

**НСТ®**

**Управление фрезерным станком и центром обработки**

**Сборник задач**

---

Производитель и разработчик: **NCT Ipari Elektronikai kft.**

H1148 Budapest Fogarasi út 7

✉ Адрес для переписки: H1631 Bp. pf.: 26

☎ Телефон: (+36 1) 467 63 00

☎ Телефакс: (+36 1) 363 6605

Электронная почта: [nct@nct.hu](mailto:nct@nct.hu)

Домашняя страница: [www.nct.hu](http://www.nct.hu)

---

© Copyright NCT 19 Июль 2006 г.

По содержанию настоящего описания все издательские права сохраняются за собой. Для допечатки даже сокращённого издания требуется наше разрешение.

Описание составлено с максимальной осмотрительностью и тщательно проверено, однако за возможные ошибки или ошибочные данные *и за истекающие из этого ущерба ответственность на себя не берём.*

## Содержание:

1 Программирование интерполяции окружности с наводкой	6
2 Программирование интерполяции окружности со слежением за контуром	7
3 Программирование интерполяции окружности с наводкой по касательной	8
4 Программирование интерполяции окружности со слежением за контуром	9
5 Интерполяция окружности по внутреннему контуру	10
6 Внутренняя интерполяция окружности со слежением за контуром	11
7 Программирование внутренней интерполяцией окружности с наводкой по касательной	12
8 Программирование прямого контура	13
9 Программирование прямого контура со слежением за контуром	14
10 Программирование нулевых окружностей при слежении за контуром	15
11 Слежение за контуром со скруглениями	16
12 Четыреугольник с внутренним слежением за контуром, с наводкой по касательной	17
13 Программирование произвольного контура слежением за контуром	18
14 Описание произвольного контура с внутренним слежением за контуром	19
15 Произвольный внутренний контур (Подпрограмма)	20
16 Техника подпрограммы со смещением нулевой точки	21
17 Применение масштабирования с вызовом подпрограммы	22
18 Программирование масштабирования вокруг произвольной точки	23
19 Программирование ряда отверстий	24
20 Программирование рядов отверстий	25
21 Программирование окружности центров	26
22 Программирование фрагмента окружности центров	27
23 Программирование окружности центров в произвольную позицию	28
24 Программирование ряда отверстий, как подпрограммы	28
25 Программирование сетки отверстий с помощью подпрограммы	29
26 Программирование сетки отверстий с помощью цикла	30
27 Программирование сетки отверстий с помощью двух циклов, уложенных друг в друга	31
28 Автоматический геометрический расчёт между дугами (Внутренний контур)	32
29 Автоматический геометрический расчёт между дугами (Внешний контур)	33
30 Автоматический геометрический расчёт между дугами (Внутренний контур)	34
31 Подпрограмма программирования нарезания зубьев	35
32 Главная программа программирования нарезания зубьев	36
33 Программирование нарезание зубьев с одной программой	37
34 Программирование окружности центров с отражением, в подпрограмме	38
35 Подпрограмма программирования окружности центров с отражением:	39
36 Макропрограммирование - синусоидальная кривая	40
37 Макропрограммирование - круговое фрезерование, интерполяция цилиндра	41
38 Макропрограммирование - полушар	42
39 Макропрограммирование - порождение полушара	43

19 Июль 2006 г.

## 1 Программирование интерполяции окружности с наводкой

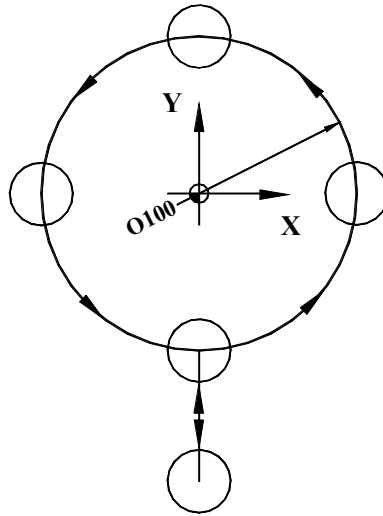


Рисунок 1

```
%O7011 (1.1)
N100 T1
N110 G54 G0 X0 Y-100
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 X0 Y-50 F50
N170 G3 J50
N180 G1 X0 Y-100
N190 G0 Z100
N200 M30
%
```

Первое предложение содержит загрузку желаемого инструмента. Во втором предложении имеется ориентировка в плоскости X-Y, в соответствующей системе координат. Третье предложение отвечает за взятие в учёт коррекции инструмента по длине. Остальные программы сборника задач имеет подобное начало программы, поэтому в дальнейшем на это не распространяемся. В начале программы виден ещё знак процента, а также за буквой O имеются четыре цифры, чем идентифицируется программа. Программа закачивается также с процентным знаком. Эти характеры нужны вытавить только при программировании на внешних устройствах, в других случаях они создаются управлением автоматически. Описание траектории инструмента начинается с четвёртого предложения. Запрограммирование полной окружности выполняется с помощью адреса J, что представляет собой задачу центра окружности инкрементально по сравнению начальной точки окружности. Завершение программы выполняется с кодом M30.

## 2 Программирование интерполяции окружности со слежением за контуром

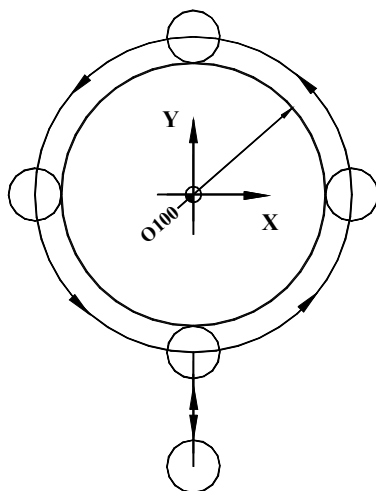


Рисунок 2

```

%O7012(1.2)
N100 T1
N110 G54 G0 X0 Y-100
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G42 X0 F50 D1
N170 G3 J-50
N180 G1 G40 Y-100
N190 G0 Z100
N200 M30
%
```

Разница по сравнению к предыдущему примеру имеется в слежении за контуром. В предложении N160 включить слежение. При этом конечная точка предложения будет изменено управлением так, чтобы следующее предложение можно было начинать со смещением диаметра указанного инструмента. Остановка работает подобным образом, здесь коррекция будет ноль в конечной точке предложения (N180).

### 3 Программирование интерполяции окружности с наводкой по касательной

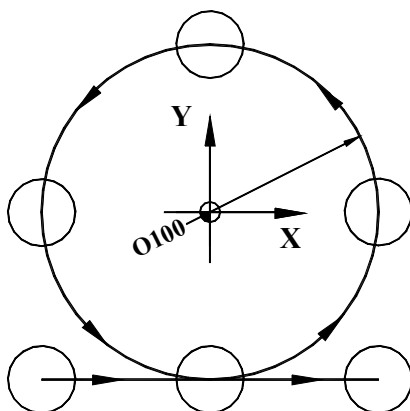


Рисунок 3

```

%O7013(1.3)
N100 T1
N110 G54 G0 X-50 Y-50
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 X0 F50
N170 G3 J50
N180 G1 X50
N190 G0 Z100
N200 M30
%
```

Разница по сравнению к предыдущему примеру имеется в наводке. Поскольку желаемая позиция достигается осями ускорением-замедлением, поэтому в предыдущем примере инструмент оставляет след в начальной точке окружности. Для избежания этого следует в каждом случае приблизиться к контуру по касательной.



## 4 Программирование интерполяции окружности со слежением за контуром

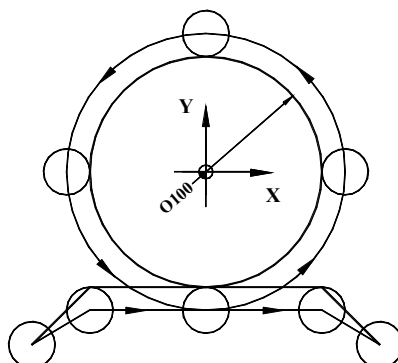


Рисунок 4

```

%O7014 (1.4)
N100 T1
N110 G54 G0 X-70 Y-70
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G42 X-50 Y-50 D1 F50
N170 G1 X0
N180 G3 J50
N190 G1 X50
N200 G1 G40 X70 Y-70
N210 G0 Z100
N220 M30
%
```

Разница по сравнению к предыдущему примеру имеется в начальной позиции и при наводке к контуру далее при отводе от него. Координаты наводки и отвода следует запрограммировать так, чтобы смещение без коррекции было больше радиуса инструмента.

## 5 Интерполяция окружности по внутреннему контуру

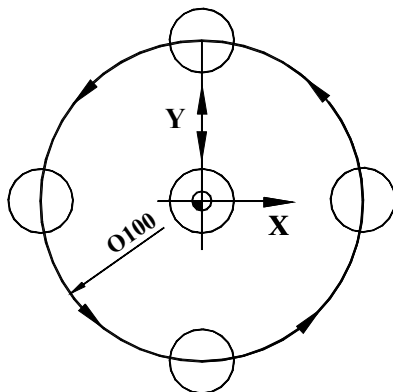


Рисунок 5

```
%O7021(2.1)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z0 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 X0 Y50 F50
N170 G3 J-50
N180 G1 X0 Y0
N190 G0 Z100
N200 M30
%
```

В этом случае проблема создана наводкой под прямым углом также, как в примере 1.

## 6 Внутренняя интерполяция окружности со слежением за контуром

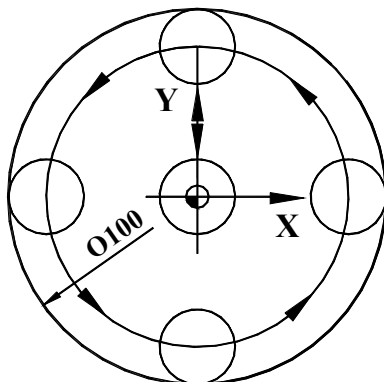


Рисунок 6

```
%O7022(2.2)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z0 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G41 X0 Y50 F50 D1
N170 G3 J-50
N180 G1 G40 X0 Y0
N190 G0 Z100
N200 M30
%
```

По сравнению к слежению по внешнему контуру изменилось лишь G41-G42. Смену G41-G42 можно было бы заменить и на смену G2-G3. Выход в начальной точке окружности этим решением также не устранено.

## 7 Программирование внутренней интерполяцией окружности с наводкой по касательной

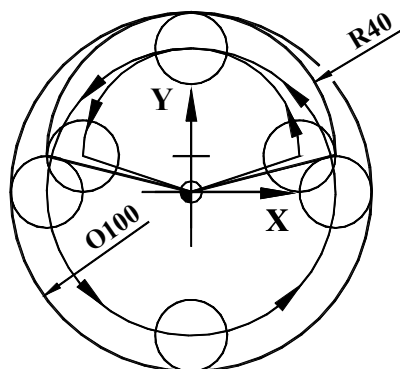


Рисунок 7

```

%O7023(2.3)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G41 X40 Y10 D1 F50
N170 G3 X0 Y50 R40
N180 G3 J-50
N190 G3 X-40 Y10 R40
N200 G1 G40 X0 Y0
N210 G0 Z100
N220 M30
%
```

Наводка на внутреннюю дугу происходит по касательной также, как при внешнего контура во избежание поверхностных дефектов, с применением интерполяции окружности. Окружность касательной - это окружность с произвольным радиусом, большим радиуса инструмента, но меньше радиуса окружности. Её центр смещён по направлению Y на разницу радиуса контура и радиуса окружности касательной, по сравнению к первоначальному центру окружности.

## 8 Программирование прямого контура

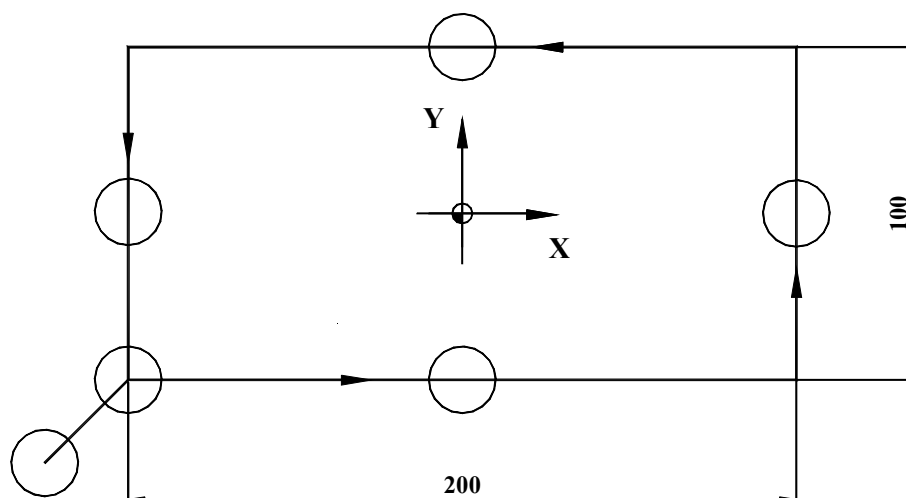


Рисунок 8

```

%O7031(3.1)
N100 T1
N110 G54 G0 X-120 Y-70
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 X-100 Y-50 F50
N170 G1 X100
N180 G1 Y50
N190 G1 X-100
N200 G1 Y-50
N210 G1 X-120 Y-70
N220 G0 Z100
N230 M30
%
```

Самым простым примером на мире является обфрезерование четырёхугольника. Коды G1 повторяются лишь ради обозримости, повторяющиеся коды с точки зрения управления являются лишними.

## 9 Программирование прямого контура со слежением за контуром

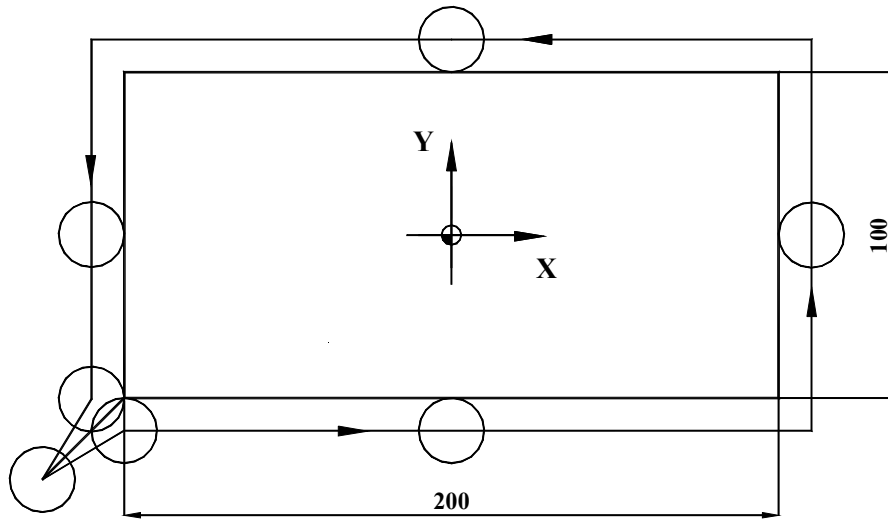


Рисунок 9

```

%O7032(3.2)
N100 T1
N110 G54 G0 X-120 Y-70
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G42 X-100 Y-50 F50 D1
N170 G1 X100
N180 G1 Y50
N190 G1 X-100
N200 G1 Y-50
N210 G1 G40 X-120 Y-70
N220 G0 Z100
N230 M30
%
```

Меру наводки следует определить такой же, как в примере 1.4, чтобы она была больше радиуса инструмента. В случае, изображенного на рисунке, на углу начальной точки может оставаться заусеница, поэтому целесообразно запрограммировать несколько лишних миллиметров при наводке до угла, а при остановке после угла.

## 10 Программирование нулевых окружностей при слежении за контуром

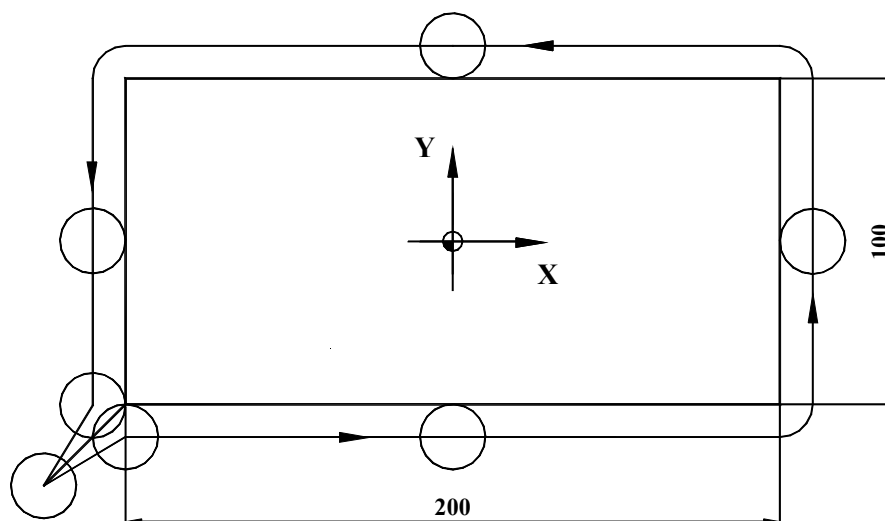


Рисунок 10

```

%O7033(3.3)
N100 T1
N110 G54 G0 X-120 Y-70
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G42 X-100 Y-50 F50 D1
N170 G1 X100 ,R0
N180 G1 Y50 ,R0
N190 G1 X-100 ,R0
N200 G1 Y-50
N210 G1 G40 X-120 Y-70
N220 G0 Z100
N230 M30
%
```

,R0 понимается управлением как дуга с нулевым радиусом, и таким образом при слежении за контуром эта часть участка выполняется как дуга. Радиус траектории инструмента будет равен радиусу инструмента, в то же время по контуру остаётся острый угол. При большем припуске на черновую обработку этот вариант способствует более равномерной нагрузке инструмента.

## 11 Слежение за контуром со скруглениями

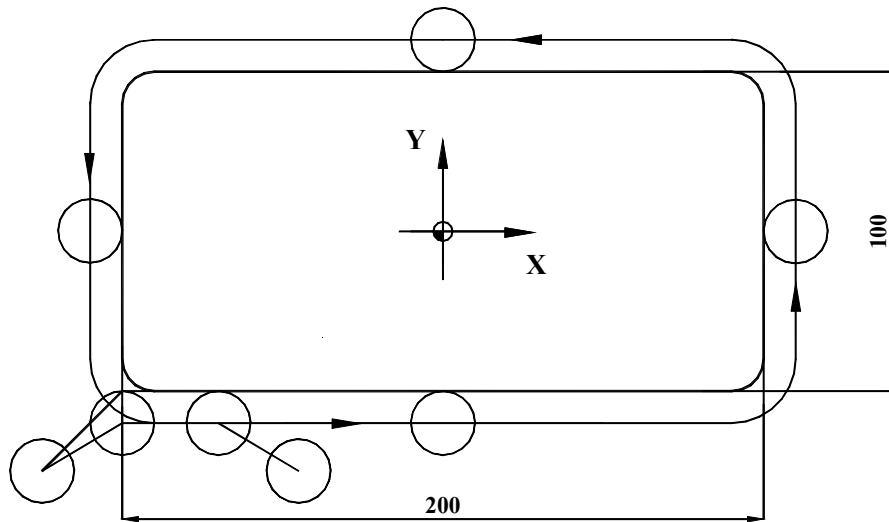


Рисунок 11

```

%O7034(3.4)
N100 T1
N110 G54 G0 X-120 Y-70
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G42 X-100 Y-50 F50 D1
N170 G1 X100 ,R10
N180 G1 Y50 ,R10
N190 G1 X-100 ,R10
N200 G1 Y-50 ,R10
N210 G1 X-70
N220 G1 G40 X-50 Y-70
N230 G0 Z100
N240 M30
%
```

Для скругления четвёртого угла необходимо вернуть инструмент на первую кромку, хотя бы на её кусочек. Это расстояние должно быть больше суммы радиусов скругления и инструмента. При скруглениях радиус траектории инструмента равен сумме радиусов скругления и инструмента.



## 12 Четыреугольник с внутренним слежением за контуром, с наводкой по касательной

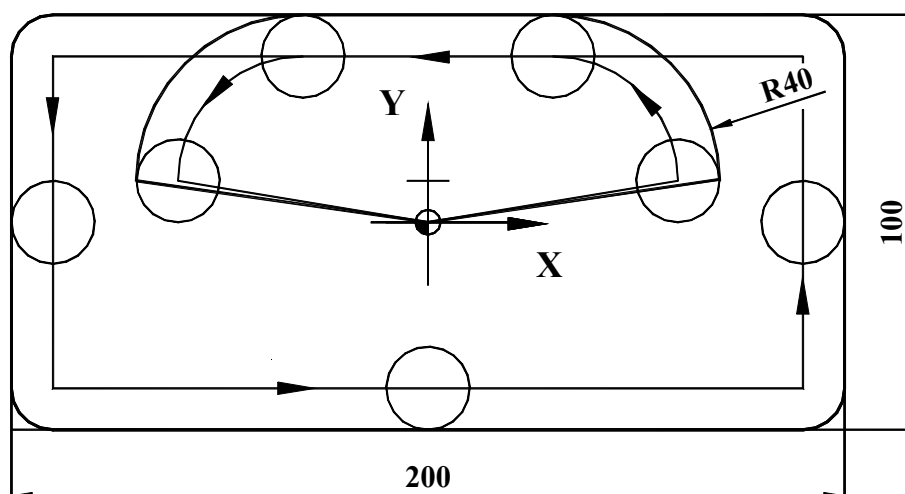


Рисунок 12

```

%O7041(4.1)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G41 X80 Y10 F50 D1
N170 G3 X40 Y50 R40
N180 G1 X-100
N190 G1 Y-50
N200 G1 X100
N210 G1 Y50
N220 G1 X-40
N230 G3 X-80 Y10 R40
N240 G1 G40 X0 Y0
N250 G0 Z100
N260 M30
%
```

Наводка совершается здесь также по дуге, как в примере 3.2, радиус которого больше радиуса инструмента, но меньше половины размера полости в направлении Y.

### 13 Программирование произвольного контура слежением за контуром

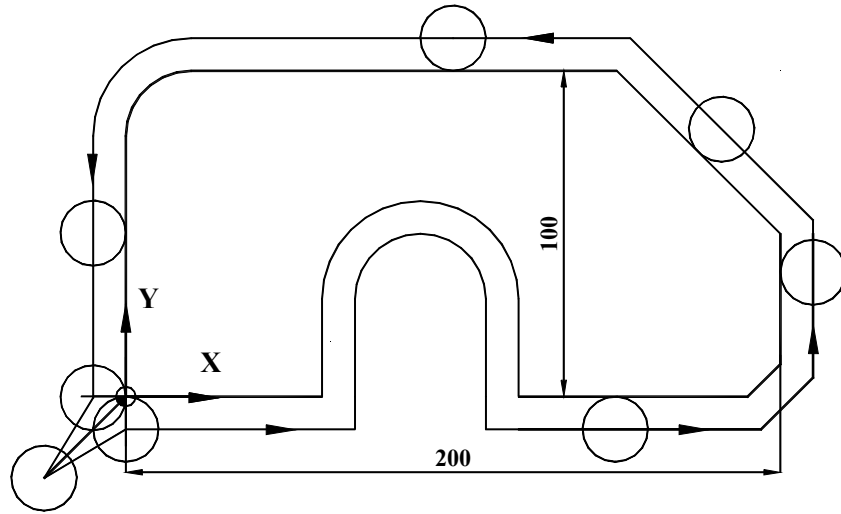


Рисунок 13

```

%O7051(5.1)
N100 T1
N110 G54 G0 X-40 Y-40
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G42 X0 Y0 F50 D1
N170 G1 X50
N180 G1 Y30
N190 G2 X100 R25
N200 G1 Y0
N210 G1 X200 ,C10
N220 G1 Y50
N230 G1 X150 Y100
N240 G1 X0 ,R20
N250 G1 Y0
N260 G1 G40 X-30 Y-30
N270 G0 Z100
N280 M30
%
```

В этом примере приведен произвольный контур. В строке 210 имеется фаска, а в строке 240 скругление. Фаска в строке 210 имеет обозначение ,C, где по прямому, описанным предыдущей и последующей строками, промерится управлением заданное расстояние и соединяются эти точки, как получаемый контур.

## 14 Описание произвольного контура с внутренним слежением за контуром

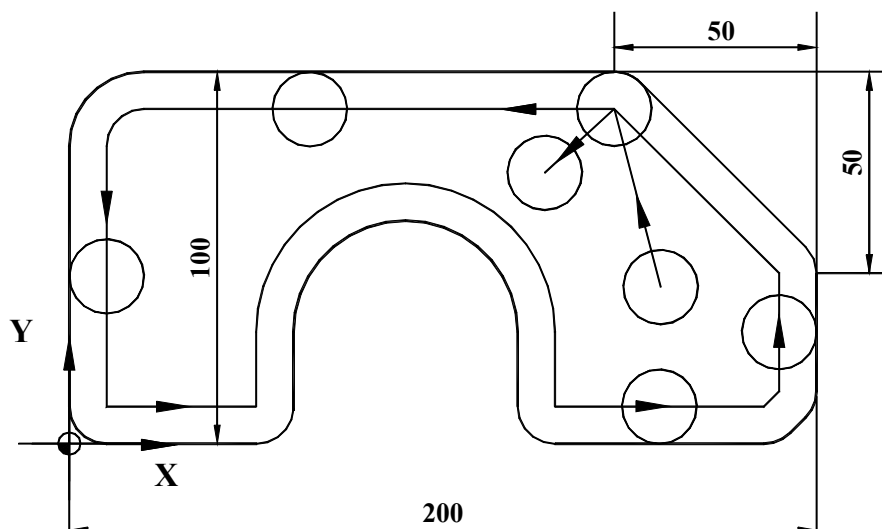


Рисунок 14

```

%O7052(5.2)
N100 T1
N110 G54 G0 X160 Y50
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G41 X150 Y100 I-1 J1 F50 D1
N170 G1 X0 ,R20
N180 G1 Y0
N190 G1 X50
N200 G1 Y30
N210 G2 X100 R25
N220 G1 Y0
N230 G1 X200 ,C10
N240 G1 Y50
N250 G1 X150 Y100
N260 G1 G40 XI-20 YI-20 I-1
N270 G0 Z100
N280 M30
%
```

Наводка на внутренний контур производится по адресу G41 I\_ J\_ в предложении N160. По адресу I и J можно задавать нуль-вектор предыдущего предложения, в этом случае управление наводится на контур так, что для расчёта координат принимает во внимание прямую, определённую заданным вектором направления, а не предложением наводки. Этот способ работает подобным образом и при остановке. По адресу I и J можно задавать только угловой коэффициент, значит, абсолютная величина безразлична, важным считается только знак и частное.

## 15 Произвольный внутренний контур (Подпрограмма)

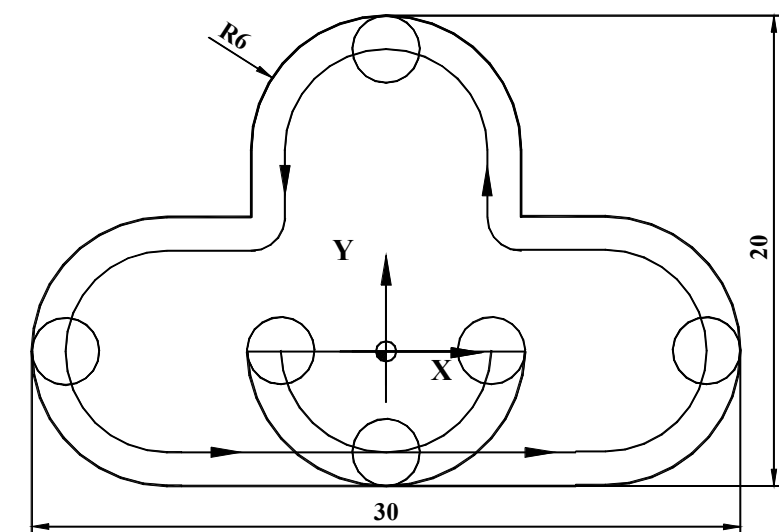


Рисунок 15

```

%O7053 (5.3)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G41 X-6 F50 D1
N170 G3 X0 Y-6 R6
N180 G1 X9
N190 G3 Y6 R6
N200 G1 X6
N210 G1 Y14
N220 G3 X-6 R6
N230 G1 Y6
N240 G1 X-9
N250 G3 Y-6 R6
N260 G1 X0
N270 G3 X6 Y0 R6
N280 G1 G40 X0 Y0
N290 G0 Z100
N300 M99
%
```

Программа написана как подпрограмма, ссылкой является на это, что вместо M30 в конце программы имеется M99. Если запустить как главную программу, совершает бесконечные циклы. Этой программой пользуемся позже в качестве подпрограммы.

## 16 Техника подпрограммы со смещением нулевой точки

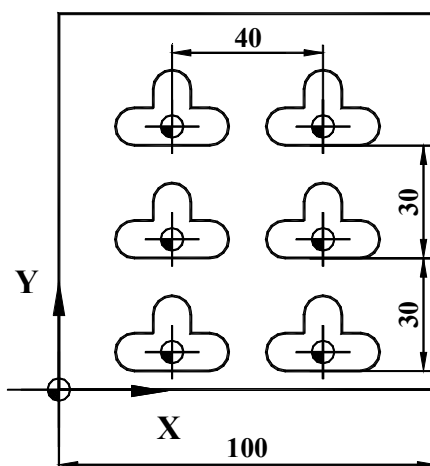


Рисунок 16

```

%O7061(6.1)
N100 T1
N110 G54
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G52 X30 Y10
N150 M98 P7053
N160 G52 X70 Y10
N170 M98 P7053
N180 G52 X30 Y40
N190 M98 P7053
N200 G52 X70 Y40
N210 M98 P7053
N220 G52 X30 Y70
N230 M98 P7053
N240 G52 X70 Y70
N250 M98 P7053
N260 G52 X0 Y0
N270 G0 Z100
N280 M30
%
```

Подпрограмма составлена с нулевой точкой, описанной с абсолютной задачей размера, изображенной на рисунке. Подпрограмма подробно описана в примере 5.3. В главной программе имеется одно локальное смещение системы координат и один вызов подпрограммы попарно. Подпрограмма завершается командой M99.

## 17 Применение масштабирования с вызовом подпрограммы

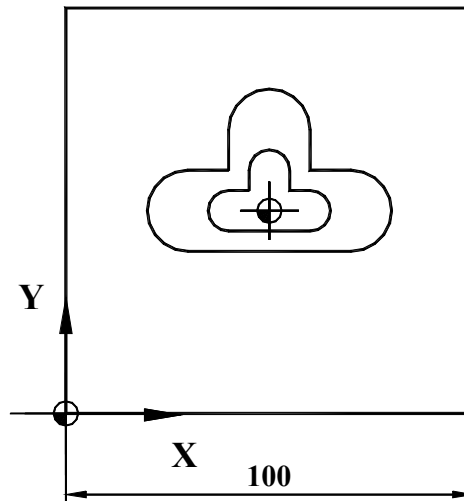


Рисунок 17

```
%O7062 (6.2)
N100 T1
N110 G54
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G52 X50 Y50
N150 G51 X0 Y0 P2
N160 M98 P7053
N170 G50 X0 Y0
N180 G52 X0 Y0
N190 G0 Z100
N200 M30
%
```

G52 отвечает за загрузку локальной системы координат, а G51 - за включение масштабирования. Последующие перемещения увеличиваются управлением соразмерно значению, заданной по адресу P, вокруг центра, заданного координатами предложения G51. Описание контура выполняется и здесь в подпрограмме.

## 18 Программирование масштабирования вокруг произвольной точки

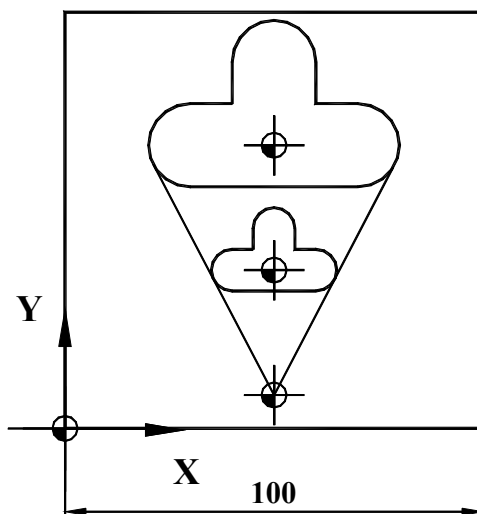


Рисунок 18

```

%O7063(6.3)
N100 T1
N110 G54
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G52 X50 Y50
N150 G51 X0 Y-40 P2
N160 M98 P7053
N170 G50 X0 Y0
N180 G52 X0 Y0
N190 G0 Z100
N200 M30
%
```

Координаты, заданные в предложении G51, необходимо задавать по отношению смещения нулевой точки по G52, поэтому имеется по адресу Y значение -40. В этом случае нулевая точка тоже смещается по отношению точки, заданной в предложении G51, как точка центра увеличения.

## 19 Программирование ряда отверстий

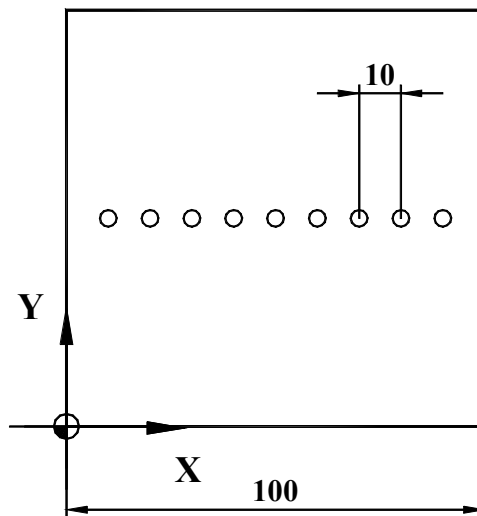


Рисунок 19

```

%O7071 (7.1)
N100 T1
N110 G54 G0 X0 Y50
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G81 X110 Y50 R2 Z-10 L9
N150 G80
N160 G0 Z100
N170 M30
%
```

Программирование отверстий, расположенных друг от друга на 10 мм выполнено инкрементально, повторением. В этом случае надо следить за тем, что необходимо одно предложение для ориентации выставить **перед** первое отверстие, так как задача точек совершилась не абсолютными значениями. Цикл начинается отшагиванием инкрементного расстояния, затем выполняется сверление, и это повторяется по значению, заданному по адресу L. Сверлильный цикл завершается кодом G80.



## 20 Программирование рядов отверстий

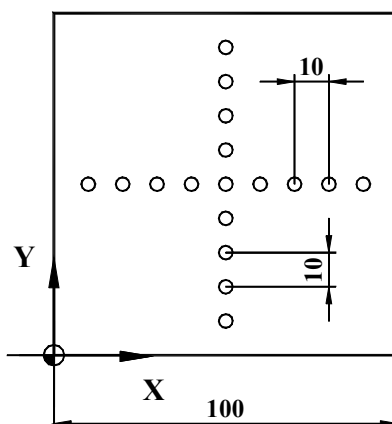


Рисунок 20

```

%O7072(7.2)
N100 T1
N110 G54 G0 X0 Y50
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G81 XI10 Y50 R2 Z-10 L9
N150 G80
N160 G0 X50 Y0
N170 G81 X50 YI10 R2 Z-10 L4
N180 G80
N190 G0 X50 Y50
N200 G81 X50 YI10 R2 Z-10 L4
N210 G80
N220 G0 Z100
N230 M30
%
```

Запрограммировав второй ряд отверстий в качестве нового сверлильного цикла, размеры можно задавать также инкрементно. Поскольку отверстие по середине вошло бы в оба цикла, поэтому второй ряд надо разделить на верхнюю и нижнюю часть.

## 21 Программирование окружности центров

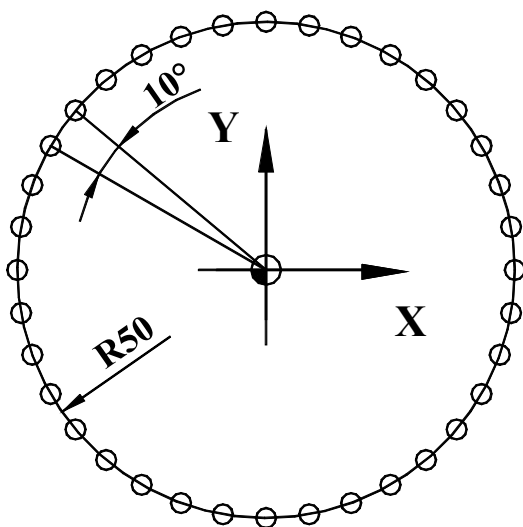


Рисунок 21

```
%O7073(7.3)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G16 G0 X50 Y-10
N150 G81 X50 YI10 R2 Z-10 L36
N160 G80 G15
N170 G0 Z100
N180 M30
%
```

Программирование окружности центров отличается от программирования ряда отверстий в том, что используется задача данных в полярных координатах. Ориентация выполняется тоже перед первое отверстие, но это не имеет значение в случае полной окружности.

## 22 Программирование фрагмента окружности центров

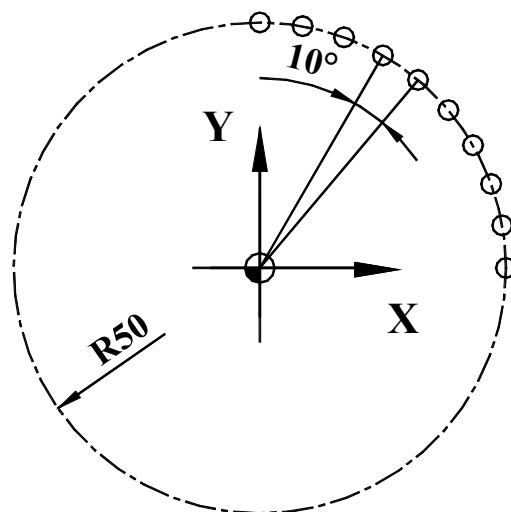


Рисунок 22

```
%O7074 (7.4)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G16 G0 X50 Y-10
N150 G81 X50 YI10 R2 Z-10 L10
N160 G80 G15
N170 G0 Z100
N180 M30
%
```

Программирование фрагмента окружности центров отличается от программирования окружности центров в том, что произведение числа повторения и разницы угла не достигает 360 градусов. Ориентация выполняется тоже перед первое отверстие.

## 23 Программирование окружности центров в произвольную позицию

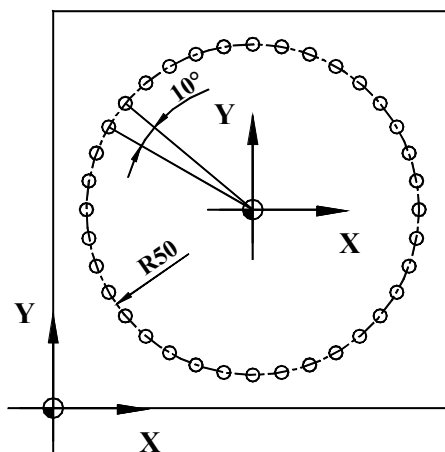


Рисунок 23

```

%O7075 (7.5)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G52 X60 Y60
N150 G16 G0 X50 Y-10
N160 G81 X50 YI10 R2 Z-10 L36
N170 G80 G15
N180 G52 X0 Y0
N190 G0 Z100
N200 M30
%
```

В этом случае нулевая точка заготовки расположена не в центре заготовки, а в одном из углов. При этом необходимо запрограммировать локальную систему координат (G52), затем программа в дальнейшем совпадает с предыдущим примером. В конце цикла необходимо выключить локальную систему координат, иначе остальные координаты, заданные абсолютным значением, вычисляются от этой нулевой точки.

## 24 Программирование ряда отверстий, как подпрограммы

```

%O7076 (7.6)
N100 G81 YI10 R2 Z-10 L9
N110 G80
N120 G0 XI10 Y0
N130 M99
%
```

Программирование ряда отверстий совершенно похоже к предыдущему примеру с той разницей, что X в конце ряда отверстий содержит и ориентировку. И так, вызывая с номером, заданным из главной программы, получим сеть отверстий.

## 25 Программирование сетки отверстий с помощью подпрограммы

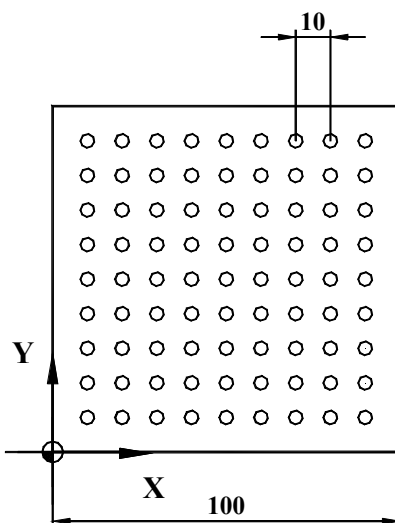


Рисунок 24

```

%O7077(7.7)
N100 T1
N110 G54 G0 X10 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 M98 P7076 L9
N150 G0 Z100
N160 M30
%
```

Эту задачу можно решить разными способами. Самым простым способом является горизонтальные ряды отверстий запрограммировать в качестве подпрограммы, и подпрограмму многократно вызывать. В подпрограмме имеется смещение нулевой точки, которое необходимо выключить в конце главной программы. Здесь в начале и конце программы видны процентный знак и номер программы, имя программы, с целью облегчения различать программы друг от друга.

## 26 Программирование сетки отверстий с помощью цикла

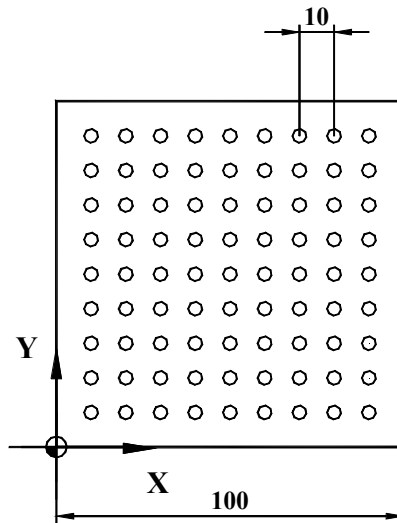


Рисунок 26

```

%O7078 (7.8)
N100 T1
N110 G54 G0 X10 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 #1=1
N150 WHILE[#1LE9] DO1
N160 G0 X10 Y0
N170 G81 YI10 R2 Z-10 L9
N180 G80
N190 G52 XI10 Y0
N200 #1=#1+1
N210 END1
N220 G52 X0 Y0
N230 G0 Z100
N240 M30
%
```

Другим способом является организация вертикальных рядов отверстий во внутренний цикл. В этом случае не требуется подпрограмма, но необходимо ввести так называемые макропеременные. Эти макропеременные определяются присвоением им значений, и можно с ними как угодно пользоваться. Цикл характеризуется командой WHILE, которую можно охарактеризовать словом ПОКА. Начало цикла обозначает DO1, конец цикла - END1, где цифры указывают на тесную связь.

## 27 Программирование сетки отверстий с помощью двух циклов, уложенных друг в друга

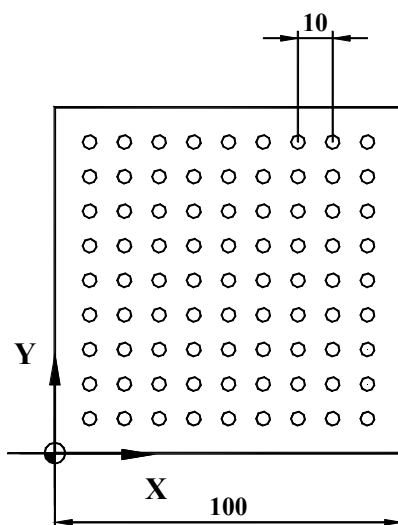


Рисунок 27

```

%O7079(7.9)
N100 T1
N110 G54 G0 X10 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 #1=10
N150 WHILE[#1LE90] DO1
N140 #2=10
N150 WHILE[#2LE90] DO2
N170 G81 X#1 Y#2 R2 Z-10
N200 #2=#2+10
N210 END2
N200 #1=#1+10
N210 END1
N220 G80
N230 G0 Z100
N240 M30
%
```

В третьем случае при двух циклов, уложенных друг в друга, макропеременные согласуются координатами отверстий X, а также Y. Таким образом начальное значение макропеременного надо установить по координатам первого, а конечное значение - по координатам последнего. Увеличение макропеременных осуществляется разницей между координатами.

## 28 Автоматический геометрический расчёт между дугами (Внутренний контур)

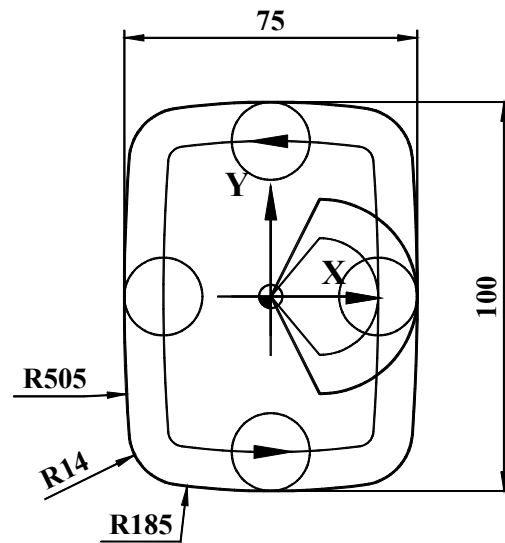


Рисунок 28

```
%O7081(8.1)
N100 G54 G90 G17 G0
N110 T1
N120 G43 Z50 H1
N130 S1000 M3
N140 G0 X0 Y0
N150 G0 Z2
N160 G1 Z-19
N170 G0 X0 Y0
N180 G41 G1 X17.5 Y-20 D1
N190 G3 X37.5 Y0 R20
N200 G3 XI-505 YI505 R505 ,R14
N210 G3 IO J-135 R185 Q-1 ,R14
N220 G3 I467.5 JO R505 Q-1 ,R14
N230 G3 IO J135 R185 Q-1 ,R14
N240 G3 X37.5 Y0 I-467.5 JO R505 Q-1
N250 G3 X17.5 Y20 R20
N260 G1 G40 X0 Y0
N270 Z50
N280 M30
%
```



## 29 Автоматический геометрический расчёт между дугами (Внешний контур)

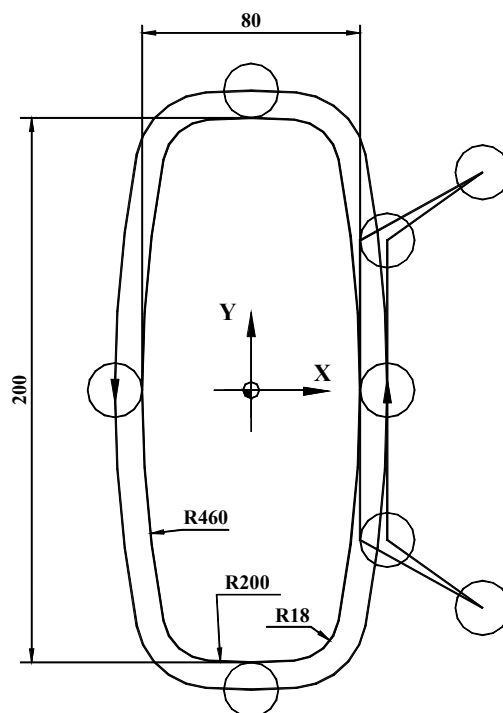


Рисунок 29

```

%O7082(8.2)
N100 G54 G90 G17 G0
N110 T1
N120 G43 Z50 H1
N130 S1000 M3
N140 G0 X0 Y0
N150 G0 Z2
N160 G1 Z-19
N170 G0 X90 Y-70
N180 G42 G1 X40 Y-50 D1
N190 G1 Y0
N200 G3 XI-460 YI460 R460 ,R18
N210 G3 IO J-100 R200 Q-1 ,R18
N220G3 I420 J0 R40 Q-1 ,R18
N230 G3 IO J100 R200 Q-1 ,R18
N240 G3 X40 Y0 I-420 J0 R460 Q-1
,R18
N260 G1 X40 Y50
N270 G1 G40 X90 Y70
N280 Z50
N290 M30
%
```

### 30 Автоматический геометрический расчёт между дугами (Внутренний контур)

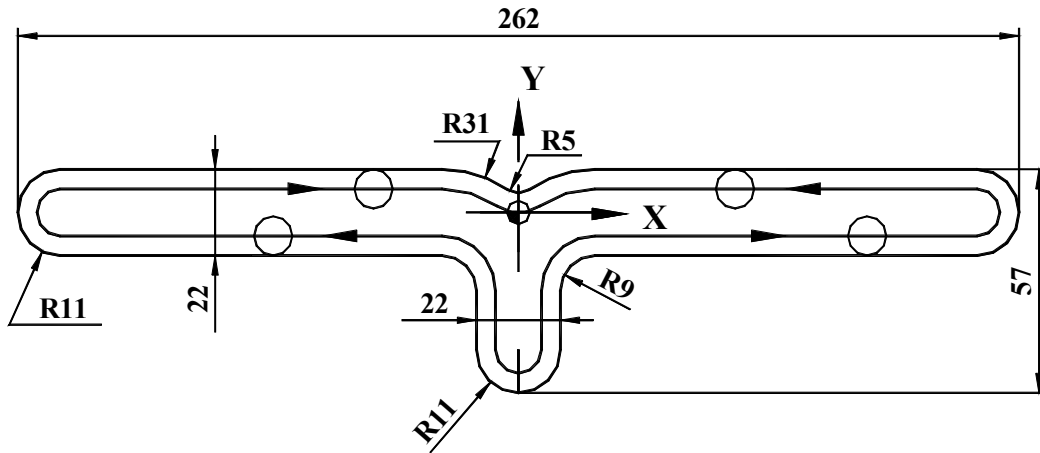


Рисунок 30

```

%O7083(8.3)
N100 G17 G0 G90 G94
N110 T1
N120 G0 X0 Y0
N130 G43 Z50 H1
N140 S1000 M3
N150 G0 Z-50
N160 G0 X0 Y-35
N170 G1 G41 Y-46 D1
N180 G3 X11 Y-35 R11
N190 G1 Y-11 ,R9
N200 G1 X120
N210 G3 X120 Y11 R11
N220 G1 X20
N230 G3 X-11 Y-20 R31 ,R5
N240 G3 X-20 Y11 I-20 J-20 R31 Q1
N250 G1 X-120
N260 G3 Y-11 R11
N270 G1 X-11 ,R9
N280 G1 Y-35
N290 G3 X0 Y-46 R11
N300 G1 G40 Y-35
N310 G0 X0 Y0
N320 Z50
N330 M30
%
```

Если использовать хвостовую фрезу 22, радиусы 11 выполняются фрезой, и таким образом совпадает траектория фрезы двух направлений. В этом случае при наводке отказались от приближения по касательной.

## 31 Подпрограмма программирования нарезания зубьев

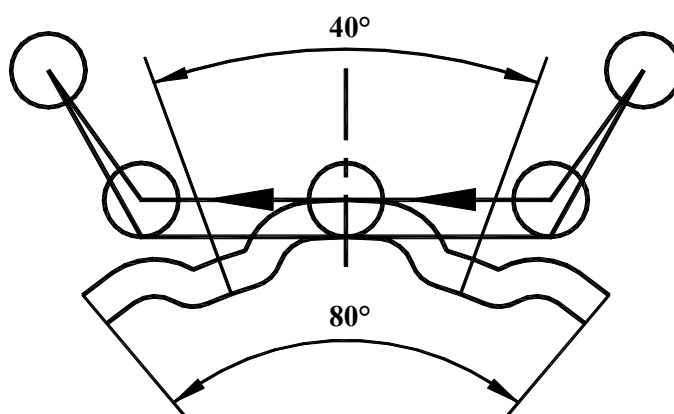


Рисунок 31

```

%O7091(9.1)
N100 G1 X0 Y50
N110 G3 X-4.587 Y49.789 R50
N120 G3 X-8.846 Y46.468 R5
N130 G2 X-12.206 Y43.313 R5
N140 G3 X-15.391 Y42.286 R45
N150 G3 X-18.490 Y41.026 R45
N160 G2 X-23.093 Y41.282 R5
N170 G3 X-28.490 Y41.090 R5
N180 G3 X-32.139 Y38.302 R50
N190 G68 X0 Y0 RI40
N190 M99
%
```

В этом примере, как в подпрограмме нужно описывать координаты одного зуба. К сожалению, как при нарезании вообще, профиль зуба нельзя описать простыми геометрическими расчётами, поэтому координаты профиля придётся рассчитывать с помощью какой-то системы CAD/(CAM). В строке G68 осуществляется инкрементное вращение системы координат.

### 32 Главная программа программирования нарезания зубьев

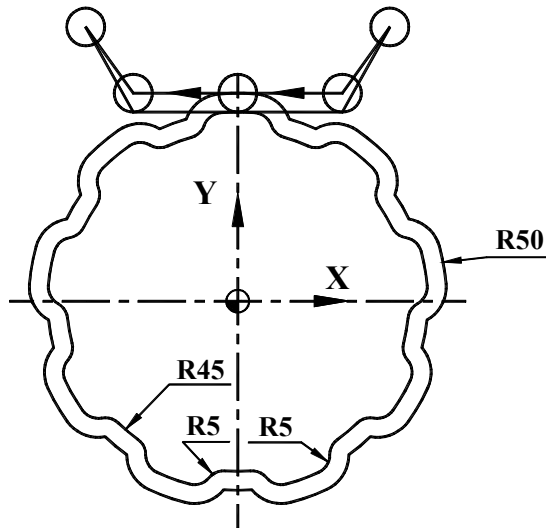


Рисунок 32

```
%O7092(9.2)
N100 T1
N110 G54 G0 X40 Y70
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 G42 X30 Y50 D1
N150 G1 X0 Y50
N180 M98 P7091 L9
N220 G69
N230 G1 X-30 Y50
N240 G0 G40 X-40 Y70
N250 G0 Z100
N260 M30
%
```

В этой программе вызывается подпрограмма столько раз, сколько число зубьев. В конце программы необходимо выключить вращение системы координат, а также отходить от контура.

## 33 Программирование нарезание зубьев с одной программой

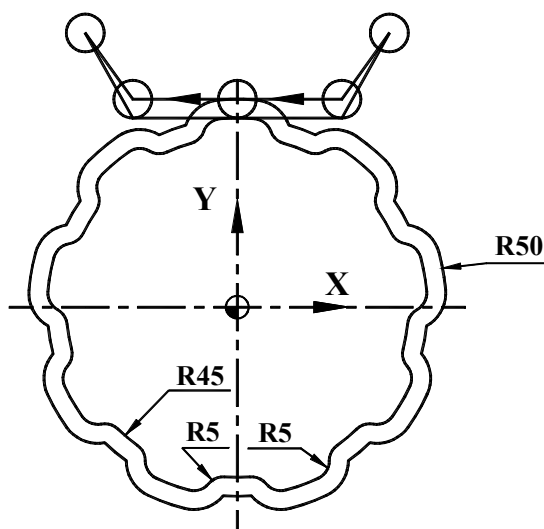


Рисунок 33

```

%O7093(9.3)
N100 T1
N110 G54 G0 X40 Y70
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 G42 X30 Y50 D1
N150 G1 X0 Y50
N160 #1=0
N170 WHILE[#1LT9] DO1
N180 G68 X0 Y0 R[#1*40]
N190 G1 X0 Y50
N200 G3 X-4.587 Y49.789 R50
N210 G3 X-8.846 Y46.468 R5
N220 G2 X-12.206 Y43.313 R5
N230 G3 X-15.391 Y42.286 R45
N240 G3 X-18.490 Y41.026 R45
N250 G2 X-23.093 Y41.282 R5
N260 G3 X-28.490 Y41.090 R5
N270 G3 X-32.139 Y38.302 R50
N280 #1=#1+1
N290 END1
N300 G69
N310 G1 X-30 Y50
N320 G40 X-40 Y70
N330 G0 Z100
N340 M30
%
```

С помощью макропеременных и применением внутреннего цикла профиль зуба можно уложить в программу, далее имеется возможность абсолютной задачи поворота. Готовый контур во всём совпадает с предыдущим примером.

### 34 Программирование окружности центров с отражением, в подпрограмме

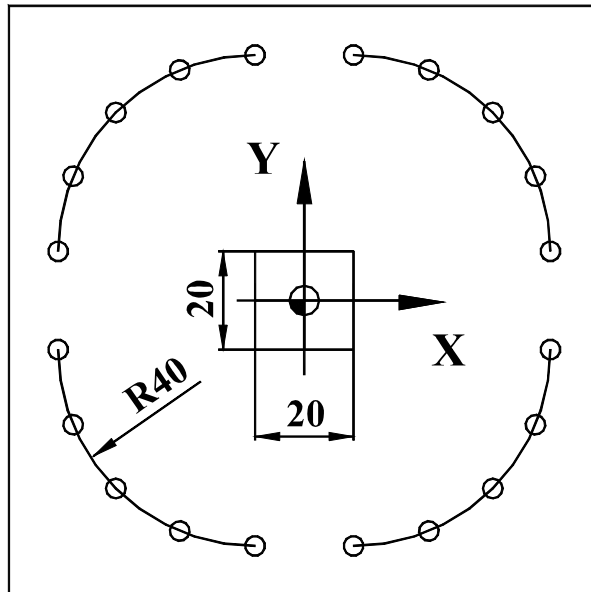


Рисунок 34

```

%O7101 (10.1)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G52 X10 Y10
N150 M98 P7102
N160 G51.1 Y0
N170 M98 P7102
N180 G51.1 X0
N190 M98 P7102
N200 G50.1 Y0
N210 M98 P7102
N220 G50.1 X0
N230 G52 X0 Y0
N240 G0 Z100
N250 M30
%
```

В этом случае нулевая точка заготовки расположена не в центре заготовки, а в одном из углов. При этом нужно запрограммировать локальную систему координат (G52), далее программа совпадает с предыдущим примером той же разницей, что смещение надо запрограммировать в четыре цикла. В конце циклов необходимо выключить локальную систему координат, иначе расчёт остальных координат, заданных абсолютным значением, производится тоже от этой нулевой точки.

### 35 Подпрограмма программирования окружности центров с отражением:

```
%O7102(10.2)
N140 G16 G0 X40 Y-10
N150 G81 X50 YI10 R2 Z-10 L5
N160 G80 G15
N170 G0 Z100
N180 M99
%
```

Для правильной работы главной программы, представленной в предыдущем примере, требуется эта подпрограмма. Эта программа в прочем почти во всём совпадает с примером, представленным в главе 22. на странице 27.

### 36 Макропрограммирование - синусоидальная кривая

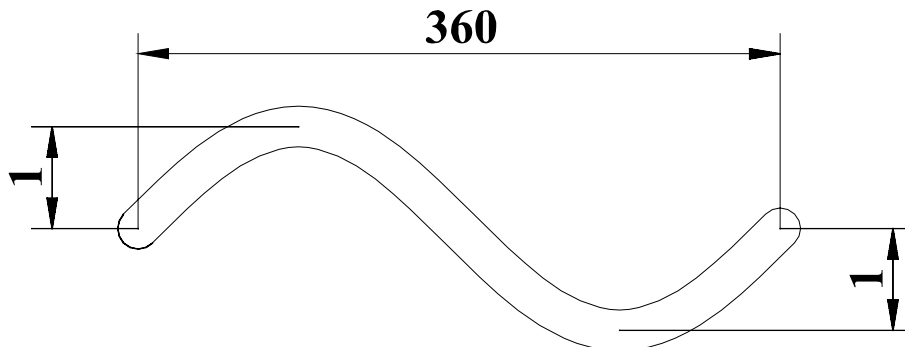


Рисунок 35

```

%O7111(11.1)
N100 T1
N110 G0 G90 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G1 Z-10
N150 #1=0
N160 WHILE[#1LE360] DO1
N170 #2=SIN#1
N180 G1 X#1 Y#2 F100
N190 #1=#1+0.1
N200 END1
N210 G0 Z50
N220 M30
%
```

В этом примере представляется программирование простой синусоидальной кривой с помощью макро переменных. Начало кривой находится в начале координат, и её продольная ось параллельна оси X. Для простоты отказались даже от коррекции инструмента. #1 является так называемой текущей переменной, которая  $0 < \#1 < 360$ . (Поскольку управление NC является частью технической жизни а не математической, естественно, что углы измеряются в градусах и не в радианах.) Вторая наша переменная #2 вычисляется из неё с помощью синусоидальной функции. Отсюда уже просто назначить #1 к координатам X, а #2 к координатам Y. Связав всё это в цикл, получим синусоидальную кривую.

Если начальной точкой требуемой кривой является не начало координат, далее её амплитуда не является единичной, или её длина не равна 360 мм, строка N180 может измениться следующим образом:

```
N180 G1 X[A+#1*B] Y[C+#2*D] F100,
```

где A и C - значение смещения (X;Y), B - коэффициент длины и D - коэффициент амплитуды.



### 37 Макропрограммирование - круговое фрезерование, интерполяция цилиндра

```
%O7112 (11.2)
N100 T1
N110 G43 Z50 H1
N120 S1000 M3 M8
N130 G0 X50 Y0
N140 #1=0
N150 WHILE[#1LE360] DO1
N160 #2=50*SIN[#1]
N170 #3=50*SIN[#1]
N180 G1 X#2 Y#3 Z[50-#1]
N190 #1=#1+1
N200 END1
N210 G0 Z50
N220 M30
%
```

Наша задача в этом примере заключается в программировании пространственного эллипса, проекцией которого в плоскости XY является окружность. На самом деле при этом программируется окружность, но в то же время позиция Z вычисляется из актуальной позиции X. Для этого требуется, чтобы программирование окружности была тоже параметрической. Окружность можно описывать проще всего зависимостью  $X=R*\cos\alpha$  и  $Y=R*\sin\alpha$ . Отсюда просто можно вычислить и координату Z. Текущей переменной является центральный угол.

### 38 Макропрограммирование - полушар

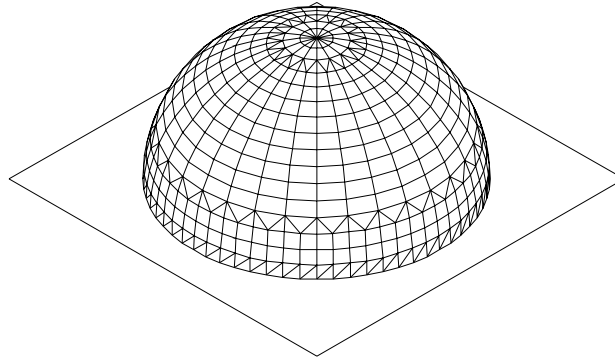


Рисунок 36

```

%O7118 (11.8)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 X-50 Y-50 Z0
N150 #1=-50
N160 WHILE[#1LE50] DO1
N170 #2=-50
N180 WHILE[#2LE50] DO2
N190 #3=0
N200 IF[[[#1*#1]+[#2*#2]]GT1600] GOTO220
N210 #3=SQRT[1600-[#1*#1]-[#2*#2]]
N220 G1 X#1 Y#2 Z#3 F1000
N230 #2=#2+1
N240 END2
N250 #1=#1+1
N260 IF[#1GE50]GOTO370
N270 #2=50
N280 WHILE[#2GE-50] DO2
N295 #3=0
N300 IF[[[#1*#1]+[#2*#2]]GT1600] GOTO320
N310 #3=SQRT[1600-[#1*#1]-[#2*#2]]
N320 G1 X#1 Y#2 Z#3
N330 #2=#2-1
N340 END2
N350 #1=#1+1
N360 END1
N370 M30
%
```

На рисунке изображён на плоскости 100x100 полушар R40. Траектория инструмента, описанная с помощью макропеременных, видна в следующей программе. Недостатком программы, написанной такой технологией является, что из-за чрезвычайно много расчётов не проявляется фактическая подача, а решающее - это время процессора. Это заметно тем, что между перемещениями инструмент приостановится, “думает”. С увеличением/ уменьшением разрешения - строки N230, N250, N330 и N350 скорость можно замедлить, или ускорить.

## 39 Макропрограммирование - порождение полушара

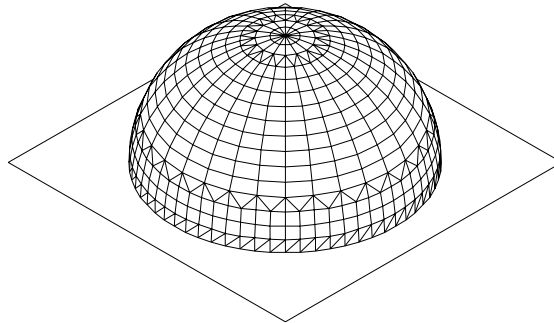


Рисунок 37

```

%O7119 (11.9)
N100 POPEN31
N110 DPRNT [O7120 (FELGOMB) ]
N120 DPRNT [T1]
N130 DPRNT [G54 G0 X-50 Y-50]
N140 DPRNT [G43 Z50 H1]
N150 DPRNT [G90 G01 S1000 M3 M8 F1000]
N160 #1=-50
N170 WHILE [#1LE50] DO1
N180 #2=-50
N190 WHILE [#2LE50] DO2
N200 #3=0
N210 IF [[[#1*#1]+[#2*#2]]GT1600] GOTO230
N220 #3=SQRT [1600-[#1*#1]-[#2*#2]]
N230 DPRNT [G1 X#1 [53] Y#2 [53] Z#3 [53]]
N240 #2=#2+1
N250 END2
N260 #1=#1+1
N270 IF [#1GE50] GOTO370
N280 #2=50
N290 WHILE [#2GE-50] DO2
N300 #3=0
N310 IF [[[#1*#1]+[#2*#2]]GT1600] GOTO330
N320 #3=SQRT [1600-[#1*#1]-[#2*#2]]
N330 DPRNT [G1 X#1 [53] Y#2 [53] Z#3 [53]]
N340 #2=#2-1
N350 END2
N360 #1=#1+1
N370 END1
N380 PCLOS31
N390 M30
%
```

пропускается чрезвычайно много расчётов, и проявляется фактическая подача, решает не время процессора. С увеличением/уменьшением разрешения - строки N240, N260, N340 и N360 - скорость порождения программы меняется, скорость обработки остаётся неизменной. Существенной разницей по сравнению с предыдущим примером является обращение с периферией. Важным моментом является открытие, закрытие периферии, в противном случае может повредиться программа, записанная в накопитель. Сложная задача оценить размер составленной программы, поэтому рекомендуется избежать запись непосредственно в накопитель а пользоваться последовательной периферией.

На рисунке - согласно предыдущему примеру - виден на плоскости 100x100 полушар R40. В этом примере покажем, как надо без макропеременных с помощью макропрограммы составить технологическую программу, пригодную для быстрого пробега. Преимуществом программы, написанной такой технологией по сравнению к предыдущим является, что во время пробега

