

**NCT<sup>®</sup> 99M**

**NCT<sup>®</sup> 2000M**

**Marógép és megmunkáló központ vezérlő**

**Példatár**

**(Szerkesztés alatt)**

---

Gyártó és fejlesztő: **NCT Ipari Elektronikai kft.**

H1148 Budapest Fogarasi út 7

: Levélcím: H1631 Bp. pf.: 26

F Telefon: (+36 1) 467 63 00

F Telefax:(+36 1) 363 6605

Villanyposta: [nct@nct.hu](mailto:nct@nct.hu)

Honlap: [www.nct.hu](http://www.nct.hu)

---

© Copyright NCT March 20, 2003

E leírás tartalmára minden kiadói jog fenntartva. Utánnnyomáshoz – kivonatosan is – engedélyünk megszerzése szükséges.

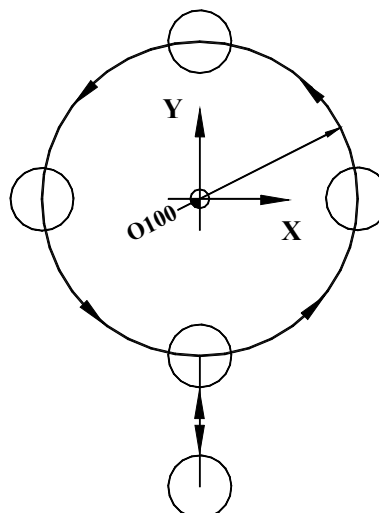
A leírást a legnagyobb körültekintéssel állítottuk össze és gondosan ellenőriztük, azonban az esetleges hibákért vagy téves adatokért *és az ebből eredő károkért felelősséget nem vállalunk.*

## TARTALOMJEGYZÉK:

1.1 Körinterpoláció programozása ráállással: .....	<u>5</u>
1.2 Körinterpoláció programozása kontúrkövetéssel: .....	<u>5</u>
1.3 Körinterpoláció programozása érintő irányú ráállással: .....	<u>6</u>
1.4 Körinterpoláció programozása kontúrkövétéssel: .....	<u>6</u>
2.1 Körinterpoláció belső kontúron: .....	<u>7</u>
2.2 Belső körinterpoláció kontúrkövetéssel: .....	<u>7</u>
2.3 Belső körinterpoláció programozása érintő irányú ráállással: .....	<u>8</u>
3.1 Egyenes kontúr programozása: .....	<u>9</u>
3.2 Egyenes kontúr programozása kontúrkövétéssel: .....	<u>9</u>
3.3 Nullkörök programozása kontúrkövetésnél: .....	<u>10</u>
3.4 Kontúrkövetés lekerekítésekkel: .....	<u>11</u>
4.1 Téglalap belső kontúrkövétéssel, érintő irányú ráállással: .....	<u>12</u>
5.1 Tetszőleges kontúr programozása kontúrkövetéssel: .....	<u>14</u>
5.2 Tetszőleges kontúr leírása belső kontúrkövetéssel .....	<u>15</u>
5.3 Tetszőleges belső kontúr (Alprogram) .....	<u>16</u>
6.1 Alprogramtechnika nullponteltolással: .....	<u>17</u>
6.2 Léptékezés használata alprogramhívással: .....	<u>17</u>
6.3 Léptékezés programozása tetszőleges pont körül: .....	<u>17</u>
7.1 Furatsor programozása: .....	<u>18</u>
7.2 Furatsorok programozása: .....	<u>19</u>
7.3 Lyukkör programozása .....	<u>19</u>
7.4 Lyukkör részlet programozása .....	<u>20</u>
7.5 Lyukkör programozása tetszőleges pozícióba: .....	<u>21</u>
7.6 Furatháló programozása alprogram segítségével: .....	<u>21</u>
7.7 Furatháló programozása ciklussal: .....	<u>22</u>
7.8 Furatháló programozása két egymásbaágyazott ciklussal: .....	<u>22</u>
8.1 Automatikus geometriai számítás körívek között (Belső kontúr) .....	<u>23</u>
8.2 Automatikus geometriai számítás körívek között (Külső kontúr) .....	<u>23</u>
8.3 Automatikus geometriai számítás körívek között (Belső kontúr) .....	<u>24</u>

## 1.1 Körinterpoláció programozása ráállással:

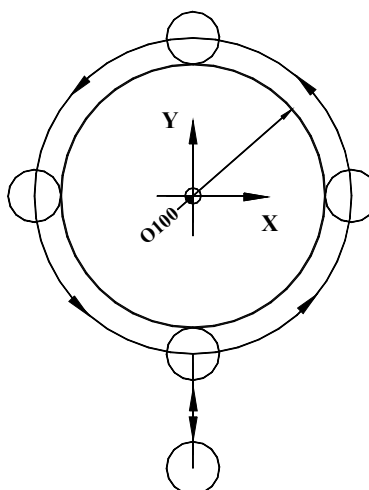
```
%O7011(1.1)
N100 T1
N110 G54 G0 X0 Y-100
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 X0 Y-50 F50
N170 G3 J50
N180 G1 X0 Y-100
N190 G0 Z100
N200 M30
%
```



Az első mondat a kívánt szerszám beváltását tartalmazza. A második mondatban szerepel az X-Y síkban a pozicionálás, a megfelelő koordinátarendszerben. A harmadik mondat a szerszám hosszkorrekciójának a figyelembevételéért felelős. A példatár többi programja hasonló programkezdéssel szerepel, ezért a továbbiakban nem térünk ki rá. A program elején szerepel még egy százalékjel, valamint egy O betűt követő négy számjegy, ami a program azonosítója. A program szintén egy százalékjellel fejeződik be. Ezeket a karaktereket csak külső eszközön való programozáskor kell kitenni, egyéb esetben a vezérlés önműködően létrehozza. A negyedik mondattól kezdődik a szerszám pályájának a leírása. A teljes kör programozása a J cím segítségével történt, ami a kör középpontjának a megadása inkrementálisan a kör kezdőpontjához képest. A program lezárása az M30 kóddal történik.

## 1.2 Körinterpoláció programozása kontúrkövetéssel:

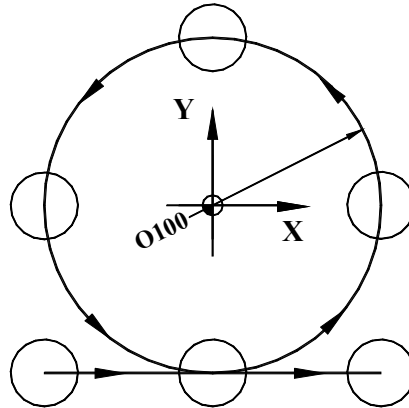
```
%O7012(1.2)
N100 T1
N110 G54 G0 X0 Y-100
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G42 X0 F50 D1
N170 G3 J-50
N180 G1 G40 Y-100
N190 G0 Z100
N200 M30
%
```



Az előző példához képest a különbség a kontúrkövetésben van. Az N160-as mondatban kapcsoljuk be a követést. Ilyenkor a vezérlés a mondat végpontját úgy módosítja, hogy a következő mondatot már a hivatkozott szerszámátmérővel eltolva kezdhesse. A leállítás hasonlóképpen működik, itt a mondat (N180) végpontján lesz nulla a korrekció.

### 1.3 Körinterpoláció programozása érintő irányú ráállással:

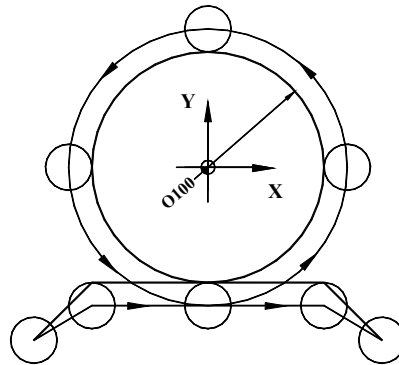
```
%O7013(1.3)
N100 T1
N110 G54 G0 X-50 Y-50
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 X0 F50
N170 G3 J50
N180 G1 X50
N190 G0 Z100
N200 M30
%
```



Az előző példához képest a különbség a ráállásban van. Mivel a tengelyek a kívánt pozíciót gyorsítás-lassítás használatával érik el, ezért az előző példában a szerszám a kör kezdőpontjában nyomot hagy. Ennek kiküszöbölésére illik minden esetben érintő irányban megközelíteni a kontúrt.

### 1.4 Körinterpoláció programozása kontúrkövétéssel:

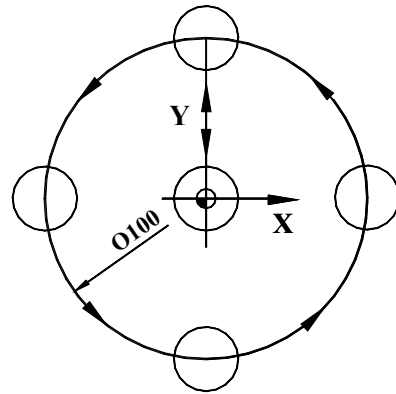
```
%O7014(1.4)
N100 T1
N110 G54 G0 X-70 Y-70
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G42 X-50 Y-50 D1 F50
N170 G1 X0
N180 G3 J50
N190 G1 X50
N200 G1 G40 X70 Y-70
N210 G0 Z100
N220 M30
%
```



Az előző példához képest a különbség a kezdeti pozícióban és a kontúrra ráállás, illetve leállásnál van. A ráálló és leálló koordinátákat programozzuk úgy, hogy a korrekció nélküli elmozdulás legyen nagyobb, mint a szerszám sugara.

## 2.1 Körinterpoláció belső kontúron:

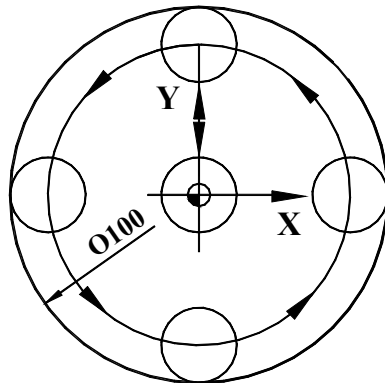
```
%O7021(2.1)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z0 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 X0 Y50 F50
N170 G3 J-50
N180 G1 X0 Y0
N190 G0 Z100
N200 M30
%
```



Ennél az esetnél is, csakúgy mint az 1.1-es példában a gondot a derékszögű rállás okozza.

## 2.2 Belső körinterpoláció kontúrkövetéssel:

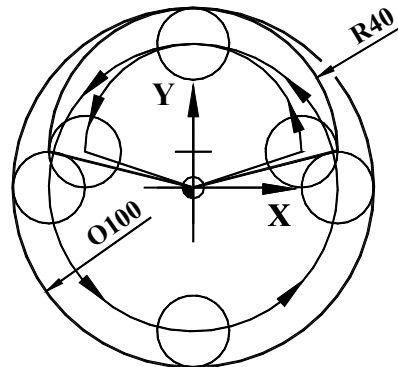
```
%O7022(2.2)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z0 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G41 X0 Y50 F50 D1
N170 G3 J-50
N180 G1 G40 X0 Y0
N190 G0 Z100
N200 M30
%
```



A külső kontúrkövetéshez képest csak a G41-G42 változott. A G41-G42 cserét a G2-G3 cseréjével is helyettesíthettük volna. A kör kezdőponti kijárást ez a megoldás sem küszöböli ki.

### 2.3 Belső körinterpoláció programozása érintő irányú ráállással:

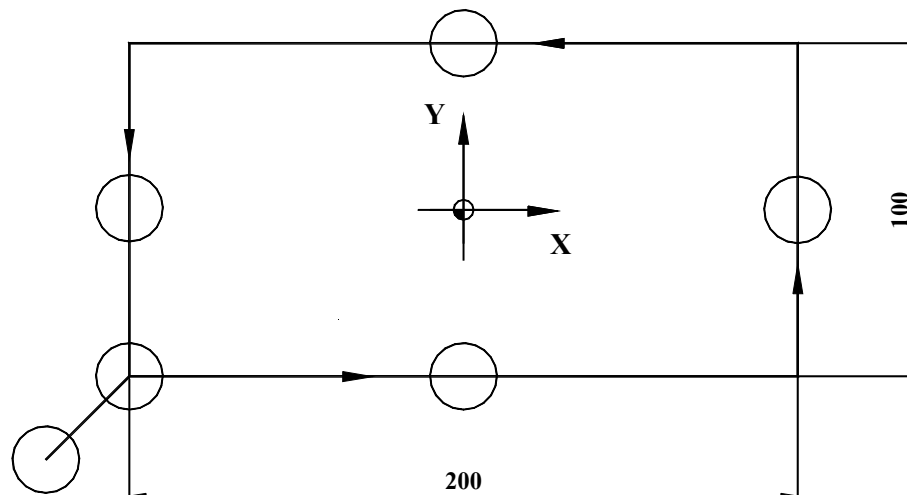
```
%O7023(2.3)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G41 X40 Y10 D1 F50
N170 G3 X0 Y50 R40
N180 G3 J-50
N190 G3 X-40 Y10 R40
N200 G1 G40 X0 Y0
N210 G0 Z100
N220 M30
%
```



A belső ívre ráállás a felületi hibák kiküszöbölése érdekében, csakúgy, mint a külső kontúrnál, érintő irányban történik, körinterpoláció alkalmazásával. Az érintőkör egy tetszőleges sugarú, a szerszám sugaránál nagyobb, de a kör sugaránál kisebb sugarú kör. A középpontja Y irányban a kontúr sugarának és az érintőkör sugarának a különbségével van eltolva, az eredeti körközépponthoz képest.



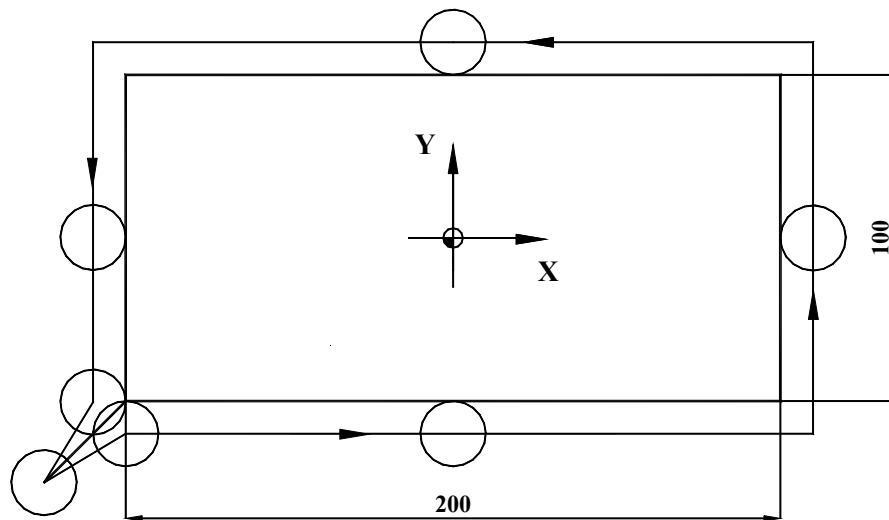
### 3.1 Egyenes kontúr programozása:



```
%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
%
```

A világ legegyszerűbb példája egy téglalap körbemarása. A G1-es kódok csak az áttekinthetőség miatt kerültek ismétlésre, az ismétlődő kódok a vezérlés szempontjából feleslegesek.

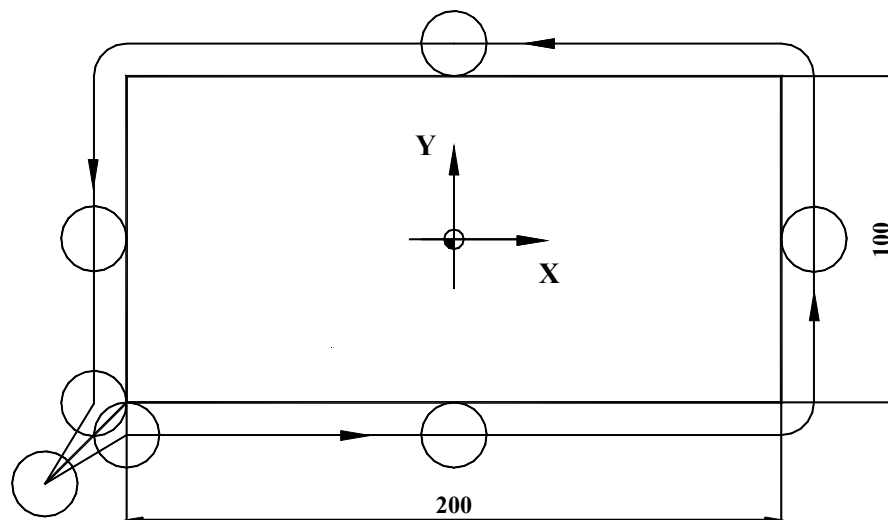
### 3.2 Egyenes kontúr programozása kontúrkövetéssel:



```
%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
%
```

A ráállás mértékét, csakúgy, mint az 1.4-es példánál, úgy kell meghatározni, hogy nagyobb legyen, mint a szerszám sugara. A ábrán vázolt esetben a kezdőponti sarkon a sorja megmaradhat, ezért célszerű a ráállásnál sarok előtt, a leállásnál a sarok után pár millimétert túlprogramozni.

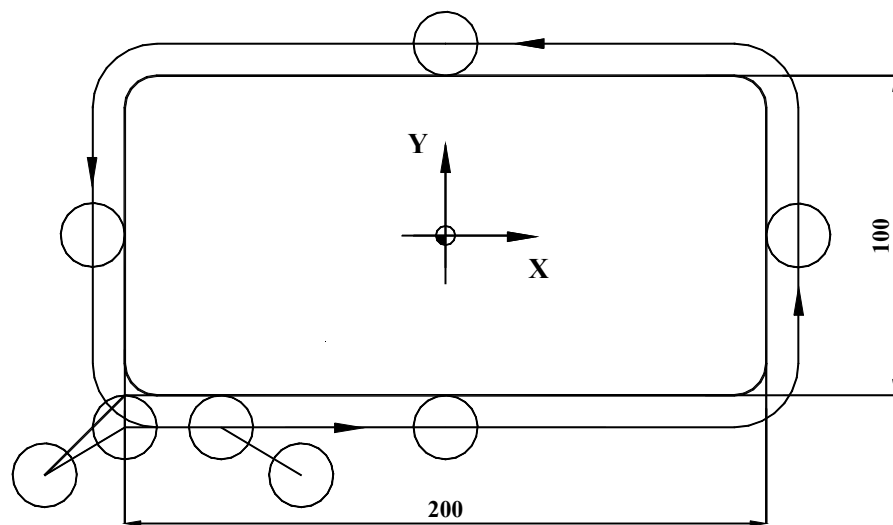
### 3.3 Nullkörök programozása kontúrkövetésnél:



```
%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
%
```

A ,R0-t a vezérlés egy nulla sugarú körívnek értelmezi, és így a kontúrkövetésnél a erre a szakaszra eső részt is körívként hajtja végre. A szerszám pályája sugara a szerszám sugrával lesz egyenlő, míg a kontúron éles sarok marad. Nagyobb mértékű nagyolási ráhagyásnál ez a változat a szerszám egyenletesebb terhelését segíti.

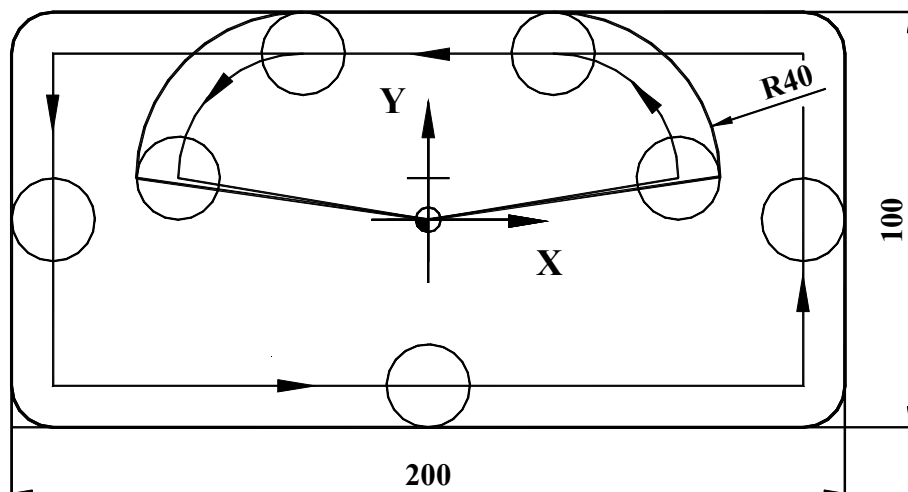
### 3.4 Kontúrkövetés lekerekítésekkel:



```
%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
%
```

A negyedik sarok lekerekítéséhez a szerszámot az első élre, legalábbis egy darabjára vissza kell vezetni. Ez a távolság legyen nagyobb, mint a lekerekítés és a szerszám sugarának az összege. A lekerekítéseknél a szerszám pályája sugra a lekerekítés és a szerszám sugarának az összegével egyezik.

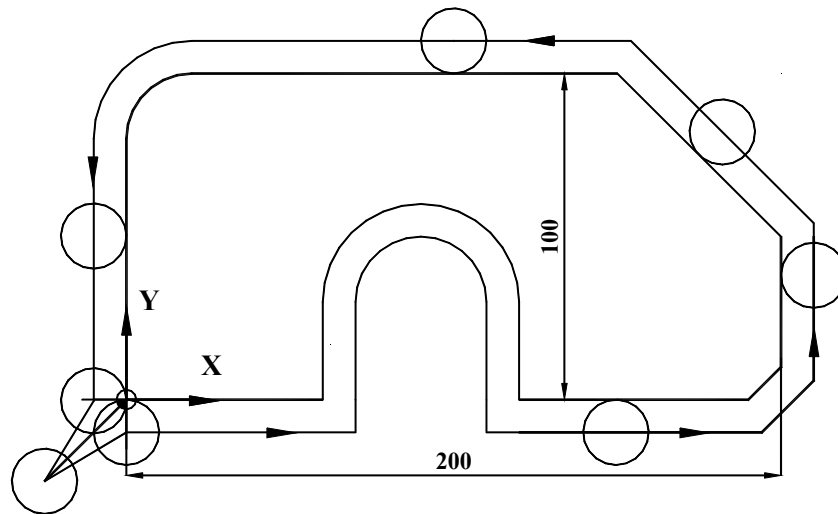
#### 4.1 Téglalap belső kontúrkövétéssel, érintő irányú ráállással:



```
%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
%
```

A ráállás itt is, mint a 3.2-es példákban egy körív mentén történik, amelynek sugara nagyobb, mint a szerszámsugár, de kisebb, mint az üreg Y irányú méretének a fele.

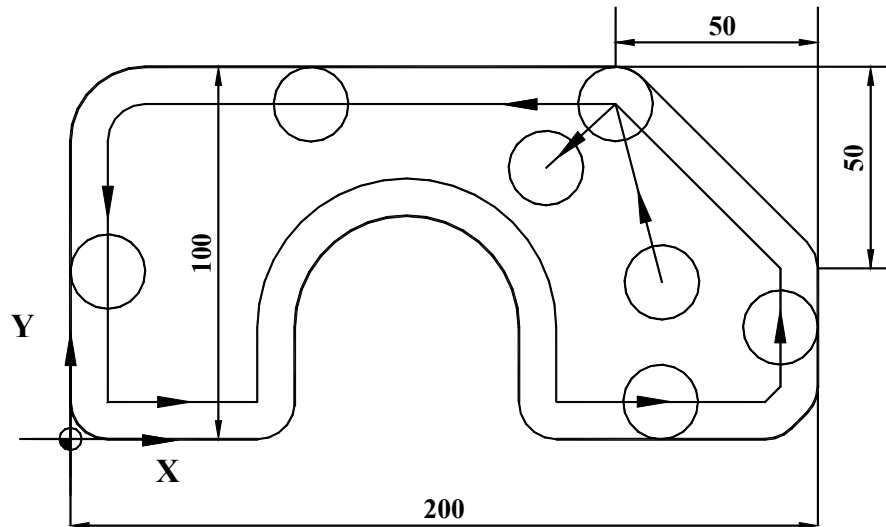
## 5.1 Tetszőleges kontúr programozása kontúrkövetéssel:



```
%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
%
```

Ebben a példában egy tetszőleges kontúr szerepel. A 210-es sorban letörés, a 240-as sorban lekerekítés szerepel. A letörést a 210-es sorban a ,C jelöli, ahol az előző és a következő sorral leírt egyenesekre a megadott távolságot méri fel a vezérlő és ezeket a pontokat köti össze, mint adódó kontúrt.

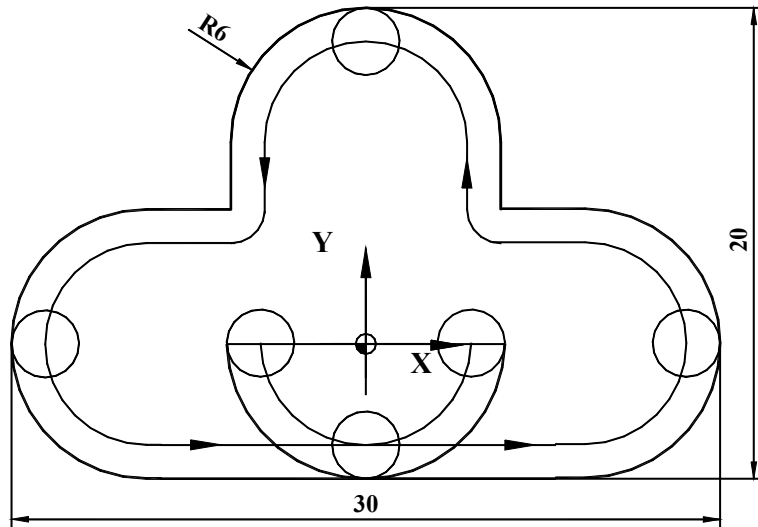
## 5.2 Tetszőleges kontúr leírása belső kontúrkövetéssel:



```
%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
%
```

A belső kontúrra ráállás a G41 I\_ J\_ címen történik az N160-as mondatban. I és J címen a megelőző mondat irányvektorát adhatjuk meg, ebben az esetben a vezérlő úgy áll a kontúrra, hogy nem a ráálló mondat, hanem a megadott irányvektor által meghatározott egyenest veszi a koordináták kiszámításánál figyelembe. Ez a módszer a leállásnál is hasonlóan működik. Az I és J címen csak az iránytangentst adjuk meg, tehát az abszolútérték mindegy, csak az előjel és a hányados számít.

### 5.3 Tetszőleges belső kontúr (Alprogram):



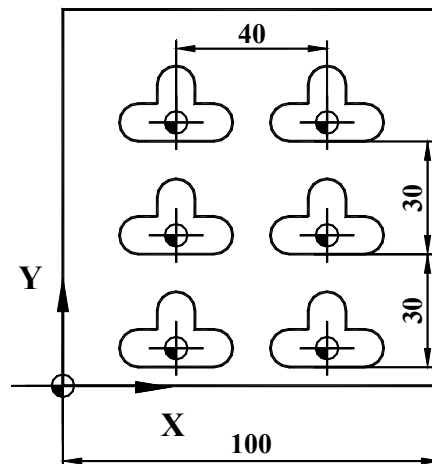
```
%O7053(5.3)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 Z5
N150 G1 Z-10 F20
N160 G1 G41 X-6 F50 D1
N170 G3 X0 Y-6 R6
N180 G1 X9
N190 G3 Y6 R6
N200 G1 X6
N210 G1 Y14
N220 G3 X-6 R6
N230 G1 Y6
N240 G1 X-9
N250 G3 Y-6 R6
N260 G1 X0
N270 G3 X6 Y0 R6
N280 G1 G40 X0 Y0
N290 G0 Z100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

A program mint alprogram van megírva, erre utal az M30 helyett a program végén az M99. Ha főprogramként indítjuk, végtelen ciklusként fut. Ezt a programot használjuk a későbbiekben, mint alprogram.



## 6.1 Alprogramtechnika nullponteltolással:

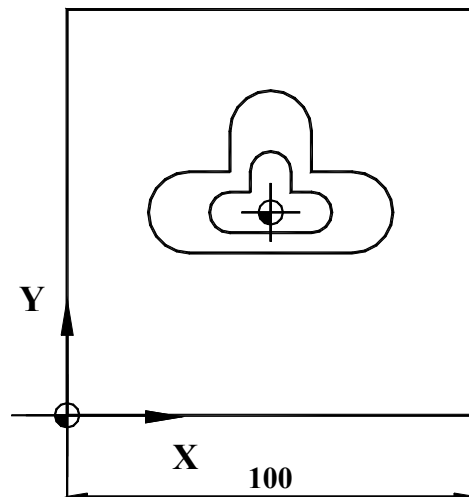
```
%O7061(6.1)
N100 T1
N110 G54
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G52 X30 Y10
N150 M98 P7053
N160 G52 X70 Y10
N170 M98 P7053
N180 G52 X30 Y40
N190 M98 P7053
N200 G52 X70 Y40
N210 M98 P7053
N220 G52 X30 Y70
N230 M98 P7053
N240 G52 X70 Y70
N250 M98 P7053
N260 G52 X0 Y0
N270 G0 Z100
N280 M30
%
```



Az alprogram abszolút méretmegadással leírt, az ábrán vázolt nullponttal készült. Az alprogram az 5.3-as példában van részletezve. A főprogramban páronként egy lokális koordináta-rendszer-eltolás és egy alprogramhívás szerepel. Az alprogramot az M99 parancs zárja le.

## 6.2 Léptékezés használata alprogramhívással:

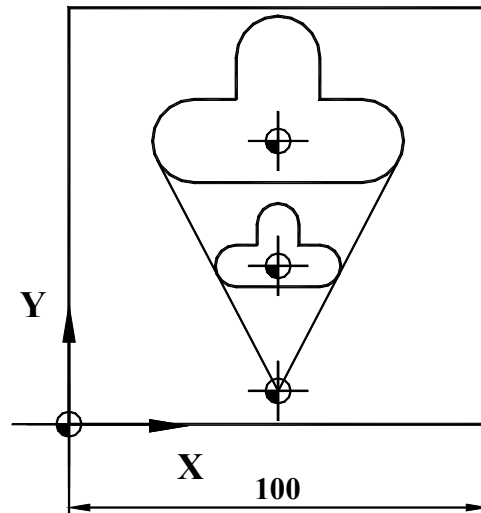
```
%O7062(6.2)
N100 T1
N110 G54
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G52 X50 Y50
N150 G51 X0 Y0 P2
N160 M98 P7053
N170 G50 X0 Y0
N180 G52 X0 Y0
N190 G0 Z100
N200 M30
%
```



A G52 a lokális koordináta-rendszer beállításáért felelős, a G51 a léptékezésbekapcsolásáért a felelős. A G51-es mondatban megadott koordináták, mint középpont körül egy a P címen megadott értékkel nagyítja a vezérlő a következő elmozdulásokat. A kontúrleírás itt is alprogramban történik.

### 6.3 Léptékezés programozása tetszőleges pont körül:

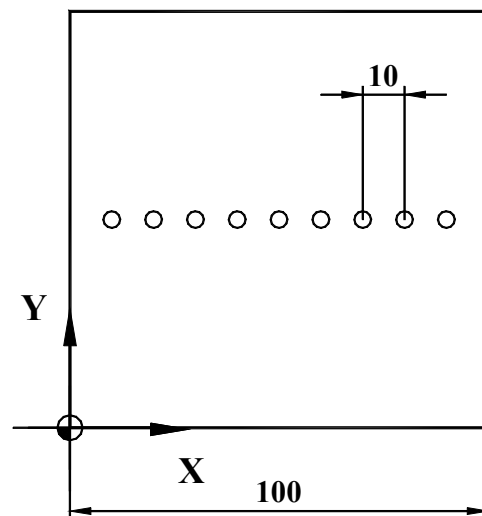
```
%O7063(6.3)
N100 T1
N110 G54
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G52 X50 Y50
N150 G51 X0 Y-40 P2
N160 M98 P7053
N170 G50 X0 Y0
N180 G52 X0 Y0
N190 G0 Z100
N200 M30
%
```



A G51-es mondatban a megadott koordinátákat a G52-es nullponteltoláshoz képest kell megadni, ezért szerepel Y címen -40. Ebben az esetben a nullpont is tolódik a G51-es mondatban megadott pont, mint középponti nagyítási pont szerint.

### 7.1 Furatsor programozása:

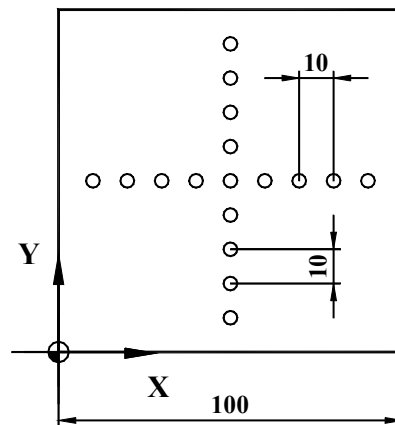
```
%O7071(7.1)
N100 T1
N110 G54 G0 X0 Y50
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G81 XI10 Y50 R2 Z-10 L9
N150 G80
N160 G0 Z100
N170 M30
%
```



Az egymásról 10 mm-re lévő furatok programozása inkrementális módon, ismétléssel lett megadva. Ebben az esetben figyelni kell arra, mivel a pontok megadása nem abszolút értékkel történt, hogy egy pozícionáló mondatnál az első furat **elő** kell állni. A ciklust az inkrementális távolság lelépésével kezdi, majd következik a fúrás, és ezt ismétli az L címen megadott értékkel. A furóciklust a G80 kód zárja le.

## 7.2 Furatsorok programozása:

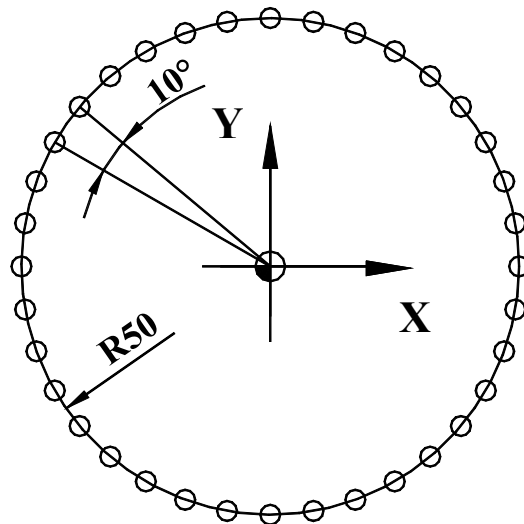
```
%O7072(7.2)
N100 T1
N110 G54 G0 X0 Y50
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G81 XI10 Y50 R2 Z-10 L9
N150 G80
N160 G0 X50 Y0
N170 G81 X50 YI10 R2 Z-10 L4
N180 G80
N190 G0 X50 Y50
N200 G81 X50 YI10 R2 Z-10 L4
N210 G80
N220 G0 Z100
N230 M30
%
```



A második furatsort egy új furóciklusként programozva, szintén inkrementálisan adhatjuk meg a méreteket. Mivel a középső furatot mindkét cilus tartalmazná, ezért a második sort külön kell szedni alsó illetve felső részre.

## 7.3 Lyukkör programozása

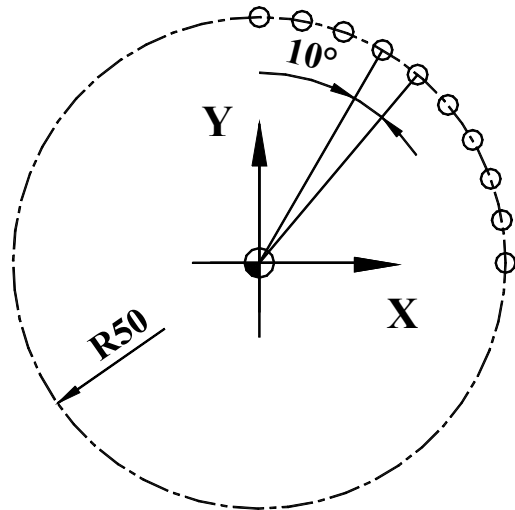
```
%O7073(7.3)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G16 G0 X50 Y-10
N150 G81 X50 YI10 R2 Z-10 L36
N160 G80 G15
N170 G0 Z100
N180 M30
%
```



A lyukkör programozása a furatsortól annyiban különbözik, hogy a polárkoordinátás adatmegadást használunk. A pozicionálás szintén az első furat elé történik, de ennek teljes kör esetén nincs jelentősége.

## 7.4 Lyukkör részlet programozása

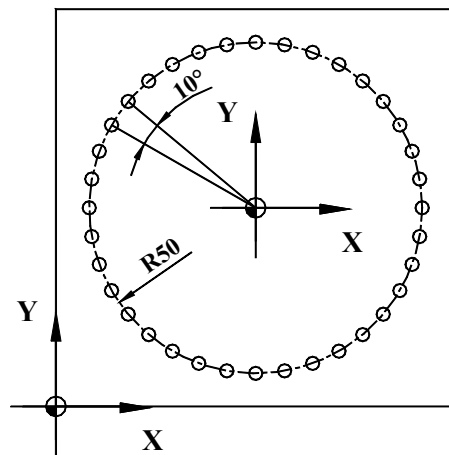
```
%O7074(7.4)
N100 T1
N110 G54 G0 X0 Y0
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G16 G0 X50 Y-10
N150 G81 X50 YI10 R2 Z-10 L10
N160 G80 G15
N170 G0 Z100
N180 M30
%
```



A lyukkör részlet programozása a lyukkör programozásától annyiban különbözik, hogy az ismétlési szám és a szöghelyesítés szorzata nem adja ki a 360 fokot. A pozicionálás szintén az első furat elé történik.

## 7.5 Lyukkör programozása tetszőleges pozícióba:

```
%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
%
```



A munkadarabnullpont ebben az esetben nem a darab közepén, hanem az egyik sarokban helyezkedik el. Ekkor egy lokális koordinátarendszert (G52) kell programozni, majd a további program egyezik az előző példával. A ciklus végén a lokális koordinátarendszert ki kell kapcsolni, különben a további, abszolút értékkel megadott koordináták is ettől a nullponttól kerülnek számításra.

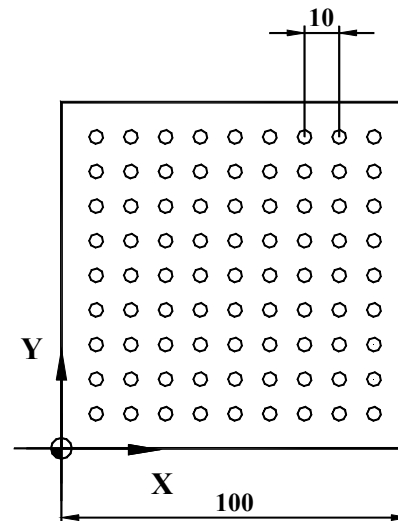
## 7.6 Furatsor programozása, mint alprogram:

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

A furatsor programozása teljesen hasonló, mint az előző példákban, azzal a különbséggel, hogy a furatsor végi X pozicionálást is tartalmazza. Így, ha főprogramból megadott számmal meghívjuk, egy furathálót kapunk.

## 7.7 Furatháló programozása alprogram segítségével:

```
%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
%
```

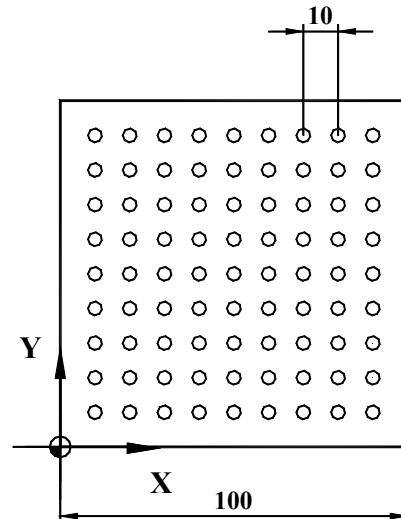


Ezt a feladatot több, különböző módszerrel lehet megoldani. A legegyszerűbb módszer, ha a vízszintes furatsorokat alprogramként programozzuk, és az alprogramot hívjuk meg többször. Az alprogramban van egy nullponteltolás, amit a főprogram végén kell kikapcsolni. Itt látható a program elején és végén a százalékjel és a programszám, programnév, abból a célból, hogy könnyen el lehessen különíteni a programokat egymástól.

### 7.8 Furatháló programozása ciklussal:

```

%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
%
```

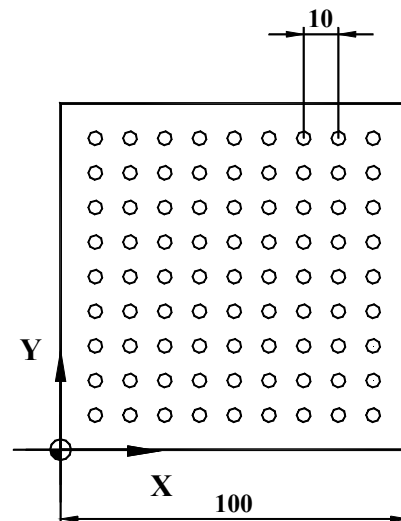


A második módszer, hogy a függőleges furatsorokat egy belső ciklusba szervezzük. Ebben az esetben nincs szükség alprogramra, de be kell vezetni úgynevezett makróváltozókat. Ezeket a makróváltozókat értékadással definiáljuk, és tetszőlegesen felhasználhatók. A ciklust a WHILE utasítás jellemzi, amely az ADDIG szóval jellemhető. A ciklus elejét a DO1, a végét az END1 jelzi, ahol a számok az összetartozást jelölik.

### 7.9 Furatháló programozása két egymásbaágyazott ciklussal:

```

%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
%
```

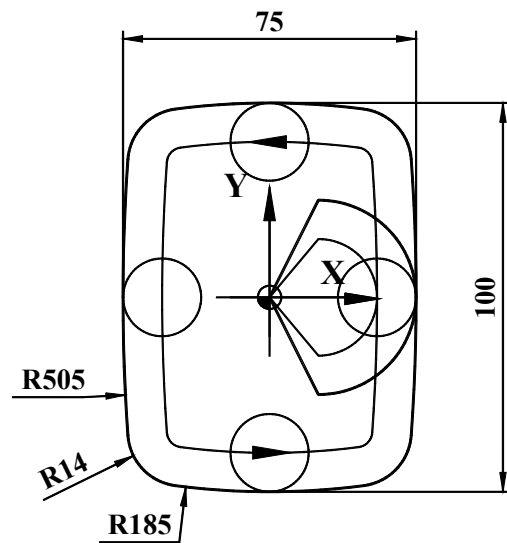


A harmadik esetben a két egymásba ágyazott ciklusnál a makróváltozókat megfeleltetjük a furatok X, illetve Y koordinátáinak. Így a makróváltozó kezdőértéket az első koordinátájára, végértéket az utolsó koordinátáira állítjuk be. A makróváltozók növelését a koordináták közötti különbséggel végezzük.

### 8.1 Automatikus geometriai számítás körívek között (Belső kontúr)

```

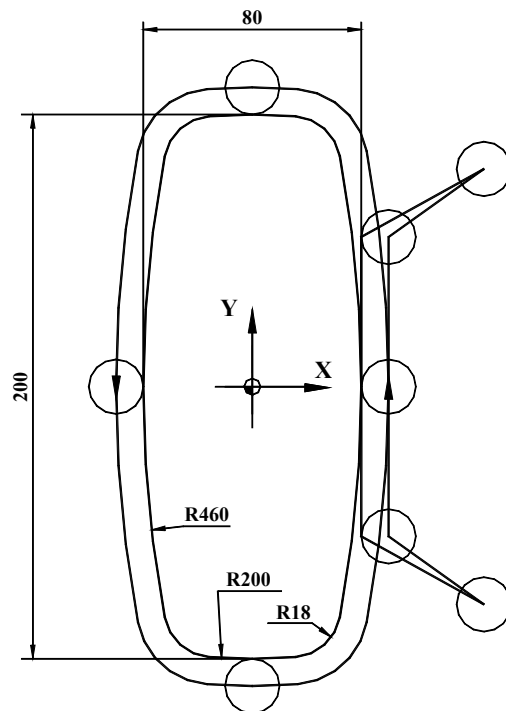
%O7081(8.1)
N100 G54 G90 G17 G0
N110 T1
N120 G43 Z50 H1
N130 S1000 M3
N140 G0 X0 Y0
N150 G0 Z2
N160 G1 Z-19
N170 G0 X0 Y0
N180 G41 G1 X17.5 Y-20 D1
N190 G3 X37.5 Y0 R20
N200 G3 XI-505 YI505 R505 ,R14
N210 G3 I0 J-135 R185 Q-1 ,R14
N220 G3 I467.5 J0 R505 Q-1 ,R14
N230 G3 I0 J135 R185 Q-1 ,R14
N240 G3 X37.5 Y0 I-467.5 J0 R505 Q-1
N250 G3 X17.5 Y20 R20
N260 G1 G40 X0 Y0
N270 Z50
N280 M30
%
```



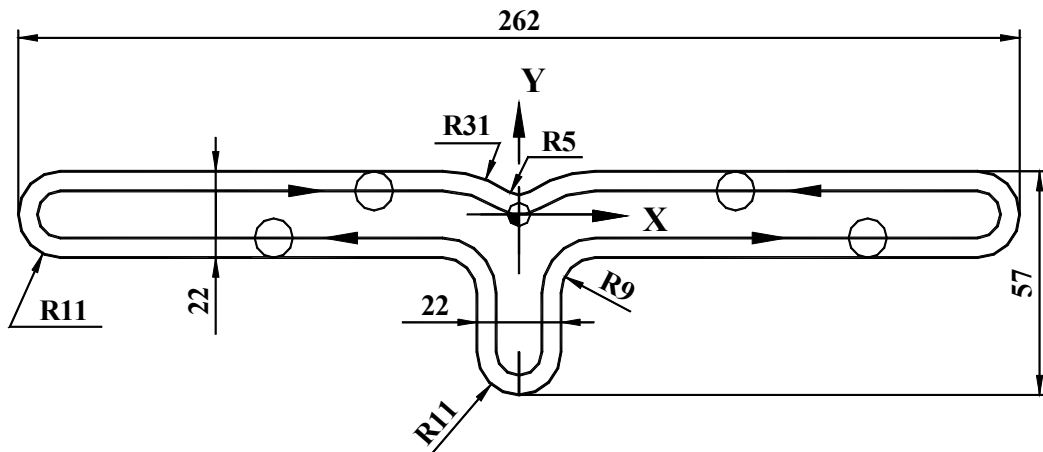
### 8.2 Automatikus geometriai számítás körívek között (Külső kontúr)

```

%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 I0 J-100 R200 Q-1 ,R18
N220 G3 I420 J0 R40 Q-1 ,R18
N230 G3 I0 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
%
```



### 8.3 Automatikus geometriai számítás körívek között (Belső kontúr)



```

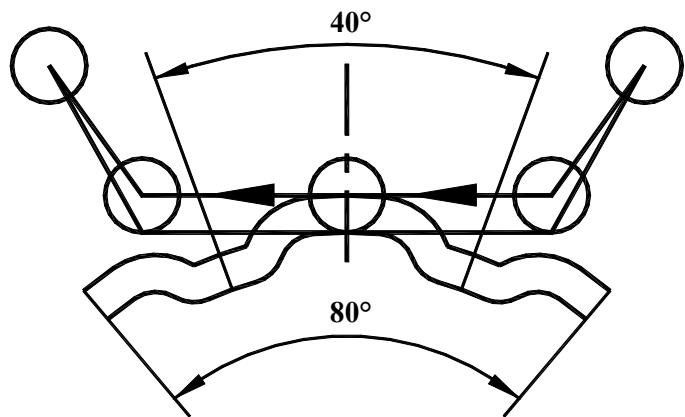
%O7083(8.3)
N100 G17 G0 G90 G94
N110 T1
N120 G0 X0 Y0
N130 G43 Z50 H1
N140 S1000 M3
N150 G0 Z-50
N160 G0 X0 Y-35
N170 G1 G41 Y-46 D1
N180 G3 X11 Y-35 R11
N190 G1 Y-11 ,R9
N200 G1 X120
N210 G3 X120 Y11 R11
N220 G1 X20
N230 G3 X-11 Y-20 R31 ,R5
N240 G3 X-20 Y11 I-20 J-20 R31 Q1
N250 G1 X-120
N260 G3 Y-11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

Amennyiben egy 22-es szármarót alkalmazunk, a 11-es rádiuszokat a maró testesíti meg, és így a két irányú marópálya egybeesik. A ráállásnál ebben az esetben az érintő irányú megközelítéstől eltekintettünk.



## 9.1 Fogazás programozása alprogram

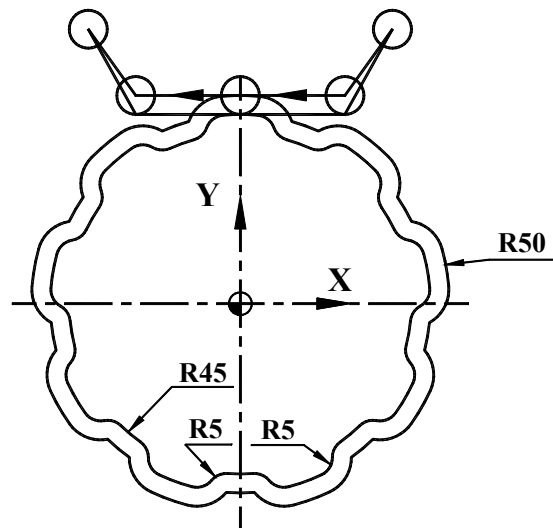
```
%O7091(9.1)
N100 G1 X0 Y50
N110 G3 X-4.587 Y49.789 R50
N120 G3 X-8.846 Y46.468 R5
N130 G2 X-12.206 Y43.313 R5
N140 G3 X-15.391 Y42.286 R45
N150 G3 X-18.490 Y41.026 R45
N160 G2 X-23.093 Y41.282 R5
N170 G3 X-28.490 Y41.090 R5
N180 G3 X-32.139 Y38.302 R50
N190 G68 X0 Y0 RI40
N190 M99
%
```



Ebben a példában, mint alprogramban egy fog koordinátáit kell leírni. Sajnos, mint a fogazásoknál általában a fog profilja egyszerű geometria számításokkal nem leírható, ezért a profil koordinátáit valamilyen CAD/(CAM) rendszer segítségével kell kiszámoltatni. A G68-as sorban a koordináta-rendszerinkrementális forgatását valósítjuk meg.

## 9.2 Fogazás programozása főprogram

```
%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
%
```

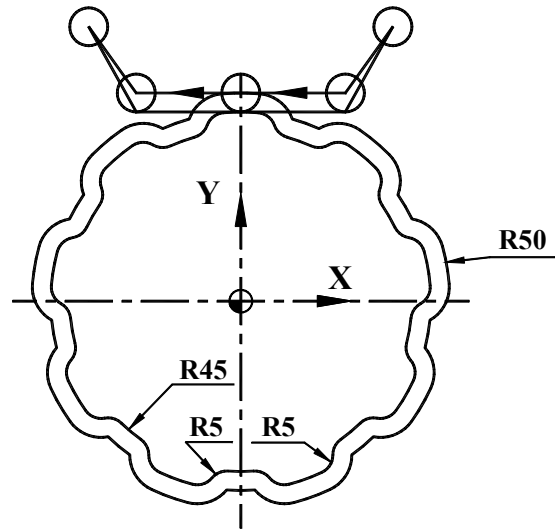


Ebben a programban az alprogramot hívjuk meg annyiszor, amennyi a fogsám. A program végén a koordináta-rendszer-elforgatás kikapcsolása, valamint a kontúr elhagyása szükséges.

### 9.3 Fogazás programozása egy programmal

```

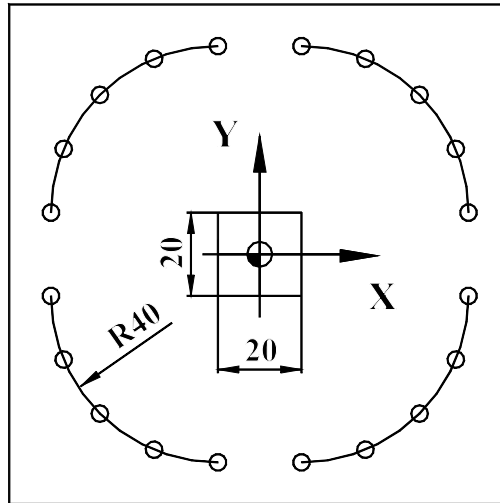
%O7093(9.3)
N100 T1
N110 G54 G0 X40 Y70
N120 G43 Z50 H1
N130 S1000 M3 M8
N140 G0 G42 X30 Y50 D1
N150 G1 X0 Y50
N160 #1=0
N170 WHILE[#1LT9] DO1
N180 G68 X0 Y0 R[#1*40]
N190 G1 X0 Y50
N200 G3 X-4.587 Y49.789 R50
N210 G3 X-8.846 Y46.468 R5
N220 G2 X-12.206 Y43.313 R5
N230 G3 X-15.391 Y42.286 R45
N240 G3 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
%
```



A makróváltozók segítségével és egy belső ciklus használatával a programba ágyazható a fogprofil, valamint lehetőség van az elforgatás abszolút megadására is. A kész kontúr minden tekintetben egyezik az előző példával.

### 10.1 Lyukkör programozása tükrözéssel alprogramban:

```
%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
%
```



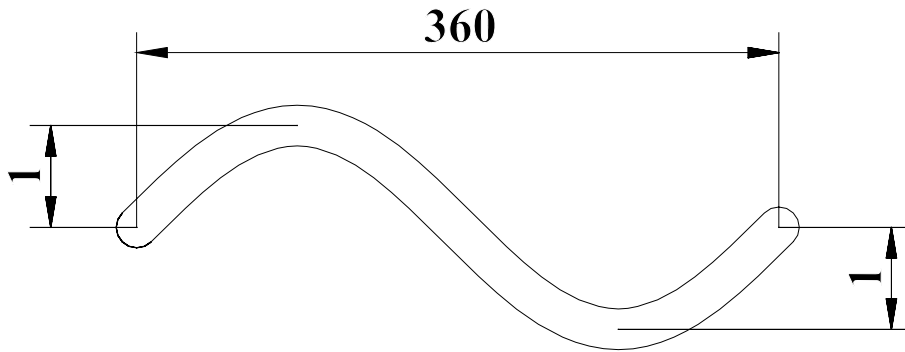
A munkadarabnullpont ebben az esetben nem a darab közepén, hanem az egyik sarokban helyezkedik el. Ekkor lokális koordináta-rendszert (G52) kell programozni, majd a program egyezik az előző példával, azzal a különbséggel, hogy négy ciklusban programozzuk az eltolást. A ciklusok végén a lokális koordináta-rendszert ki kell kapcsolni, különben a további, abszolút értékkel megadott koordináták is ettől a nullponttól kerülnek számításra.

### 10.2 Lyukkör programozása tükrözéssel alprogramban:

```
%O7102(10.2)
N140 G16 G0 X40 Y-10
N150 G81 X50 Y110 R2 Z-10 L5
N160 G80 G15
N170 G0 Z100
N180 M99
```

Az előző példában bemutatott főprogramnak a helyes működéshez erre az alprogramra van szüksége. Ez a program egyébként szinte teljesen egyezik a 7.4 -es példában ismertetett példával.

## 11.1 Makróprogramozás - szinuszgörbe



Ebben a példában egy egyszerű szinuszgörbe programozását mutatjuk be makróváltozók segítségével. A görbe kezdete az origó, és a hossz tengelye az X tengellyel párhuzamos. Az egyszerűség kedvéért a szerszámkorrekciótól is eltekintünk. A #1 az úgynevezett futó változó, ami  $0 < \#1 < 360$ . (Mivel az NC vezérlő a műszaki élet része és nem a matematikáé, természetesen a fokokat szögben mérjük és nem radiánban.) A másik változónk a #2 ebből számítható a szinuszfüggvény segítségével. Innen már egyszerű az X koordinátához a #1, míg az Y koordinátához a #2 hozzárendelése. Ezt a egészet egy ciklusba fűzve, megkapjuk a szinuszgörbét.

```
%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
%
```

Amennyiben a szükséges görbe kezdőpontja nem az origó, illetve amplitúdója nem egységnyi, vagy a hossza nem 360 mm, az N180-as sor a következőképpen módosulhat:

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

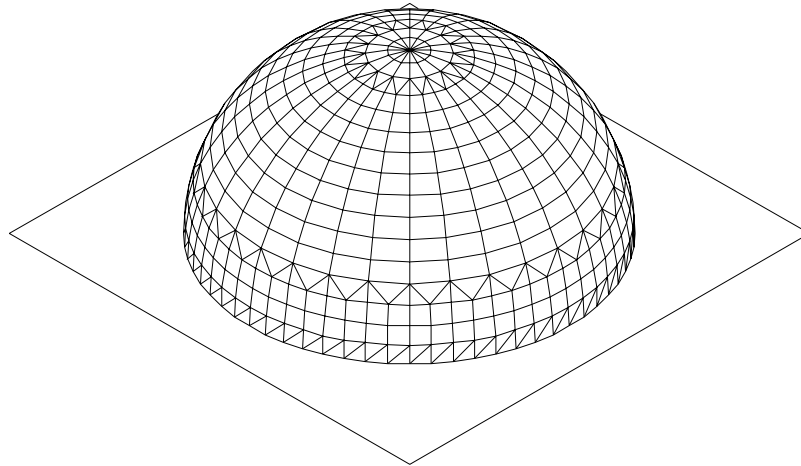
, ahol A és C az eltolásérték (X;Y), B a hossztenyező és D az amplitúdótenyező.

## 11.2 Makróprogramozás - körmarás, hengerinterpoláció

Ebben a példában egy térbeli ellipszis programozása a feladat, amelynek XY síkú vetülete egy kör. Ekkor tulajdonképpen egy kört programozunk, de közben a Z pozíciót a mindenkori X pozícióból számítjuk. Ehhez arra van szükség, hogy a körprogramozás is parametrikus legyen. a Kör legegyszerűbben az  $X=R*\cos\alpha$  és az  $Y=R*\sin\alpha$  összefüggéssel írható le. Innen egyszerűen számítható a Z koordináta is. A futóváltozó a középponti szög.

```
%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
```

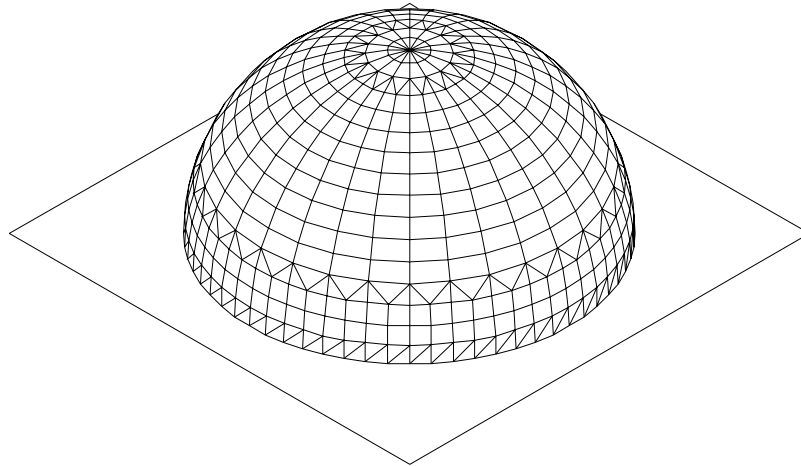
## 11.8 Makróprogramozás - félgömb



Az ábrán egy 100x100-as síkon egy R40-es félgömb látható. Ennek a makróváltozók segítségével leírt szerszám pályája látható a következő programban. Az ilyen technológiával megírt program hátránya, hogy a rendkívül sok számítás miatt a tényleges előtolás nem érvényesül, hanem a processzoridő dönt. Ezt arról vesszük észre, hogy az elmozdulások között a szerszám megáll, “gondolkodik”. A felbontás finomításával/durvításával - N230, N250, N330 és N350-es sorok a sebesség lassítható, gyorsítható.

```
%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
%
```

## 11.9 Makróprogramozás - félgömb szülés



Az ábrán - az előző példával megegyezően - egy 100x100-as síkon egy R40-es félgömb látható. Ebben a példában azt mutatjuk be, hogyan kell egy makró segítségével egy gyors futásra alkalmas, makróváltozók nélküli technológiai programot készíteni. Az ilyen technológiával megírt program előnye az előzőhöz képest, hogy a rendkívül sok számítás a futásidő alatt elmarad, és a tényleges előtolás érvényesül, nem a processzoridő dönt. A felbontás finomításával/durvításával - N240, N260, N340 és N360-es sorok - a programszülési sebesség változik, a megmunkálási sebesség változatlan marad. Lényeges különbség - az előző példához képest - a periféria kezelés. Fontos momentum a periféria megnyitása, lezárása, ellenkező esetben a háttértárba íratott program megsérülhet. Nehézkes dolog a készített program méretének becslése, ezért ajánlott a közvetlen tárbairást kerülése és helyette a soros periféria használata.

```
%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] GOTO220
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] GOTO320
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
%
```

