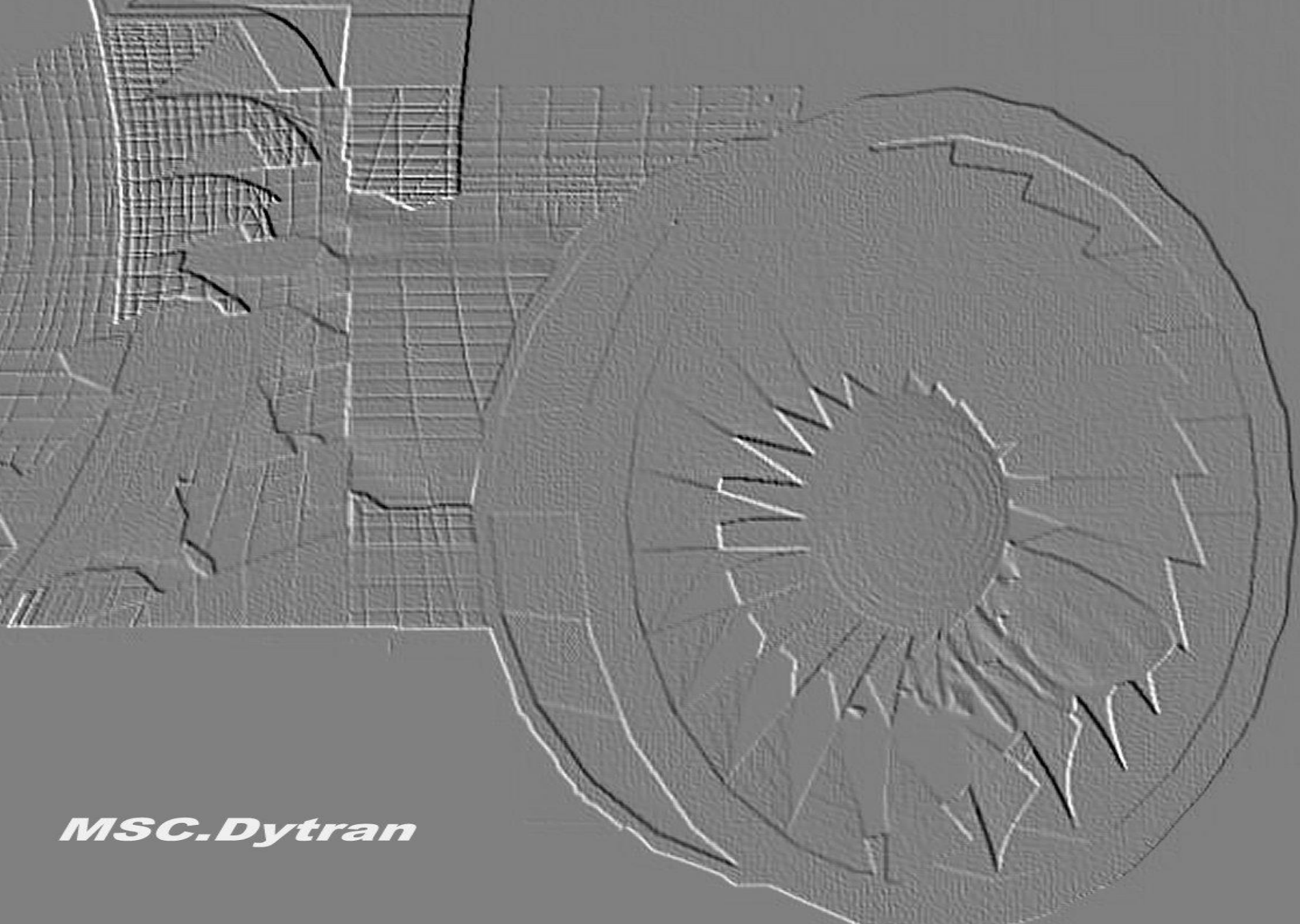


**CHAPTER 10 - LAGRANGIAN BOUNDARY CONDITIONS**



**MSC.Dytran**

## СОДЕРЖАНИЕ

- ❑ **Закрепление узлов – SPCn**
- ❑ **Вынужденное перемещение – FORCE/MOMENT**
- ❑ **Жёсткие стенки – WALL**
- ❑ **Элементы-связи – RCONN**
- ❑ **Элементы – жёсткие тела**
  - RBE2
  - KJOIN
  - VJOIN

## ЗАКРЕПЛЕНИЕ УЗЛОВ

- ❑ Предотвращает движение узла в указанном направлении
- ❑ Операторы Bulk Data, “закрепляющие” узлы, должны быть инициированы оператором Case Control
  - SPC = SID
    - Операторы SPCn раздела Bulk Data, не инициированные операторами Case Control, будут игнорированы
- ❑ Операторы SPC и SPC1 “закрепляют” узлы в той системе координат, в которой вычисляются их перемещения (*в MSC.Dytran – это система координат, в которой задано расположение узлов*)
- ❑ Закрепления узлов могут использоваться для моделирования граничных условий и условий симметрии
- ❑ Закрепляемые компоненты перемещений кодируются цифрами от 1 до 6, например 23 или 156
  - SPC = 100
  - BEGIN BULK
  - ...
  - SPC, 100, 27, 123
  - SPC1, 100, 156, 19, THRU, 28

## “ВРАЩАТЕЛЬНЫЕ” ГРАНИЧНЫЕ УСЛОВИЯ

- ❑ Оператор SPC2 используется для задания окружной и радиальной скоростей узлов
- ❑ Операторы SPC2 должны быть инициированы соответствующим оператором Case Control
- ❑ Пример:

1	2	3	4	5	6	7	8	9	10
SPC2	SID	G		TYPE1	VALUE1	NX	NY	NZ	+CONT1
SPC2	1	10		CONSTANT	100.	1.	1.	0.	+CONT1
+CONT1	TYPE2	VALUE2							+CONT2
+CONT1	CONSTANT	0.							+CONT2
+CONT2	G1	G2	THRU	G3	BY	G5	-etc-		
+CONT2	10	13	THRU	56	BY	4			

Узел на оси вращения

Величина и направление угловой скорости

Характеристика закрепления в радиальном направлении

- ❑ Угловая скорость задаётся в РАДИАНАХ в единицах времени

## ЗАКРЕПЛЕНИЕ УЗЛОВ В ЛОКАЛЬНЫХ СИСТЕМАХ КООРДИНАТ

- ❑ Оператор SPC3 используется для закрепления узлов в локальной системе координат (которая, в свою очередь, может быть закреплена в другой локальной системе координат)
- ❑ Операторы SPC3 должны быть инициированы соответствующим оператором Case Control
- ❑ Пример:

“Первичная” система координат, в которой закрепляются узлы

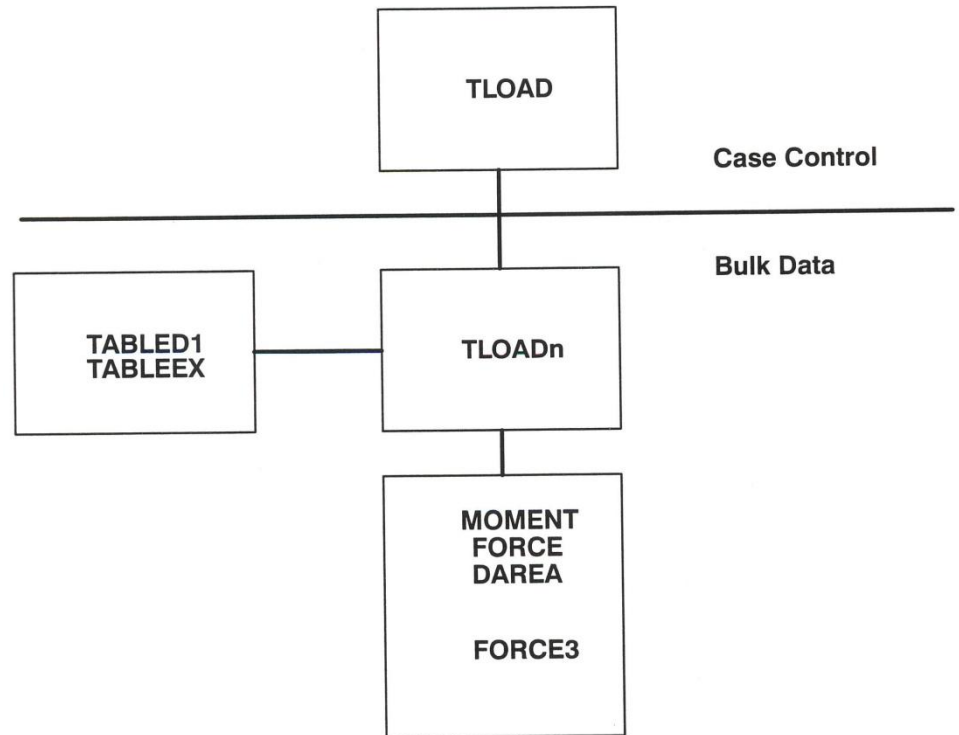
**Закрепляемые компоненты перемещения**

1	2	3	4	5	6	7	8	9	10
SPC3	SID	CID1	C1	CID2	C2				+CONT1
SPC3	1	5	12						+CONT1
+CONT1	G1	G2	THRU	G3	BY	G4	-etc.-		
+CONT1	5	6	THRU	18	BY	3			

“Вторичная” система координат (в которой может закрепляться “первичная” система координат) и закрепляемые компоненты перемещения “первичной” системы координат во “вторичной”

## ВЫНУЖДЕННОЕ ПЕРЕМЕЩЕНИЕ

- ❑ Задаётся *кинематическое* перемещение узлов
- ❑ Задание вынужденного перемещения инициируется при TYPE=2 в операторе TLOADn
- ❑ Операторы TLOADn раздела Bulk Data должны быть инициированы оператором Case Control
- ❑ Вынужденное перемещение может быть задано в локальной системе координат



## ВЫНУЖДЕННОЕ ПЕРЕМЕЩЕНИЕ УЗЛОВ

- ❑ Поле 5 (TYPE=2) в операторах TLOAD1 и TLOAD2 определяет, что задаётся вынужденное перемещение  
TLOAD1, 100, 110, , 2, 120  
TLOAD2, 100, 110, , 2,0., 10.E-3, 1000., 90.,+  
+, 0., 2.
- ❑ Оператор DAREA задаёт поступательную или угловую скорость по *отдельным* составляющим
- ❑ Операторы FORCE и MOMENT задают компоненты поступательной или угловой скорости по всем составляющим
- ❑ Вынужденная скорость может варьироваться во времени (задаётся оператором TABLED1)

```
TLOAD = 100
BEGIN BULK
...
TLOAD1, 100, 110, , 2, 120
TABLED1, 120, , , , , , +
+, 0., 0., 1., 1., ENDT
FORCE, 110, 27, , -6., , 1., 0.
```

## ВЫНУЖДЕННОЕ ПЕРЕМЕЩЕНИЕ

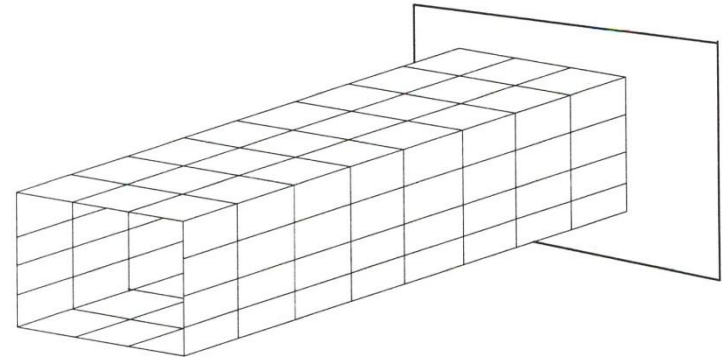
- Поле CORDXXX в операторе FORCE определяет систему координат, в которой задаётся вынужденное перемещение

FORCE, 110, 27, 2, -6., , 1.



## ЖЁСТКИЕ СТЕНКИ

- ❑ С помощью оператора WALL моделируется жёсткая плоскость, сквозь которую указанные “slave” узлы “проникать” не могут
- ❑ “Стенка” определяется заданием точки в пространстве и вектором, перпендикулярным к задаваемой плоскости
- ❑ Два типа контакта со “стенкой”
  - PENALTY – допускается внедрение (контактная сила нарастает с увеличением внедрения), трение может быть ненулевым
  - KINEMATIC – узлы “просто” “возвращаются” в плоскость жёсткой стенки, трение отсутствует



```
WALL,101,0.,0.,0.,0.,0.,1.,102,+  
+,PENALTY,0.2  
SET1,102,1,THRU,1999
```

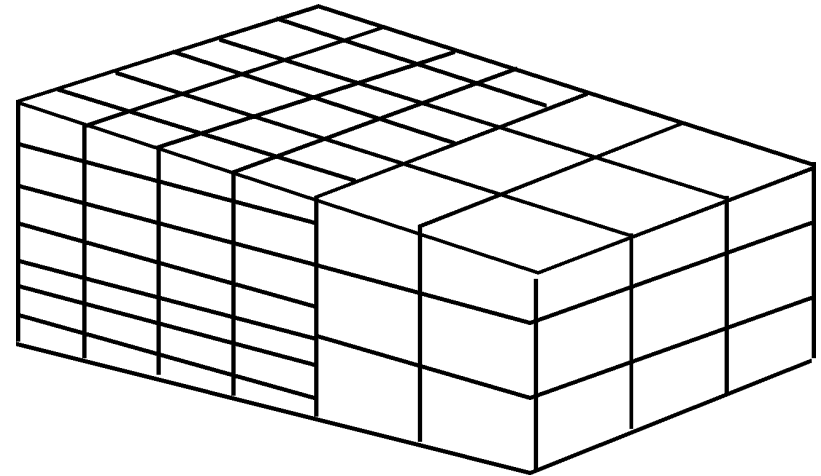
## ЭЛЕМЕНТЫ - СВЯЗИ

- ❑ С помощью элементов - связей связываются две *разные* (с разными размерами элементов) конечно-элементные сетки
- ❑ Возможно совмещение сеток из балочных, оболочечных и объёмных элементов без совмещения положения узлов – “заполнение” зазоров между несовпадающими сетками
- ❑ Не рекомендуется применять в зонах, где ожидаются “пики” напряжений или “разрушение” модели
- ❑ Три типа связи:
  - Поверхность – поверхность
  - Узлы – поверхность
  - Ребро оболочки – поверхность оболочки

## СВЯЗЬ ПОВЕРХНОСТЬ - ПОВЕРХНОСТЬ

□ **Две поверхности постоянно связаны между собой**

- **Master-поверхность: всегда связана с “грубой” сеткой**
- **Slave-поверхность: всегда связана с “подробной” сеткой**



□ **Взаимосвязь сил и скоростей:**

- **Силы: *slave*-узлы  $\Rightarrow$  *master*-узлы**
- **Скорости: *master*-узлы  $\Rightarrow$  *slave*-узлы**

□ **Пример: два объёмных тела связаны между собой посредством общих поверхностей 7 и 8**

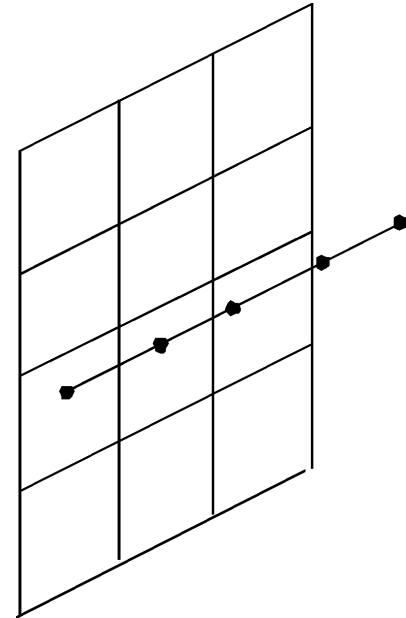
**RCNN, 1, SURF, SURF, 7, 8**

## СВЯЗЬ УЗЛЫ - ПОВЕРХНОСТЬ

- ❑ **Связь *отдельных* узлов с поверхностью (в операторе RCONN параметр OPTION=NORMAL)**
- ❑ **Узлы определяют slave-поверхность, master-поверхность определяется как набор сегментов**
- ❑ **Связываются только поступательные степени свободы**
- ❑ **Пример: узлы с 1-го по 10-ый (принадлежащих балочным элементам) связаны с поверхностью 7**

RCONN, 1, GRID, SURF, 3, 7, NORMAL

SET1, 3, 1, THRU, 10

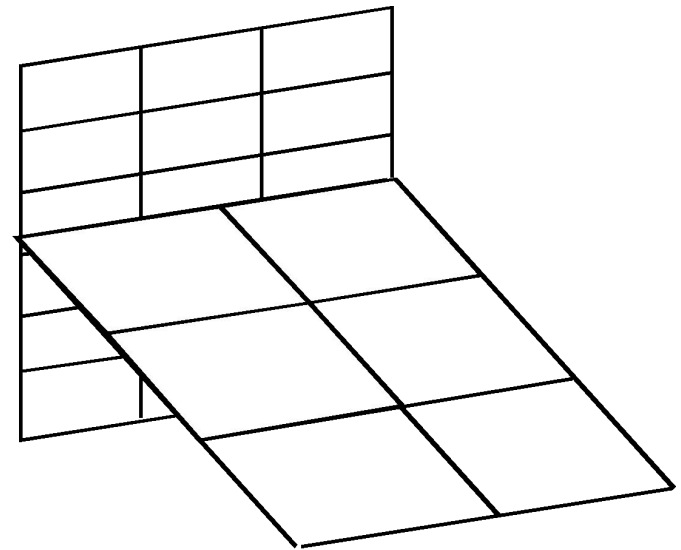


## СВЯЗЬ РЕБРО ОБЛОЧКИ - ПОВЕРХНОСТЬ

- ❑ Связь балок или ребер оболочек с поверхностью (в операторе RCONN параметр OPTION=SHELL)
- ❑ Узлы определяют slave-поверхность, master-поверхность определяется как набор сегментов
- ❑ Связываются только поступательные степени свободы
- ❑ Пример: узлы с 1-го по 10-ый (принадлежащих оболочечным элементам) связаны с поверхностью 7

RCONN, 1, GRID, SURF, 3, 7, SHELL

SET1, 3, 1, THRU, 10



## ЭЛЕМЕНТЫ - ЖЁСТКИЕ ТЕЛА

- ❑ **Оператор RBE2 задаёт набор узлов, определяющих жёсткое тело**
- ❑ **С помощью этого оператора можно сформировать набор узлов, перемещение которых по указанным направлениям, будут одинаковы**
  - **Может применяться для моделирования неразрушаемых точек сварки**
  - **Пример: узлы с 1-го по 28-ой будут иметь перемещения в направлениях x и z, равные перемещениям по x и z узла 55**  
**RBE2, 12, 55, 13, 1, THRU, 28**
- ❑ **Элемент RBE2 может использоваться наряду с элементами-связями**
  - **Использование вместо перечисления связываемых степеней свободы параметра FULLRIG приводит к тому, что перечисленные узлы “ведут себя” аналогично одному жёсткому телу**
  - **Пример: узлы с 1-го по 28-ой будут “вести себя” как жёсткое тело с именем FR12**  
  
**RBE2, 12, 55, FULLRIG, 1, THRU, 28**

## КИНЕМАТИЧЕСКАЯ СВЯЗЬ

- ❑ Элемент KJOIN может использоваться для связи оболочки с объёмным элементом (достигается связь по всем степеням свободы)
  - Элемент JOIN может использоваться в случае, если необходимо связать только поступательные степени свободы
- ❑ Связывание оболочки и объёмного элемента осуществляется наложением кинематических связей на узлы оболочки

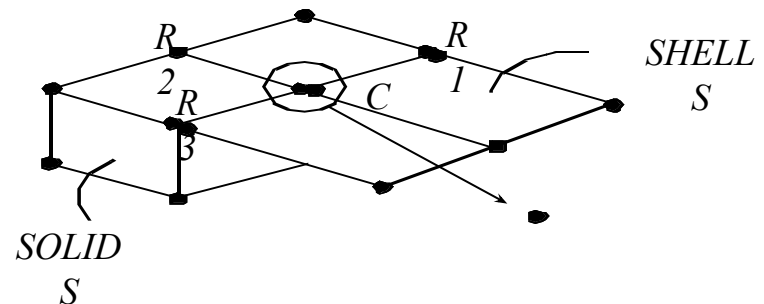
- ❑ Пример: элемент KJOIN связывает

а) узлы 30, 32, 40, 42, 50 и 52;

б) узлы, отстоящие друг от друга на расстояние не более  $1 \cdot 10^{-5}$

KJOIN,2,333,1.0E-5,,0.5

SET1,333,30,32,40,42,50,52



## РАЗРУШАЮЩАЯСЯ СВЯЗЬ

- ❑ Элемент VJOIN может применяться для моделирования связи между узлами балочных или оболочечных элементов с возможностью разрушения
- ❑ Разрушение связи происходит при наступлении соответствующих условий
- ❑ Модели разрушения:
  - По значению силы или момента
  - По значению отдельных компонентов силы или момента
  - “Точка сварки”
  - Пользователя (посредством пользовательской подпрограммы)
- ❑ Узлы, связываемые элементом VJOIN, могут отстоять на некоторое расстояние (моделирование точки сварки)
- ❑ Пример: соединяются элементы, отстоящие не более чем на  $1 \cdot 10^{-4}$ , разрушение связи при достижении силы  $1 \cdot 10^6$   
 VJOIN, 1, 333, 1.0E-4, FOMO, 1.0E6  
 SET1, 333, 31, THRU, 2000

