select
1677:     'Spalte C, X-Koordinate Punkt 1
1678:     FreieRechtecklast(i).p1.x = Tabelle28.Cells(i + intOffset, 3)
1679:     'Spalte D, Y-Koordinate Punkt 2
1680:     FreieRechtecklast(i).p1.y = Tabelle28.Cells(i + intOffset, 4)
1681:     'Spalte E, Z-Koordinate Punkt 3
1682:     FreieRechtecklast(i).p1.z = Tabelle28.Cells(i + intOffset, 5)
1683:     'Spalte F, X-Koordinate Punkt 1
1684:     FreieRechtecklast(i).p2.x = Tabelle28.Cells(i + intOffset, 6)
1685:     'Spalte G, Y-Koordinate Punkt 2
1686:     FreieRechtecklast(i).p2.y = Tabelle28.Cells(i + intOffset, 7)
1687:     'Spalte H, Z-Koordinate Punkt 3
1688:     FreieRechtecklast(i).p2.z = Tabelle28.Cells(i + intOffset, 8)
1689:     'Spalte I, Lasttyp
1690:     strZelle = Tabelle28.Cells(i + intOffset, 9)
1691:     Select Case strZelle
1692:       Case "Konstant"
1693:         FreieRechtecklast(i).Type = LRT_CONSTANT
1694:       Case "GradientX"
1695:         FreieRechtecklast(i).Type = LRT_GRADIENT_X
1696:       Case "GradientY"
1697:         FreieRechtecklast(i).Type = LRT_GRADIENT_Y
1698:       Case "GradientZ"
1699:         FreieRechtecklast(i).Type = LRT_GRADIENT_Z
1700:       Case Else 'default Konstant
1701:         FreieRechtecklast(i).Type = LRT_CONSTANT
1702:     End Select
1703:     'Spalte J, Lastrichtung
1704:     strZelle = Tabelle28.Cells(i + intOffset, 10)
1705:     Select Case strZelle
1706:       Case "Lx"
1707:         FreieRechtecklast(i).direction = LFD_LOCAL_X

```

```

1708:         Case "Ly"
1709:             FreieRechtecklast(i).direction = LFD_LOCAL_Y
1710:         Case "Lz"
1711:             FreieRechtecklast(i).direction = LFD_LOCAL_Z
1712:         Case "GX"
1713:             FreieRechtecklast(i).direction = LFD_GLOBAL_X
1714:         Case "GY"
1715:             FreieRechtecklast(i).direction = LFD_GLOBAL_Y
1716:         Case "GZ"
1717:             FreieRechtecklast(i).direction = LFD_GLOBAL_Z
1718:         Case "PX"
1719:             FreieRechtecklast(i).direction = LFD_PROJECT_X
1720:         Case "PY"
1721:             FreieRechtecklast(i).direction = LFD_PROJECT_Y
1722:         Case "PZ"
1723:             FreieRechtecklast(i).direction = LFD_PROJECT_Z
1724:         Case Else 'default GZ
1725:             FreieRechtecklast(i).direction = LFD_GLOBAL_Z
1726:     End Select
1727:     'Spalte K, Last p-1
1728:     FreieRechtecklast(i).fLoad1Value = Tabelle28.Cells(i +
intOffset, 11)
1729:     'Spalte L, Last p-2
1730:     FreieRechtecklast(i).fLoad2Value = Tabelle28.Cells(i +
intOffset, 12)
1731:     'Spalte M, Kommentar
1732:     FreieRechtecklast(i).strRemark = Tabelle28.Cells(i +
intOffset, 13)
1733:     Next i
1734:     'Übergabe an RFEM
1735:     RFLoad.rfPrepareModification
1736:     RFLoad.rfSetFreeRectangleArr intAnzahlZeilen, FreieRechtecklast(0)
1737:     RFLoad.rfFinishModification
1738:     RFApp.rfUnlockLicence
1739:     'Freigabe der Objekte
1740:     Set RFPos = Nothing
1741:     Set RFTop = Nothing
1742:     Set RFApp = Nothing
1743:     Set RFLoad = Nothing
1744: End Sub
    
```

In der Zeile 1650: ist das Feld FreieRechtecklast() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 2.7 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_FREE\_RECTANGLE.

RF\_LOAD\_FREE\_RECTANGLE ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
direction	Lastrichtung	Benutzerdefinierter Aufzählungstyp RF_LOAD_FREE_DIRECTION
fLoad1Value	Belastung 1	Single
fLoad2Value	Belastung 2	Single
iNo	Nummer der freien Linienlast	Long
p1	Koordinaten der Last am Anfangspunkt	Benutzerdefinierter Typ RF_POINT
p2	Koordinaten der Last am Endpunkt	Benutzerdefinierter Typ RF_POINT
plane	Projektionsebene	Benutzerdefinierter Aufzählungstyp RF_LOAD_FREE_PLANE
strRemark	Kommentar	String
strSurfaceList	Liste der belasteten Flächen	String
type	Lasttyp	Benutzerdefinierter Aufzählungstyp

		typ RF_LOAD_FREE_RECTANGLE_TYPE
--	--	------------------------------------

Tabelle 3.77

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_FREE\_DIRECTION sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LFD_LOCAL_X	Lokal in x
1	LFD_LOCAL_Y	Lokal in y
2	LFD_LOCAL_Z	Lokal in z
3	LFD_GLOBAL_X	Global in X
4	LFD_GLOBAL_Y	Global in Y
5	LFD_GLOBAL_Z	Global in Z
6	LFD_PROJECT_X	Global in X bezogen auf die projizierte Länge
7	LFD_PROJECT_Y	Global in Y bezogen auf die projizierte Länge
8	LFD_PROJECT_Z	Global in Z bezogen auf die projizierte Länge

Tabelle 3.78

RF\_POINT ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
x	X-Koordinate (auf das globale Koordinatensystem bezogen)	Single
y	Y-Koordinate	Single
z	Z-Koordinate	Single

Tabelle 3.79

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_FREE\_PLAN sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LFP_XY	Lastprojektionsebene XY
1	LFP_YZ	Lastprojektionsebene YZ
2	LFP_XZ	Lastprojektionsebene XZ

Tabelle 3.80

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_FREE\_RECTANGLE\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LRT_CONSTANT	Konstant
1	LRT_GRADIENT_X	Veränderlich in X
2	LRT_GRADIENT_Y	Veränderlich in Y
3	LRT_GRADIENT_Z	Veränderlich in Z

Tabelle 3.81

Die Methode `rfSetFreeRectangleArr` übergibt das Feld `FreieRechtecklast()` an RFEM. Mit der Variable `intAnzahlZeilen` wird `rfSetFreeRectangleArr` mitgeteilt, wie viele Zeilen aus dem Feld `FreieRechtecklast()` übertragen werden sollen.

### 3.3.8 Freie Polygonlasten erzeugen

Eine ausführlichere Erläuterung zum Erzeugen von Belastungsdaten ist im Kapitel 3.3.1 (S. 46) zu finden.

In einer bestehenden RFEM-Struktur sollen freie Polygonlasten erzeugt werden.

Voraussetzungen sind, dass mindestens ein Lastfall und die zu belastenden Flächen existieren. Die freie Polygonlasten werden im Excel-Tabellenblatt 2.8 definiert und mit dem Makro `CommandButton29_Click()` an RFEM übertragen.

```

1745: Sub CommandButton29_Click()
1746:     Dim intAnzahlZeilen As Integer
1747:     Dim FreiePolygonlast() As RF_LOAD_FREE_POLYGON
1748:
1749:     Set RFPos = GetObject(, "RFEM.Position")
1750:     Set RFApp = RFPos.rfGetApplication
1751:     RFApp.rfLockLicence
1752:     intLastfallNr = Tabelle29.Cells(2, 6)
1753:     Set RFLoad = RFPos.rfGetLoads.rfGetLoadCase(intLastfallNr - 1,
AT_INDEX)
1754:     Set RFTop = RFPos.rfGetTopology
1755:     intAnzahlZeilen = Tabelle29.Cells(2, 3)
1756:     ReDim FreiePolygonlast(intAnzahlZeilen)
1757:     intOffset = 4
1758:     For i = 0 To intAnzahlZeilen - 1
1759:         FreiePolygonlast(i).iNo = i + 1
1760:         'Spalte A, Flächenliste
1761:         FreiePolygonlast(i).strSurfaceList = Tabelle29.Cells(i +
intOffset, 1)
1762:         'Spalte B, Projektion
1763:         strZelle = Tabelle29.Cells(i + intOffset, 2)
1764:         Select Case strZelle
1765:             Case "XY"
1766:                 FreiePolygonlast(i).plane = LFP_XY
1767:             Case "YZ"
1768:                 FreiePolygonlast(i).plane = LFP_YZ
1769:             Case "XZ"
1770:                 FreiePolygonlast(i).plane = LFP_XZ
1771:             Case Else 'default XY
1772:                 FreiePolygonlast(i).plane = LFP_XY
1773:         End Select
1774:         'Spalte C, Eckpunkte
1775:         FreiePolygonlast(i).strPolygonPointList = Tabelle29.Cells(i +
intOffset, 3)
1776:         'Spalte D, Lastrichtung
1777:         strZelle = Tabelle29.Cells(i + intOffset, 4)
1778:         Select Case strZelle
1779:             Case "Lx"
1780:                 FreiePolygonlast(i).direction = LFD_LOCAL_X
1781:             Case "Ly"
1782:                 FreiePolygonlast(i).direction = LFD_LOCAL_Y
1783:             Case "Lz"
1784:                 FreiePolygonlast(i).direction = LFD_LOCAL_Z
1785:             Case "GX"
1786:                 FreiePolygonlast(i).direction = LFD_GLOBAL_X
1787:             Case "GY"
1788:                 FreiePolygonlast(i).direction = LFD_GLOBAL_Y
1789:             Case "GZ"
1790:                 FreiePolygonlast(i).direction = LFD_GLOBAL_Z
1791:             Case "PX"
1792:                 FreiePolygonlast(i).direction = LFD_PROJECT_X
1793:             Case "PY"
1794:                 FreiePolygonlast(i).direction = LFD_PROJECT_Y
1795:             Case "PZ"
1796:                 FreiePolygonlast(i).direction = LFD_PROJECT_Z
1797:             Case Else 'default GZ
1798:                 FreiePolygonlast(i).direction = LFD_LOCAL_Z
1799:         End Select

```

```

1800:         'Spalte E, Last p
1801:         FreiePolygonlast(i).fLoadValue = Tabelle29.Cells(i +
intOffset, 5)
1802:         'Spalte M, Kommentar
1803:         FreiePolygonlast(i).strRemark = Tabelle29.Cells(i + intOffset,
6)
1804:     Next i
1805:     'Übergabe an RFEM
1806:     RFLoad.rfPrepareModification
1807:     RFLoad.rfSetFreePolygonArr intAnzahlZeilen, FreiePolygonlast(0)
1808:     RFLoad.rfFinishModification
1809:     RFApp.rfUnlockLicence
1810:     'Freigabe der Objekte
1811:     Set RFPos = Nothing
1812:     Set RFTop = Nothing
1813:     Set RFApp = Nothing
1814:     Set RFLoad = Nothing
1815: End Sub
    
```

In der Zeile 1747: ist das Feld FreiePolygonlast() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 2.8 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_FREE\_POLYGON.

RF\_LOAD\_FREE\_POLYGON ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
direction	Lastrichtung	Benutzerdefinierter Aufzählungstyp RF_LOAD_FREE_DIRECTION
fLoadValue	Belastung	Single
fZ	Lastposition in Z	Single
iNo	Nummer der freien Linienlast	Long
plane	Projektionsebene	Benutzerdefinierter Aufzählungstyp RF_LOAD_FREE_PLANE
strPolygonPointList	Liste der Polygonpunkte	String
strRemark	Kommentar	String
strSurfaceList	Liste der belasteten Flächen	String

Tabelle 3.82

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_FREE\_DIRECTION sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LFD_LOCAL_X	Lokal in x
1	LFD_LOCAL_Y	Lokal in y
2	LFD_LOCAL_Z	Lokal in z
3	LFD_GLOBAL_X	Global in X
4	LFD_GLOBAL_Y	Global in Y
5	LFD_GLOBAL_Z	Global in Z
6	LFD_PROJECT_X	Global in X bezogen auf die projizierte Länge
7	LFD_PROJECT_Y	Global in Y bezogen auf die projizierte Länge
8	LFD_PROJECT_Z	Global in Z bezogen auf die projizierte Länge

Tabelle 3.83

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_FREE\_PLAN sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LFP_XY	Lastprojektionsebene XY
1	LFP_YZ	Lastprojektionsebene YZ
2	LFP_XZ	Lastprojektionsebene XZ

Tabelle 3.84

Die Methode rfSetFreePolygonArr übergibt das Feld FreiePolygonlast() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetFreePolygonArr mitgeteilt, wie viele Zeilen aus dem Feld FreiePolygonlast() übertragen werden sollen.

### 3.3.9 Punkt-Zwangsverformung erzeugen

Eine ausführlichere Erläuterung zum Erzeugen von Belastungsdaten ist im Kapitel 3.3.1 (S. 46) zu finden.

In einer bestehenden RFEM-Struktur sollen Punktzwangsverformungen erzeugt werden. Voraussetzungen sind, dass mindestens ein Lastfall und mindestens ein gelagerter Punkt existieren. Die Punktzwangsverformungen werden im Excel-Tabellenblatt 2.9 definiert und mit dem Makro CommandButton30\_Click() an RFEM übertragen.

```

1816: Sub CommandButton30_Click()
1817:   Dim intAnzahlZeilen As Integer
1818:   Dim PunktZwangsverformung() As RF_LOAD_NODAL_DEFORM
1819:
1820:   Set RFPos = GetObject(, "RFEM.Position")
1821:   Set RFApp = RFPos.rfGetApplication
1822:   RFApp.rfLockLicence
1823:   intLastfallNr = Tabelle30.Cells(2, 6)
1824:   Set RFLoad = RFPos.rfGetLoads.rfGetLoadCase(intLastfallNr - 1,
AT_INDEX)
1825:   Set RFTop = RFPos.rfGetTopology
1826:   intAnzahlZeilen = Tabelle30.Cells(2, 3)
1827:   ReDim PunktZwangsverformung(intAnzahlZeilen)
1828:   intOffset = 4 'In der Zeile 4 soll mit dem Auslesen begonnen
werden
1829:   For i = 0 To intAnzahlZeilen - 1
1830:     PunktZwangsverformung(i).iNo = i + 1
1831:     'Spalte A, Punktliste
1832:     PunktZwangsverformung(i).strNodeList = Tabelle30.Cells(i +
intOffset, 1)
1833:     'Spalte B, u-x
1834:     PunktZwangsverformung(i).drift.x = Tabelle30.Cells(i +
intOffset, 2)
1835:     'Spalte C, u-y
1836:     PunktZwangsverformung(i).drift.y = Tabelle30.Cells(i +
intOffset, 3)
1837:     'Spalte D, u-z
1838:     PunktZwangsverformung(i).drift.z = Tabelle30.Cells(i +
intOffset, 4)
1839:     'Spalte E, phi-x
1840:     PunktZwangsverformung(i).moment.x = Tabelle30.Cells(i +
intOffset, 5)
1841:     'Spalte F, phi-y
1842:     PunktZwangsverformung(i).moment.y = Tabelle30.Cells(i +
intOffset, 6)
1843:     'Spalte G, phi-z
1844:     PunktZwangsverformung(i).moment.z = Tabelle30.Cells(i +
intOffset, 7)
1845:     'Spalte H, Kommentar
1846:     PunktZwangsverformung(i).strRemark = Tabelle30.Cells(i +
intOffset, 8)
1847:     Next i
1848:     'Übergabe an RFEM
1849:     RFLoad.rfPrepareModification

```

```

1850:      RFLoad.rfSetNodalDeformArr intAnzahlZeilen,
          PunktZwangsverformung(0)
1851:      RFLoad.rfFinishModification
1852:      RFApp.rfUnlockLicence
1853:      'Freigabe der Objekte
1854:      Set RFPos = Nothing
1855:      Set RFTop = Nothing
1856:      Set RFApp = Nothing
1857:      Set RFLoad = Nothing
1858: End Sub
    
```

In der Zeile 1818: ist das Feld PunktZwangsverformung() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 2.9 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_NODAL\_DEFORM.

RF\_LOAD\_NODAL\_DEFORM ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
drift	Zwangsverschiebung	Benutzerdefinierter Typ RF_POINT
iNo	Nummer der Zwangsverformung	Long
moment	Zwangsverdrehung	Benutzerdefinierter Typ RF_POINT
strNodeList	Liste der Punkte	String
strRemark	Kommentar	String

Tabelle 3.85

RF\_POINT ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
x	Belastung in oder um X	Single
y	Belastung in oder um Y	Single
z	Belastung in oder um Z	Single

Tabelle 3.86

Die Methode rfSetNodalDeformArr übergibt das Feld PunktZwangsverschiebung() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetNodalDeformArr mitgeteilt, wie viele Zeilen aus dem Feld PunktZwangsverschiebung() übertragen werden sollen.

### 3.3.10 Linien-Zwangsverformung erzeugen

Eine ausführlichere Erläuterung zum Erzeugen von Belastungsdaten ist im Kapitel 3.3.1 (S. 46) zu finden.

In einer bestehenden RFEM-Struktur sollen Linienzwangsverformungen erzeugt werden. Voraussetzungen sind, dass mindestens ein Lastfall und mindestens ein gelagerter Linie existieren. Die Linienzwangsverformungen werden im Excel-Tabellenblatt 2.10 definiert und mit dem Makro CommandButton31\_Click() an RFEM übertragen.

```

1859: Sub CommandButton31_Click()
1860:   Dim intAnzahlZeilen As Integer
1861:   Dim LinienZwangsverformung() As RF_LOAD_LINE_DEFORM
1862:
1863:   Set RFPos = GetObject(, "RFEM.Position")
1864:   Set RFApp = RFPos.rfGetApplication
1865:   RFApp.rfLockLicence
1866:   intLastfallNr = Tabelle31.Cells(2, 6)
1867:   Set RFLoad = RFPos.rfGetLoads.rfGetLoadCase(intLastfallNr - 1,
AT_INDEX)
1868:   Set RFTop = RFPos.rfGetTopology
1869:   intAnzahlZeilen = Tabelle31.Cells(2, 3)
1870:   ReDim LinienZwangsverformung(intAnzahlZeilen)
    
```

```

1871:      intOffset = 4   'In der Zeile 4 soll mit dem Auslesen begonnen
          werden
1872:      For i = 0 To intAnzahlZeilen - 1
1873:          LinienZwangsverformung(i).iNo = i + 1
1874:          'Spalte A, Linienliste
1875:          LinienZwangsverformung(i).strLineList = Tabelle31.Cells(i +
          intOffset, 1)
1876:          'Spalte B, u-x,i
1877:          LinienZwangsverformung(i).begin.x = Tabelle31.Cells(i +
          intOffset, 2)
1878:          'Spalte C, u-y,i
1879:          LinienZwangsverformung(i).begin.y = Tabelle31.Cells(i +
          intOffset, 3)
1880:          'Spalte D, u-z,i
1881:          LinienZwangsverformung(i).begin.z = Tabelle31.Cells(i +
          intOffset, 4)
1882:          'Spalte E, u-x,j
1883:          LinienZwangsverformung(i).End.x = Tabelle31.Cells(i +
          intOffset, 5)
1884:          'Spalte F, u-y,j
1885:          LinienZwangsverformung(i).End.y = Tabelle31.Cells(i +
          intOffset, 6)
1886:          'Spalte G, u-z,j
1887:          LinienZwangsverformung(i).End.z = Tabelle31.Cells(i +
          intOffset, 7)
1888:          'Spalte H, Kommentar
1889:          LinienZwangsverformung(i).strRemark = Tabelle31.Cells(i +
          intOffset, 8)
1890:      Next i
1891:      'Übergabe an RFEM
1892:      RFLoad.rfPrepareModification
1893:      RFLoad.rfSetLineDeformArr intAnzahlZeilen,
          LinienZwangsverformung(0)
1894:      RFLoad.rfFinishModification
1895:      RFApp.rfUnlockLicence
1896:      'Freigabe der Objekte
1897:      Set RFPos = Nothing
1898:      Set RFTop = Nothing
1899:      Set RFApp = Nothing
1900:      Set RFLoad = Nothing
1901: End Sub
    
```

In der Zeile 1861: ist das Feld LinienZwangsverformung() definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 2.10 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ RF\_LOAD\_LINE\_DEFORM.

RF\_LOAD\_LINE\_DEFORM ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
begin	Verformung am Anfang der Linie	Benutzerdefinierter Typ RF_POINT
end	Verformung am Ende der Linie	Benutzerdefinierter Typ RF_POINT
iNo	Nummer der Zwangsverformung	Long
strLineList	Liste der Linien	String
strRemark	Kommentar	String

Tabelle 3.87

RF\_POINT ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
x	Verschiebung in X	Single
y	Verschiebung in Y	Single
z	Verschiebung in Z	Single

Tabelle 3.88

Die Methode `rfSetLineDeformArr` übergibt das Feld `LinienZwangsverschiebung()` an RFEM. Mit der Variable `intAnzahlZeilen` wird `rfSetLineDeformArr` mitgeteilt, wie viele Zeilen aus dem Feld `LinienZwangsverschiebung()` übertragen werden sollen.

### 3.3.11 Imperfektion erzeugen

Eine ausführlichere Erläuterung zum Erzeugen von Belastungsdaten ist im Kapitel 3.3.1 (S. 46) zu finden.

In einer bestehenden RFEM-Struktur sollen Imperfektionen erzeugt werden.

Voraussetzungen sind, dass mindestens ein Lastfall und mindestens ein Stab existieren. Die Imperfektionen werden im Excel-Tabellenblatt 2.11 definiert und mit dem Makro `CommandButton32_Click()` an RFEM übertragen.

```

1902: Sub CommandButton32_Click()
1903:     Dim intAnzahlZeilen As Integer
1904:     Dim Imperfektion() As RF_LOAD_IMPERFECTION
1905:
1906:     Set RFPos = GetObject(, "RFEM.Position")
1907:     Set RFApp = RFPos.rfGetApplication
1908:     RFApp.rfLockLicence
1909:     intLastfallNr = Tabelle32.Cells(2, 6)
1910:     Set RFLoad = RFPos.rfGetLoads.rfGetLoadCase(intLastfallNr - 1,
AT_INDEX)
1911:     Set RFTop = RFPos.rfGetTopology
1912:     intAnzahlZeilen = Tabelle32.Cells(2, 3)
1913:     ReDim Imperfektion(intAnzahlZeilen)
1914:     intOffset = 4 'In der Zeile 4 soll mit dem Auslesen begonnen
werden
1915:     For i = 0 To intAnzahlZeilen - 1
1916:         Imperfektion(i).iNo = i + 1
1917:         'Spalte A, Stabliste
1918:         Imperfektion(i).strMemberList = Tabelle32.Cells(i + intOffset,
1)
1919:         'Spalte B, Richtung
1920:         strZelle = Tabelle32.Cells(i + intOffset, 2)
1921:         Select Case strZelle
1922:             Case "2"
1923:                 Imperfektion(i).direction = LID_LOCAL_2
1924:             Case "3"
1925:                 Imperfektion(i).direction = LID_LOCAL_2
1926:             Case Else 'default 3
1927:                 Imperfektion(i).direction = LID_LOCAL_2
1928:         End Select
1929:         'Spalte C, 1/phi
1930:         Imperfektion(i).fDrift = Tabelle32.Cells(i + intOffset, 3)
1931:         'Spalte D, 1/w0
1932:         Imperfektion(i).fCamber = Tabelle32.Cells(i + intOffset, 4)
1933:         'Spalte E, Epsilon
1934:         Imperfektion(i).fRespect = Tabelle32.Cells(i + intOffset, 5)
1935:         'Spalte F, Kommentar
1936:         Imperfektion(i).strRemark = Tabelle32.Cells(i + intOffset, 6)
1937:     Next i
1938:     'Übergabe an RFEM
1939:     RFLoad.rfPrepareModification
1940:     RFLoad.rfSetImperfectionArr intAnzahlZeilen, Imperfektion(0)
1941:     RFLoad.rfFinishModification
1942:     RFApp.rfUnlockLicence
1943:     'Freigabe der Objekte
1944:     Set RFPos = Nothing
1945:     Set RFTop = Nothing
1946:     Set RFApp = Nothing
1947:     Set RFLoad = Nothing
1948: End Sub

```

In der Zeile 1904: ist das Feld `Imperfektion()` definiert. In diesem Feld wird der gesamte Inhalt vom Tabellenblatt 2.11 gespeichert und zum Schluss an RFEM übergeben. Das Feld ist von Typ `RF_LOAD_IMPERFECTION`.

`RF_LOAD_IMPERFECTION` ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
direction	Richtung der Imperfektion	Benutzerdefinierter Aufzählungstyp RF_LOAD_IMPERFECTION_DIRECTION
fCamber	Schiefstellung	Single
fDrift	Vorkrümmung	Single
fRespect	Berücksichtigung der Vorkrümmung an dieser Stabkennzahl	Single
iNo	Nummer der Imperfektion	Long
strMemberList	Stabliste	String
strRemark	Kommentar	String

Tabelle 3.89

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_IMPERFECTION\_DIRECTION sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LID_LOCAL_2	In Richtung der Stabachse 2
1	LID_LOCAL_3	In Richtung der Stabachse 3

Tabelle 3.90

Die Methode rfSetImperfectionArr übergibt das Feld Imperfektion() an RFEM. Mit der Variable intAnzahlZeilen wird rfSetImperfectionArr mitgeteilt, wie viele Zeilen aus dem Feld Imperfektion() übertragen werden sollen.

## 3.4 Steuerung der Berechnung

### 3.4.1 Berechnungsparameter

#### 3.4.1.1 Berechnungsparameter setzen

Das Makro zum Schreiben der Berechnungsparameter unterscheidet sich etwas von den vorhergehenden Makros zum Schreiben von Struktur- und Belastungsdaten. Die Datenstrukturen für die Struktur- und Belastungsdaten waren alle an die Tabellen angelehnt. Diese Tabellen wurden in Excel nachgebildet und mit dem Makro zeilenweise in ein Feld eingelesen.

Das ist bei den Berechnungsparametern nicht mehr notwendig. Es gibt genau einen Satz von Berechnungsparametern, so dass jetzt kein Feld mehr verwendet werden muss sondern einen normale Variable. Bei den Strukturdaten konnten immer mehrere Sätze angelegt werden, z.B. mehrere Flächen. Deswegen war dort ein Feld notwendig.

Die Lizenzverwaltung wurde bereits im Kapitel 3.1.1 Punkte erzeugen (siehe S. 7) erläutern.

In der Tabelle Berechnungsparameter sind alle Daten erfasst, die man im Dialog Berechnungsparameter (Menü Berechnung → Berechnungsparameter) eingeben kann. Die Zellen werden in Makro CommandButton33\_Click() ausgelesen und an RFEM übertragen.

```

1949: Sub CommandButton33_Click()
1950:     'Dim RfRes As RFEM.IrfResults
1951:     Dim ParameterRegister1 As RF_CALC_PARAMS
1952:     Dim ParameterRegister2 As RF_CALC_REACTIVATION
    
```

```
1953: Dim ParameterRegister3 As RF_CALC_RESULTS
1954: Dim ParameterOptions As RF_CALC_OPTIONS
1955: Dim strZelle As String
1956:
1957: Set RFPos = GetObject(, "RFEM.Position")
1958: Set RFApp = RFPos.rfGetApplication
1959: RFApp.rfLockLicence
1960: Set RFTop = RFPos.rfGetTopology
1961: Set RFRes = RFPos.rfGetResults
1962: '1. Register
1963: 'Berechnungsart Lastfälle
1964: strZelle = Tabelle33.Cells(3, 2)
1965: Select Case strZelle
1966:     Case "1"
1967:         ParameterRegister1.lcMethod = METHOD_1ST_ORDER
1968:     Case "2"
1969:         ParameterRegister1.lcMethod = METHOD_2ND_ORDER
1970:     Case "3"
1971:         ParameterRegister1.lcMethod = METHOD_3RD_ORDER
1972:     Case Else
1973:         ParameterRegister1.lcMethod = METHOD_1ST_ORDER
1974: End Select
1975: 'Lastfälle, Ausfallende Lager
1976: strZelle = Tabelle33.Cells(6, 2)
1977: Select Case strZelle
1978:     Case "ja"
1979:         ParameterRegister1.bLCSupportsNL = True
1980:     Case "nein"
1981:         ParameterRegister1.bLCSupportsNL = False
1982:     Case Else
1983:         ParameterRegister1.bLCSupportsNL = True
1984: End Select
1985: 'Lastfälle, Ausfallende Stäbe
1986: strZelle = Tabelle33.Cells(8, 2)
1987: Select Case strZelle
1988:     Case "ja"
1989:         ParameterRegister1.bLCMembersNL = True
1990:     Case "nein"
1991:         ParameterRegister1.bLCMembersNL = False
1992:     Case Else
1993:         ParameterRegister1.bLCMembersNL = True
1994: End Select
1995: 'Lastfälle, Plastische Gelenke
1996: strZelle = Tabelle33.Cells(10, 2)
1997: Select Case strZelle
1998:     Case "ja"
1999:         ParameterRegister1.bLCReleasesNL = True
2000:     Case "nein"
2001:         ParameterRegister1.bLCReleasesNL = False
2002:     Case Else
2003:         ParameterRegister1.bLCReleasesNL = True
2004: End Select
2005: 'Berechnungsart LF-Gruppen
2006: strZelle = Tabelle33.Cells(14, 2)
2007: Select Case strZelle
2008:     Case "1"
2009:         ParameterRegister1.lgMethod = METHOD_1ST_ORDER
2010:     Case "2"
2011:         ParameterRegister1.lgMethod = METHOD_2ND_ORDER
2012:     Case "3"
2013:         ParameterRegister1.lgMethod = METHOD_3RD_ORDER
2014:     Case Else
2015:         ParameterRegister1.lgMethod = METHOD_1ST_ORDER
2016: End Select
2017: 'Lastfälle, Ausfallende Lager
2018: strZelle = Tabelle33.Cells(17, 2)
2019: Select Case strZelle
2020:     Case "ja"
2021:         ParameterRegister1.bLGSupportsNL = True
2022:     Case "nein"
2023:         ParameterRegister1.bLGSupportsNL = False
2024:     Case Else
2025:         ParameterRegister1.bLGSupportsNL = True
2026: End Select
2027: 'Lastfälle, Ausfallende Stäbe
```

```

2028:      strZelle = Tabelle33.Cells(19, 2)
2029:      Select Case strZelle
2030:          Case "ja"
2031:              ParameterRegister1.bLGMembersNL = True
2032:          Case "nein"
2033:              ParameterRegister1.bLGMembersNL = False
2034:          Case Else
2035:              ParameterRegister1.bLGMembersNL = True
2036:      End Select
2037:      'Lastfälle, Plastische Gelenke
2038:      strZelle = Tabelle33.Cells(21, 2)
2039:      Select Case strZelle
2040:          Case "ja"
2041:              ParameterRegister1.bLGReleasesNL = True
2042:          Case "nein"
2043:              ParameterRegister1.bLGReleasesNL = False
2044:          Case Else
2045:              ParameterRegister1.bLGReleasesNL = True
2046:      End Select
2047:      '2. Register
2048:      'Reaktivieren
2049:      strZelle = Tabelle33.Cells(25, 2)
2050:      strZelle1 = Tabelle33.Cells(27, 2)
2051:      Select Case strZelle
2052:          Case "ja"
2053:              ParameterRegister2.bCheckDeform = True
2054:              ParameterRegister2.iMaxReactivations = strZelle1
2055:          Case "nein"
2056:              ParameterRegister2.bCheckDeform = False
2057:          Case Else
2058:              ParameterRegister2.bCheckDeform = True
2059:              ParameterRegister2.iMaxReactivations = strZelle1
2060:      End Select
2061:      'Besondere Behandlung
2062:      strZelle = Tabelle33.Cells(29, 2)
2063:      strZelle1 = Tabelle33.Cells(32, 2)
2064:      strZelle2 = Tabelle33.Cells(34, 2)
2065:      Select Case strZelle
2066:          Case "ja"
2067:              Select Case strZelle1
2068:                  Case "1"      'Stäbe einzeln entfernen
2069:                      ParameterRegister2.handling =
HANDLING_MEMBERS_REMOVED
2070:                  Case "2"      'kleine Steifigkeit zuweisen
2071:                      ParameterRegister2.handling =
HANDLING_MEMBERS_ASSIGNED
2072:                      ParameterRegister2.fReductionFactor = strZelle2
2073:                  Case Else
2074:                      ParameterRegister2.handling =
HANDLING_MEMBERS_ASSIGNED
2075:                      ParameterRegister2.fReductionFactor = strZelle2
2076:              End Select
2077:          Case "nein"
2078:              ParameterRegister2.handling = HANDLING_NONE
2079:          Case Else
2080:
2081:      End Select
2082:      'Register 3
2083:      'Extremale Schnittgrößen
2084:      'N
2085:      strZelle = Tabelle33.Cells(39, 2)
2086:      Select Case strZelle
2087:          Case "ja"
2088:              ParameterRegister3.bIF_N = True
2089:          Case "nein"
2090:              ParameterRegister3.bIF_N = False
2091:          Case Else
2092:              ParameterRegister3.bIF_N = True
2093:      End Select
2094:      'V-2
2095:      strZelle = Tabelle33.Cells(41, 2)
2096:      Select Case strZelle
2097:          Case "ja"
2098:              ParameterRegister3.bIF_V2 = True
2099:          Case "nein"

```

```
2100:             ParameterRegister3.bIF_V2 = False
2101:         Case Else
2102:             ParameterRegister3.bIF_V2 = True
2103:     End Select
2104:     'V-3
2105:     strZelle = Tabelle33.Cells(43, 2)
2106:     Select Case strZelle
2107:         Case "ja"
2108:             ParameterRegister3.bIF_V3 = True
2109:         Case "nein"
2110:             ParameterRegister3.bIF_V3 = False
2111:         Case Else
2112:             ParameterRegister3.bIF_V3 = True
2113:     End Select
2114:     'M-T
2115:     strZelle = Tabelle33.Cells(45, 2)
2116:     Select Case strZelle
2117:         Case "ja"
2118:             ParameterRegister3.bIF_T = True
2119:         Case "nein"
2120:             ParameterRegister3.bIF_T = False
2121:         Case Else
2122:             ParameterRegister3.bIF_T = True
2123:     End Select
2124:     'M-2
2125:     strZelle = Tabelle33.Cells(47, 2)
2126:     Select Case strZelle
2127:         Case "ja"
2128:             ParameterRegister3.bIF_M2 = True
2129:         Case "nein"
2130:             ParameterRegister3.bIF_M2 = False
2131:         Case Else
2132:             ParameterRegister3.bIF_M2 = True
2133:     End Select
2134:     'M-3
2135:     strZelle = Tabelle33.Cells(49, 2)
2136:     Select Case strZelle
2137:         Case "ja"
2138:             ParameterRegister3.bIF_M3 = True
2139:         Case "nein"
2140:             ParameterRegister3.bIF_M3 = False
2141:         Case Else
2142:             ParameterRegister3.bIF_M3 = True
2143:     End Select
2144:     'Ergebnisse bei Starr
2145:     strZelle = Tabelle33.Cells(52, 2)
2146:     Select Case strZelle
2147:         Case "ja"
2148:             ParameterRegister3.bRigid = True
2149:         Case "nein"
2150:             ParameterRegister3.bRigid = False
2151:         Case Else
2152:             ParameterRegister3.bRigid = True
2153:     End Select
2154:     'Ergebnisse bei Kopplung
2155:     strZelle = Tabelle33.Cells(54, 2)
2156:     Select Case strZelle
2157:         Case "ja"
2158:             ParameterRegister3.bCoupling = True
2159:         Case "nein"
2160:             ParameterRegister3.bCoupling = False
2161:         Case Else
2162:             ParameterRegister3.bCoupling = True
2163:     End Select
2164:     'Optionen
2165:     'Gleichungslösermethode
2166:     strZelle = Tabelle33.Cells(58, 2)
2167:     Select Case strZelle
2168:         Case "1"
2169:             ParameterOptions.solver = SM_DIRECT
2170:         Case "2"
2171:             ParameterOptions.solver = SM_ITERATIV
2172:         Case Else
2173:             ParameterOptions.solver = SM_DIRECT
2174:     End Select
```

```

2175:      'Iterationen
2176:      ParameterOptions.iMaxIteration = Tabelle33.Cells(60, 2)
2177:      'Laststeigerungen
2178:      ParameterOptions.iLoadIncrements = Tabelle33.Cells(62, 2)
2179:      'Stabteilung Ergebnisse
2180:      ParameterOptions.iMemberDivisions = Tabelle33.Cells(64, 2)
2181:      'Stabteilung Seile
2182:      ParameterOptions.iMemberDivisions2 = Tabelle33.Cells(66, 2)
2183:      'Schnittgrößen auf deformierte Konstruktion
2184:      strZelle = Tabelle33.Cells(68, 2)
2185:      Select Case strZelle
2186:      Case "ja"
2187:          ParameterOptions.bForcesAndMoments = True
2188:      Case "nein"
2189:          ParameterOptions.bForcesAndMoments = False
2190:      Case Else
2191:          ParameterOptions.bForcesAndMoments = True
2192:      End Select
2193:      'Biegetheorie
2194:      strZelle = Tabelle33.Cells(70, 2)
2195:      Select Case strZelle
2196:      Case "1"
2197:          ParameterOptions.bending = BT_MINDLIN
2198:      Case "nein"
2199:          ParameterOptions.bending = BT_KIRCHHOFF
2200:      Case Else
2201:          ParameterOptions.bending = BT_MINDLIN
2202:      End Select
2203:      'Übergabe an RFEM
2204:      RFRes.rfSetCalcParams ParameterRegister1
2205:      RFRes.rfSetCalcReactivation ParameterRegister2
2206:      RFRes.rfSetCalcResults ParameterRegister3
2207:      RFRes.rfSetCalcOptions ParameterOptions
2208:      RFApp.rfUnlockLicence
2209:      'Freigabe der Objekte
2210:      Set RFPos = Nothing
2211:      Set RFTop = Nothing
2212:      Set RFApp = Nothing
2213:      Set RFRes = Nothing
2214:  End Sub
    
```

In der Zeile 1951: ist eine Record-Variable ParameterRegister1 definiert. In dieser Variablen wird der Inhalt vom Abschnitt Berechnungsparameter – Berechnungsparameter des Tabellenblattes gespeichert. Das Variable ist von Typ RF\_CALC\_PARAMS.

RF\_CALC\_PARAMS ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bLCMembersNL	Berücksichtigung von ausfallenden Stäben bei der Berechnung von Lastfällen, ja, nein	Boolean
bLCReleasesNL	Berücksichtigung von plastischen Gelenken bei der Berechnung von Lastfällen, ja, nein	Boolean
bLCSupportsNL	Berücksichtigung von ausfallenden Lagern bei der Berechnung von Lastfällen, ja, nein	Boolean
bLGMembersNL	Berücksichtigung von ausfallenden Stäben bei der Berechnung von LF-Gruppen, ja, nein	Boolean
bLGReleasesNL	Berücksichtigung von plastischen Gelenken bei der Berechnung von	Boolean

	LF-Gruppen, ja, nein	
bLGSupportsNL	Berücksichtigung von ausfallenden Lagern bei der Berechnung von LF-Gruppen, ja, nein	Boolean
IcMethod	Berechnungsart für Lastfälle	Benutzerdefinierter Aufzählungstyp RF_CALC_METHOD
IgMethod	Berechnungsart für LF-Gruppen	Benutzerdefinierter Aufzählungstyp RF_CALC_METHOD

Tabelle 3.91

Im benutzerdefinierten Aufzählungstyp RF\_CALC\_METHOD sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	METHOD_1ST_ORDER	Theorie I. Ordnung
1	METHOD_2ST_ORDER	Theorie II. Ordnung
2	METHOD_3ST_ORDER	Theorie III. Ordnung

Tabelle 3.92

In der Zeile 1952: ist eine Record-Variable ParameterRegister2 definiert. In dieser Variablen wird der Inhalt vom Abschnitt Berechnungsparameter - Behandlung ausfallender Stäbe des Tabellenblattes gespeichert. Die Variable ist von Typ RF\_CALC\_REACTIVATION.

RF\_CALC\_REACTIVATION ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bCheckDeform	Verformung ausfallender Stäbe überprüfen und ggf. wieder reaktivieren, ja, nein	Boolean
fReductionFactor	Reduktionsfaktor für die Steifigkeit bei besonderer Behandlung	Single
handling	Besondere Behandlung	Benutzerdefinierter Aufzählungstyp RF_HANDLING_TYPE
iMaxReactivations	Maximale Anzahl von Reaktivierungen	Long

Tabelle 3.93

Im benutzerdefinierten Aufzählungstyp RF\_HANDLING\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	HANDLING_NONE	Keine besondere Behandlung
1	HANDLING_MEMBERS_REMOVED	Ausfallenden Stäben wird eine verminderte Steifigkeit (fReductionFactor) zugewiesen.
2	HANDLING_MEMBERS_ASSIGNED	Ausfallende Stäbe werden nacheinander entfernt.

Tabelle 3.94

In der Zeile 1953: ist eine Record-Variable ParameterRegister3 definiert. In dieser Variablen wird der Inhalt vom Abschnitt Berechnungsparameter – Ergebnisse des Tabellenblattes gespeichert. Das Variable ist von Typ RF\_CALC\_RESULTS.

RF\_CALC\_RESULTS ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bCoupling	Ergebnisse an Flächen, die als „Kopplung“ definiert sind, ja, nein	Boolean
bIF_M2	Ermittlung der extremalen Schnittgrößen für M-2	Boolean
bIF_M3	Ermittlung der extremalen Schnittgrößen für M-3	Boolean
bIF_N	Ermittlung der extremalen Schnittgrößen für N	Boolean
bIF_T	Ermittlung der extremalen Schnittgrößen für T	Boolean
bIF_V2	Ermittlung der extremalen Schnittgrößen für V-2	Boolean
bIF_V3	Ermittlung der extremalen Schnittgrößen für V-2	Boolean
bRigid	Ergebnisse an Flächen, die als „Starr“ definiert sind, ja, nein	Boolean

Tabelle 3.95

In der Zeile 1954: ist eine Record-Variable ParameterOptions definiert. In dieser Variablen wird der Inhalt vom Abschnitt Berechnungsparameter – Optionen des Tabellenblattes gespeichert. Das Variable ist von Typ RF\_CALC\_OPTIONS.

RF\_CALC\_OPTIONS ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bending	Biegetheorie	Benutzerdefinierter Aufzählungstyp RF_BENDING_THEORY
bForcesAndMoments	Schnittgrößen auf deformiertes System beziehen, ja, nein	Boolean
iLoadIncrements	Anzahl der Laststeigerungen bei Theorie III. Ordnung	Long
iMaxIteration	Maximale Anzahl der Iterationen	Long
iMemberDivisions	Anzahl der Stabteilungen für Ergebnisse der Stäbe	Long
iMemberDivisions2	Anzahl der Stabteilungen für Seile, Vouten und Bettungsstäbe	Long
solver	Berechnungsmethode	Benutzerdefinierter Aufzählungstyp RF_SOLVER_METHOD

Tabelle 3.96

Im benutzerdefinierten Aufzählungstyp RF\_BENDING\_THEORY sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	BT_MINDLIN	Mindlin
1	BT_KIRCHHOFF	Kirchhoff

Tabelle 3.97

Im benutzerdefinierten Aufzählungstyp RF\_SOLVER\_METHOD sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	SM_DIRECT	Direkt
1	SM_ITERATIV	Iterativ

Tabelle 3.98

Die Methoden rfSetCalcParams, rfSetCalcReactivation, rfSetCalcResults und rfSetCalcOptions übergeben die Variablen ParameterRegister1, ParameterRegister2, ParameterRegister3 und ParameterOptions an RFEM.

## 3.4.2 FE-Netz

Das Erzeugen von FE-Netzverdichtungen ist im Kapitel FE-Netzverdichtungen erzeugen (S. 39) beschrieben.

### 3.4.2.1 FE-Netzparameter setzen

Das Makro zum Scheiben der FE-Netzparameter arbeitet ähnlich dem zum Setzen der Berechnungsparameter (siehe S. 74). Es wird auch nur eine Datensatz in eine Record-Variable eingelesen und diese dann an RFEM übergeben.

In der Tabelle FE-Netzparameter sind alle Daten erfasst, die man im Dialog FE-Netzparameter (Menü Berechnung → FE-Netzparameter) eingeben kann. Die Zellen werden in Makro CommandButton35\_Click() ausgelesen und an RFEM übertragen.

```

2215: Sub CommandButton35_Click()
2216:     Dim FE_NetzParameter As RF_MESH_SETTINGS
2217:     Dim strZelle As String
2218:
2219:     Set RFPos = GetObject(, "RFEM.Position")
2220:     Set RFApp = RFPos.rfGetApplication
2221:     RFApp.rfLockLicence
2222:     'Länge eines Elementes
2223:     FE_NetzParameter.fMemberLength = Tabelle35.Cells(3, 2)
2224:     'Abstand eines Punktes von einer Linien
2225:     FE_NetzParameter.fMaxDistanceNode2Line = Tabelle35.Cells(5, 2)
2226:     'Maximales Verhältnis der Diagonalen
2227:     FE_NetzParameter.fMaxDiagonalRatio = Tabelle35.Cells(7, 2)
2228:     'Maximale Neigung zwischen zwei Elementen
2229:     FE_NetzParameter.fMaxAngleValue = Tabelle35.Cells(9, 2)
2230:     'Netzverdichtung entlang einer Linie (nur bei 2-D)
2231:     strZelle = Tabelle35.Cells(12, 2)
2232:     Select Case strZelle
2233:         Case "ja"
2234:             FE_NetzParameter.bGenRefinLine = True
2235:             'Verhältnis dieser Elemente (nur bei 2-D)
2236:             FE_NetzParameter.fRatioRefinLineSides =
Tabelle35.Cells(14, 2)
2237:         Case "nein"
2238:             FE_NetzParameter.bGenRefinLine = False
2239:         Case Else
2240:             FE_NetzParameter.bGenRefinLine = True
2241:             'Verhältnis dieser Elemente (nur bei 2-D)

```

```

2242:             FE_NetzParameter.fRatioRefinLineSides =
Tabelle35.Cells(14, 2)
2243:         End Select
2244:         'Übergabe an RFEM
2245:         RFPos.rfGetMeshSettings.rfSetData FE_NetzParameter
2246:         RFApp.rfUnlockLicence
2247:         'Freigabe der Objekte
2248:         Set RFPos = Nothing
2249:         Set RFApp = Nothing
2250:     End Sub
    
```

In der Zeile 2216: ist eine Record-Variable FE\_Netzparameter definiert. In dieser Variablen wird der Inhalt des Tabellenblattes FE-Netzparameter gespeichert. Das Variable ist von Typ RF\_MESH\_SETTINGS.

RF\_MESH\_SETTINGS ist ein Record mit folgendem Inhalt:

Element	Bedeutung	Typ
bGenRefinLine	Soll bei entlang der Randlinie eine FE-Netzverdichtung erzeugt werden, ja, nein.  Das Setzen dieser Eigenschaft hat nur Auswirkungen bei 2-D-Positionen.	Boolean
fMaxAngleValue	Maximale Neigung zwischen zwei Elementen [°]	Single
fMaxDiagonalRatio	Maximales Verhältnis der Diagonalen	Single
fMaxDistanceNode2Line	Abstand eines Punktes von einer Linien, um in diese integriert zu werden [m]	Single
fMemberLength	Angestrebte Maschenweite [m]	Single
fRatioRefinLineSides	Seitenverhältnis der Elemente bei einer Randverdichtung.  Das Setzen dieser Eigenschaft hat nur Auswirkungen bei 2-D-Positionen.	Single

Tabelle 3.99

Die Methode rfGetMeshSettings.rfSetData übergibt die Variable FE\_NetzParameter an RFEM.

### 3.4.2.2 FE-Netz generieren

Das Makro zum Generieren des FE-Netzes ist sehr einfach. Es müssen keine Daten an RFEM übergeben werden sondern nur eine Aktion ausgeführt werden.

Auf dem Tabelleblatt FE-Netz generieren befindet sich nur eine Schaltflächen, die das folgende Makro startet.

```

2251: Sub CommandButton36_Click()
2252:     Set RFPos = GetObject(, "RFEM.Position")
2253:     Set RFApp = RFPos.rfGetApplication
2254:     RFApp.rfLockLicence
2255:     RFPos.rfRegenerateMesh
2256:     RFApp.rfUnlockLicence
2257:     'Freigabe der Objekte
2258:     Set RFPos = Nothing
2259:     Set RFApp = Nothing
2260: End Sub
    
```

Die Methode rfRegenerateMesh generiert das FE-Netz.

### 3.4.3 Starten der Berechnung

#### 3.4.3.1 Gezielte Ergebnisse berechnen

In der Tabelle Gezielt berechnen wird erfasst, ob ein Lastfall, eine Lastfallgruppe oder eine Lastfallkombination berechnet werden soll. Außerdem wird die Nummer eingegeben. Die Daten werden in Makro CommandButton37\_Click() ausgelesen und an RFEM übertragen.

```

2261: Sub CommandButton37_Click()
2262:   Dim Typ As RF_LOAD_TYPE
2263:   Dim intNummer As Long
2264:   Dim strZelle As String
2265:   '#####
2266:   Set RFPos = GetObject(, "RFEM.Position")
2267:   Set RFApp = RFPos.rfGetApplication
2268:   RFApp.rfLockLicence
2269:   Set RFRes = RFPos.rfGetResults
2270:   strZelle = Tabelle37.Cells(3, 2)
2271:   Select Case strZelle
2272:     Case "LF"
2273:       Typ = LT_CASE
2274:     Case "LG"
2275:       Typ = LT_GROUP
2276:     Case "LK"
2277:       Typ = LT_COMBINATION
2278:     Case Else
2279:       Typ = LT_CASE
2280:   End Select
2281:   intNummer = Tabelle37.Cells(5, 2)
2282:   'Übergabe an RFEM
2283:   RFRes.rfCalculate Typ, intNummer
2284:   RFApp.rfUnlockLicence
2285:   'Freigabe der Objekte
2286:   Set RFRes = Nothing
2287:   Set RFPos = Nothing
2288:   Set RFApp = Nothing
2289: End Sub
    
```

Das was berechnet werden soll, wird in der Aufzählungsvariable Typ gespeichert. In intNummer wird die Nummer des LF, der LG oder der LK gespeichert. Die Variable Typ ist durch den benutzerdefinierten Aufzählungstyp RF\_LOAD\_TYPE definiert.

Im benutzerdefinierten Aufzählungstyp RF\_LOAD\_TYPE sind folgende Konstanten definiert:

Zahlenwert	Konstante	Bedeutung
0	LT_CASE	Ein Lastfall soll berechnet werden.
1	LT_GROUP	Eine Lastfallgruppe soll berechnet werden.
2	LT_COMBINATION	Eine Lastfallkombination soll berechnet werden.

Tabelle 3.100

In der Variablen intNummer wird die Nummer des Lastfalls, der Lastfallgruppe oder der Lastfallkombination gespeichert.

Die Methode rfCalculate überträgt die Variablen Typ und intNummer an RFEM und startet die Berechnung. Mit dieser Methode kann immer nur die Berechnung eines LF, einer LG oder LK gestartet werden.

#### 3.4.3.2 Alle Ergebnisse berechnen

Das Makro zum Starten der Berechnung ist wieder sehr einfach. Es müssen keine Daten an RFEM übergeben werden sondern nur eine Aktion ausgeführt werden.

Auf dem Tabelleblatt Alles berechnen befindet sich nur eine Schaltflächen, die das folgende Makro startet.

```
2290: Sub CommandButton38_Click()  
2291:     Set RFPos = GetObject(, "RFEM.Position")  
2292:     Set RFApp = RFPos.rfGetApplication  
2293:     RFApp.rfLockLicence  
2294:     Set RFRes = RFPos.rfGetResults  
2295:     RFRes.rfCalculateAll  
2296:     RFApp.rfUnlockLicence  
2297:     'Freigabe der Objekte  
2298:     Set RFRes = Nothing  
2299:     Set RFPos = Nothing  
2300:     Set RFApp = Nothing  
2301: End Sub
```

Die Methode rfCalculateAll startet die Berechnung aller LF, LG und LK von denen noch keine Ergebnisse vorliegen.

# A: Literatur

- [1] HELD, B.: Excel-VBA Kompendium, Markt + Technik, München, 2003
- [2] MONADJEMI, P.: Visual Basic 6, Markt + Technik, München, 2000

## B: Wichtige VBA-Befehle

In diesem Anhang sollen die in den Beispielen verwendeten VBA-Befehle kurz erläutert werden, so dass die Quelltexte auch dann verstanden werden können, wenn man keine Erfahrungen in der VBA-Programmierung hat. Zum tieferen Einstieg in die VBA-Programmierung sei die Onlinehilfe von Excel empfohlen.

### Dim, ReDim

Mit der Dim-Anweisung werden Variablen deklariert und es wird Speicherplatz reserviert. Im Unterschied zu C oder Pascal ist es in VBA nicht zwingend notwendig, Variablen zu deklarieren. Anders ist es bei Feldern. Diese müssen deklariert werden.

Grundsätzlich gibt es in VBA zwei Arten von Datenfeldern:

- Felder mit fester Größe
- Dynamische Felder

#### Die Anweisung

`Dim FestesDatenfeld(10, 10) As Integer`  
legt die Größe des Feldes entgeltig fest. Mit

`Dim DynamischesDatenfeld() As Integer`

wird keine Größe des Feldes festgelegt. Mit `ReDim` kann die Größe des Feldes verändert werden. Das kann beliebig oft geschehen. Zu beachten ist jedoch, dass bei `ReDim` der Inhalt des Feldes verloren geht.

### For-Schleife

Die For-Schleife ist eine Zählschleife, die auch in ähnlicher Form in fast jeder anderen Programmiersprache wiederzufinden ist.

Die Anweisungen zwischen For und Next werden so oft ausgeführt, bis die Zählvariable den Wert hinter To erreicht hat. Die Integer-Variable wird standardmäßig um 1 erhöht. Mit Step kann auch ein anderer Wert festgelegt werden.

### Select-Case-Anweisung

Die Select-Case-Anweisung ist eine Mehrfachverzweigung. Grundsätzlich könnte diese auch durch eine Folge von If-Anweisungen ersetzt werden. Select-Case ist jedoch wesentlich eleganter und übersichtlicher.

Beispiel mit Select-Case:

```
2302: ...
2303: Select Case strZelle
2304:   Case "Polylinie"
2305:     Linie(i).lineType = LT_POLYLINE
2306:   Case "Bogen"
2307:     Linie(i).lineType = LT_ARC
2308:   Case "Kreis"
2309:     Linie(i).lineType = LT_CIRCLE
2310:   Case "Spline"
2311:     Linie(i).lineType = LT_SPLINE_CUB
2312:   Case "B-Spline"
2313:     Linie(i).lineType = LT_SPLINE_B
2314:   Case "Bezier-Spline"
2315:     Linie(i).lineType = LT_SPLINE_BEZ
2316:   Case Else 'default Polylinie
2317:     Linie(i).lineType = LT_POLYLINE
2318: End Select
2319: ...
```

Beispiel mit If:

```
2320: ...
```

```
2321: If strZelle="Polylinie" Then
2322:   Linie(i).lineType = LT_POLYLINE
2323: ElseIf strZelle="Bogen" Then
2324:   Linie(i).lineType = LT_ARC
2325: ElseIf strZelle="Kreis" Then
2326:   Linie(i).lineType = LT_CIRCLE
2327: ElseIf strZelle=""Spline" Then
2328:   Linie(i).lineType = LT_SPLINE_CUB
2329: ElseIf strZelle=""B-Spline" Then
2330:   Linie(i).lineType = LT_SPLINE_B
2331: ElseIf strZelle=""Bezier-Spline" Then
2332:   Linie(i).lineType = LT_SPLINE_BEZ
2333: Else
2334:   Linie(i).lineType = LT_POLYLINE
2335: End if
2336: ...
```

Mit der Anweisung in Zeile 2316: können Eingabefehler abgefangen werden. Wenn die Variable strZelle einen beliebigen anderen Wert hat, dann wird dieser Zweige ausgeführt.

## Cells-Eigenschaft

Diese Eigenschaft kann für zwei verschiedene Aufgaben eingesetzt werden. Mit

```
2337: intAnzahlZeilen = Tabelle2.Cells(2, 3)
```

wird im Tabellenblatt Tabelle2 die Zelle in der Zeile 2 Spalte 3 ausgelesen und der Variablen intAnzahlZeilen zugewiesen.

In der anderen Richtung funktioniert das auch. Mit

```
2338: Tabelle2.Cells(2, 3)=intAnzahlZeilen
```

Wird in die Zeile 2 Spalte 3 der Tabelle 2 der Wert der Variablen intAnzahlZeilen geschrieben.

# C: Index

## A

axis .....	34
AXIS_2 .....	35
AXIS_3 .....	35

## B

bCalcCol.....	20
bCheckDeform .....	79
bCoupling.....	80
bDividing .....	44
bDWRespect .....	43
begin .....	72
bending .....	80
bEntire.....	51, 55
bForcesAndMoments.....	80
bGenRefinLine .....	82
bIF_M2.....	80
bIF_M3.....	80
bIF_N .....	80
bIF_T .....	80
bIF_V2.....	80
bIF_V3.....	80
bLCMembersNL .....	78
bLCReleasesNL .....	78
bLCSupportsNL .....	78
bLGMembersNL .....	78
bLGReleasesNL.....	78
bLGSupportsNL.....	79
bRelative.....	50, 51, 55
bRigid .....	80
BT_KIRCHHOFF.....	81
BT_MINDLIN.....	81
bTension.....	45
BY_ANGLE.....	35
BY_MATRIX .....	16
BY_NODE .....	35
BY_THICK .....	16

## C

column .....	19
COLUMN_CIRC.....	21
COLUMN_RECT .....	21

COLUMN_SUPP_ELAST .....	21
COLUMN_SUPP_RELEASE .....	21
COLUMN_SUPP_RIGID .....	21
CS_CARTESIAN .....	9
CS_POLAR.....	9
CS_SPHERICAL .....	9
CS_XCYLINDRICAL .....	9
CS_YCYLINDRICAL .....	9
CS_ZCYLINDRICAL .....	9
csType .....	8, 9

## D

dArea .....	14, 17
direction.....	49, 53, 58, 60, 63, 66, 69, 74
distanceArr .....	32
downSupport .....	20
drift .....	71
DWDir .....	43

## E

end.....	72
ET_NONE .....	20, 23, 25, 37
ET_NOPRES.....	20, 23, 25, 37
ET_NOTENS .....	20, 23, 25, 37
exclusion .....	25, 36
exclusionX .....	19, 23
exclusionY .....	19, 23
exclusionZ .....	19, 23

## F

fA .....	50, 51, 55
fA1 .....	28
fA2 .....	28
fA3 .....	28
fAlongX.....	19, 23
fAlongY .....	19, 23
fAlongZ .....	19, 23
fAlpha .....	12, 28
fAngle .....	15
fAroundX .....	19, 23
fAroundY.....	19, 23
fAroundZ.....	19, 23
fB51, 55	

fC11 .....	36	fNode1Value .....	58
fC12 .....	36	fNode2Value .....	58
fC13 .....	37	fNode3Value .....	58
fC1x .....	25	force .....	47
fC1y .....	25	fOrtoD11 .....	16
fC1z .....	25	fOrtoD12 .....	16
fC21 .....	37	fOrtoD22 .....	16
fC22 .....	37	fOrtoD33 .....	16
fC23 .....	37	fOrtoD44 .....	16
fCamber .....	74	fOrtoD55 .....	16
fCfi .....	37	fP1 .....	50, 51, 54, 55
fDrift .....	74	fP2 .....	51, 55
fDWFactor .....	43	fParam1 .....	38
fE12 .....		fParam2 .....	38
fE1x .....	31	fParam3 .....	38
fE1y .....	31	fParam4 .....	39
fE1z .....	31	fParam5 .....	39
fE2x .....	31	fParam6 .....	39
fE2y .....	31	FPD_GLOBAL_X .....	61
fE2z .....	31	FPD_GLOBAL_Y .....	61
fG .....	12	FPD_GLOBAL_Z .....	61
fGamma_M .....	45	FPD_LOCAL_X .....	61
fH .....	49	FPD_LOCAL_Y .....	61
fHeight .....	20	FPD_LOCAL_Z .....	61
fI2 .....	28	FPT_FORCE .....	61
fI3 .....	28	FPT_MOMENT .....	61
fIT .....	28	fRatioRefinLineSides .....	82
fLCFactor .....	43	fReductionFactor .....	79
fLGFactor .....	45	fRespect .....	74
fLoad1Value .....	63, 66	fRo .....	12
fLoad2Value .....	63, 66	fRotAngle .....	23
fLoadValue .....	60, 69	fRotation .....	34
FLT_CONSTANT .....	64	fRotX .....	19
FLT_LINEAR .....	64	fRotY .....	19
fMaxAngleValue .....	82	fRotZ .....	20
fMaxDiagonalRatio .....	82	fSizeE .....	40
fMaxDistanceNode2Line .....	82	fSizeL .....	41
fMemberLength .....	82	fSizeO .....	41
fMemD11 .....	16	fSizeX .....	20
fMemD12 .....	16	fSizeY .....	20
fMemD22 .....	16	fSuppElast .....	20
fMemD33 .....	16	fThickX .....	15
fMi .....	12	fThickY .....	15

fTorsSolid.....	15	isColumn .....	20
fZ69		iSide .....	26
<b>H</b>		isLocal .....	23, 31
handling .....	79	iSurfaceNo.....	26
HANDLING_MEMBERS_ASSIGNED .....	79	<b>L</b>	
HANDLING_MEMBERS_REMOVED .....	79	LCD_X.....	43
HANDLING_NONE .....	79	LCD_Y.....	43
<b>I</b>		LCD_Z.....	43
iBeginCrossSectionNo .....	34	IcMethod.....	79
iBeginReleaseNo .....	34	LCT_EXCEPTION.....	43
iBLineCount .....	14, 17	LCT_IMPERFECTION .....	43
iCount.....	32	LCT_PERMANENT.....	43
iCurveNo.....	34	LCT_RELIEVE_PERMANENT .....	43
iEccentricityNo .....	34	LCT_RELIEVE_VARIABLE .....	43
iEndCrossSectionNo.....	34	LCT_VARIABLE .....	43
iEndReleaseNo .....	34	LD_GLOBAL_X .....	50, 54, 58
iFoundNo .....	34	LD_GLOBAL_Y.....	50, 54, 58
iHoleCount.....	14	LD_GLOBAL_Z.....	50, 54, 58
iLineCount .....	14	LD_LOCAL_1 .....	50, 54, 58
iNodeCount .....	14	LD_LOCAL_2 .....	50, 54, 58
iLineNo .....	26	LD_LOCAL_3 .....	50, 54, 58
iLoadIncrements .....	80	LD_PROJECT_X.....	50, 54, 58
iMaterialNo.....	14, 20, 28	LD_PROJECT_Y.....	50, 54, 58
iMaterialX .....	15	LD_PROJECT_Z.....	50, 54, 58
iMaterialY .....	15	length.....	11
iMaxIteration .....	80	LFD_GLOBAL_X.....	64, 67, 69
iMaxReactivations .....	79	LFD_GLOBAL_Y.....	64, 67, 69
iMemberDivisions .....	80	LFD_GLOBAL_Z.....	64, 67, 69
iMemberDivisions2 .....	80	LFD_LOCAL_X.....	64, 67, 69
iMethod.....	45	LFD_LOCAL_Y.....	64, 67, 69
iNo 8, 9, 11, 12, 14, 15, 16, 17, 20, 23, 25, 26, 28, 29, 31, 32, 34, 37, 39, 41, 43, 45, 46, 47, 49, 53, 58, 60, 63, 66, 69, 71, 72, 74		LFD_LOCAL_Z .....	64, 67, 69
iNode1No .....	15, 58	LFD_PROJECT_X.....	64, 67, 69
iNode2No .....	15, 58	LFD_PROJECT_Y.....	64, 67, 69
iNode3No .....	15, 58	LFD_PROJECT_Z .....	64, 67, 69
iNodeCount .....	11	LFP_XY.....	61, 64, 67, 70
inSurfaceNo.....	17	LFP_XZ.....	61, 64, 67, 70
intAnzahlZeilen.....	8, 9	LFP_YZ.....	61, 64, 67, 70
intOffset .....	9	IgMethod .....	79
iNum.....	41	LID_LOCAL_2 .....	74
iPartitionNo .....	34	LID_LOCAL_3 .....	74
iRefNodeNo .....	9	lineType.....	11
iRibNo.....	34	Lizenzverwaltung .....	9
		LLT_FORCE.....	51, 55

LLT_MOMENT .....	55	MT_COMPRESSION .....	35
LMT_DEFORM1 .....	51	MT_DUMMY .....	35
LMT_DEFORM2 .....	51	MT_EULER .....	35
LMT_MOMENT .....	51	MT_LINE_DIVISION .....	41
LMT_THERM1 .....	51	MT_LINE_LENGTH .....	41
LMT_THERM2 .....	51	MT_NODE .....	41
LRT_CONSTANT .....	67	MT_RIB .....	35
LRT_GRADIENT_X .....	67	MT_RIGID_TYPE_HH .....	35
LRT_GRADIENT_Y .....	67	MT_RIGID_TYPE_HR .....	35
LRT_GRADIENT_Z .....	67	MT_RIGID_TYPE_RH .....	35
LS_CONSTANT .....	50, 54	MT_RIGID_TYPE_RR .....	35
LS_SINGLE .....	50, 54	MT_SURFACE .....	41
LS_TRAPEZ .....	50, 54	MT_TENSION .....	35
LSS_GRADIENT_X .....	59	MT_TRUSS_FALSE .....	35
LSS_GRADIENT_Y .....	59	MT_TRUSS_TRUE .....	35
LSS_GRADIENT_Z .....	59	<b>N</b>	
LSS_TRIANGLE .....	58	n 26, 29	
LSS_UNIFORM .....	58	NL_EPRESS .....	39
LST_DEFORM1 .....	59	NL_EPRESS_GAP .....	39
LST_DEFORM2 .....	59	NL_EPRESS_P .....	39
LST_FORCE .....	59	NL_EPRESS_S .....	39
LST_THERM1 .....	59	NL_ETENS .....	39
LST_THERM2 .....	59	NL_ETENS_GAP .....	39
LT_ARC .....	11	NL_ETENS_P .....	39
LT_CASE .....	83	NL_ETENS_S .....	39
LT_CIRCLE .....	11	NL_GAP_SIDES .....	39
LT_COMBINATION .....	83	NL_LIM_P .....	39
LT_GROUP .....	83	NL_LIM_S .....	39
LT_POLYLINE .....	11	NL_NO .....	39
LT_SPLINE_B .....	11	NL_PLAST_RELEASE .....	39
LT_SPLINE_BEZ .....	11	<b>O</b>	
LT_SPLINE_CUB .....	11	order .....	20
<b>M</b>		ORDER_XYZ .....	21
m2 .....	26, 29	ORDER_XZY .....	21
m3 .....	26, 29	ORDER_YXZ .....	20
mem .....	16	ORDER_YZX .....	20
METHOD_1ST_ORDER .....	79	ORDER_ZXY .....	20
METHOD_2ST_ORDER .....	79	ORDER_ZYX .....	20
METHOD_3ST_ORDER .....	79	orto .....	16
moment .....	47, 71	<b>P</b>	
mt .....	26, 29	p1 .....	60, 63, 66
MT_BEAM .....	35	p2 .....	63, 66
MT_CABLE .....	35	plane .....	60, 63, 66, 69

**Q**

q2 ..... 26, 29

q3 ..... 27, 30

**R**

ReDim ..... 9

RF\_SUPPORT\_LINE ..... 22

RF\_SUPPORT\_NODE ..... 19

RF\_BENDING\_THEORY ..... 80, 81

RF\_CALC\_METHOD ..... 79

RF\_CALC\_OPTIONS ..... 80

RF\_CALC\_PARAMS ..... 78

RF\_CALC\_REACTIVATION ..... 79

RF\_CALC\_RESULTS ..... 80

RF\_COLUMN ..... 19, 20

RF\_COLUMN\_SUPP ..... 20, 21

RF\_COLUMN\_TYPE ..... 20, 21

RF\_CROSS\_SECTION ..... 28

RF\_CS\_TYPE ..... 9

RF\_ECCENTRICITY ..... 31

RF\_EXCLUSION\_TYPE ..... 19, 20, 23, 25, 36, 37

RF\_FOUND\_MEMBER ..... 36

RF\_FOUND\_SURFACE ..... 24

RF\_HANDLING\_TYPE ..... 79

RF\_HOLE ..... 17

RF\_LINE ..... 11

RF\_LINE\_TYPE ..... 11

RF\_LOAD\_CASE ..... 42

RF\_LOAD\_CASE\_DIRECTION ..... 43

RF\_LOAD\_CASE\_TYPE ..... 43

RF\_LOAD\_COMBINATION ..... 46

RF\_LOAD\_DIRECTION ..... 49, 50, 53, 54, 58

RF\_LOAD\_FREE\_DIRECTION .. 63, 64, 66, 67, 69

RF\_LOAD\_FREE\_LINE ..... 63

RF\_LOAD\_FREE\_LINE\_TYPE ..... 63, 64

RF\_LOAD\_FREE\_PLAN ..... 61, 64, 67, 70

RF\_LOAD\_FREE\_PLANE ..... 60, 63, 66, 69

RF\_LOAD\_FREE\_POINT ..... 60

RF\_LOAD\_FREE\_POINT\_DIRECTION ..... 60, 61

RF\_LOAD\_FREE\_POINT\_TYPE ..... 61

RF\_LOAD\_FREE\_POLYGON ..... 69

RF\_LOAD\_FREE\_RECTANGLE ..... 66

RF\_LOAD\_FREE\_RECTANGLE\_TYPE ..... 66, 67

RF\_LOAD\_GROUP ..... 44

RF\_LOAD\_IMPERFECTION ..... 73

RF\_LOAD\_IMPERFECTION\_DIRECTION ..... 74

RF\_LOAD\_LINE\_DEFORM ..... 72

RF\_LOAD\_LINE\_FORCE ..... 53

RF\_LOAD\_LINE\_TYPE ..... 54

RF\_LOAD\_MEMBER ..... 49

RF\_LOAD\_MEMBER\_TYPE ..... 50, 51

RF\_LOAD\_NODAL\_DEFORM ..... 71

RF\_LOAD\_NODAL\_FORCE ..... 47

RF\_LOAD\_STYLE ..... 49, 50, 54

RF\_LOAD\_STYLE\_CONSTANT ..... 49, 50, 54

RF\_LOAD\_STYLE\_SINGLE ..... 49, 50, 54

RF\_LOAD\_STYLE\_TRAPEZ ..... 50, 51, 54, 55

RF\_LOAD\_SURFACE ..... 57

RF\_LOAD\_SURFACE\_STYLE ..... 58

RF\_LOAD\_SURFACE\_TYPE ..... 58

RF\_LOAD\_SURFACE\_TYPE ..... 59

RF\_LOAD\_TYPE ..... 83

RF\_MATERIAL ..... 12

RF\_MEM ..... 16

RF\_MEMBER ..... 34

RF\_MEMBER\_PART ..... 32

RF\_MEMBER\_ROTATION\_NODE ..... 34

RF\_MEMBER\_ROTATION\_TYPE ..... 34, 35

RF\_MEMBER\_TYPE ..... 34, 35

RF\_MESH ..... 40

RF\_MESH\_SETTINGS ..... 82

RF\_NL ..... 38

RF\_NL\_TYPE ..... 39

RF\_NODE ..... 8

RF\_NODE\_ROTATION\_AXIS ..... 34, 35

RF\_ORTO ..... 16

RF\_ORTO\_THICK\_BY ..... 16

RF\_POINT ..... 47, 60, 61, 63, 64, 66, 67, 71, 72

RF\_RELEASE\_MEMBER ..... 29

RF\_RELEASE\_SIDE ..... 26, 27

RF\_RELEASE\_SURFACE ..... 26

RF\_ROT\_ORDER ..... 20

RF\_SOLVER\_METHOD ..... 80, 81

RF\_SURFACE ..... 14

RF\_SURFACE\_TYPE ..... 14, 15

RF\_THICK\_LIN ..... 14, 15

RF\_THICK\_ORTO ..... 15

RF_THICK_TYPE .....	15, 16	RFTop .....	8, 9
RFApp .....	8	rotNode.....	34
rfCalculate .....	83	rotType.....	34
rfCalculateAll .....	84	RS_LEFT .....	27
rfFinishModification.....	10	RS_NONE.....	27
rfGetLoadCase .....	47	RS_RIGHT.....	27
rfGetMeshSettings.rfSetData .....	82	<b>S</b>	
RFLoad .....	8, 47	SM_DIRECT.....	81
rfPrepareModification.....	10	SM_ITERATIV .....	81
rfRegenerateMesh .....	83	solver.....	80
RRes .....	8	ST_BSPLINE.....	15
rfSetCalcOptions.....	81	ST_LINEAR .....	15
rfSetCalcParams.....	81	ST_PIPELINE.....	15
rfSetCalcReactivation .....	81	ST_QUAD.....	15
rfSetCalcResults .....	81	ST_ROT .....	15
rfSetCrossSectionArr .....	28	strBLineList .....	14, 18
rfSetEccentricityArr .....	31	strDefinition .....	45, 46
rfSetFoundMemberArr.....	37	strDescription .....	12, 43, 45, 46
rfSetFoundSurfArr.....	25	strHoleList .....	14
rfSetFreeLineArr .....	64	strLineList .....	14
rfSetFreePointArr .....	62	strNodeList .....	14
rfSetFreePolygonArr.....	70	strLineList .....	23, 53, 72
rfSetFreeRectangleArr .....	68	strMemberList .....	37, 39, 49, 74
rfSetHoleArr .....	18	strName .....	28
rfSetImperfectionArr.....	74	strNodeList .....	11, 20, 47, 71
rfSetLineArr.....	11, 55	strObjectList .....	41
rfSetLineDeformArr.....	73	strPolygonPointList.....	69
rfSetLoadCaseArr .....	43	strRemark 9, 11, 12, 14, 15, 16, 18, 20, 23, 25, 27, 28, 30, 31, 32, 34, 37, 39, 41, 43, 47, 49, 53, 58, 61, 63, 66, 69, 71, 72, 74	
rfSetLoadCombinationArr.....	46	strSurfaceList.....	25, 58, 61, 63, 66, 69
rfSetLoadGroupArr .....	45	style.....	49, 54, 58
rfSetMaterialArr.....	13	styleConstant .....	49, 54
rfSetMemberArr.....	35, 51	styleSingle .....	49, 54
rfSetMemberPartArr .....	32	styleTrapez .....	50, 54
rfSetMeshArr .....	41	surfaceType .....	14
rfSetNLArr.....	39	<b>T</b>	
rfSetNodalDeformArr.....	71	thick1 .....	15
rfSetNodalForceArr .....	47	thick2 .....	15
rfSetNodeArr .....	10	thick3 .....	15
rfSetReleaseMemberArr .....	30	thickBy .....	16
rfSetReleaseSurfArr .....	27	thickConst .....	14
rfSetSuppLineArr .....	23	thickLin.....	14
rfSetSuppNodeArr .....	21	thickOrto .....	15
rfSetSurfaceArr .....	17, 59		

thickType .....	15	TT_ORTO.....	16
TT_CONST .....	16	type..20, 34, 39, 41, 43, 50, 54, 58, 61, 63, 66	
TT_LINEAR.....	16	Type Library.....	8