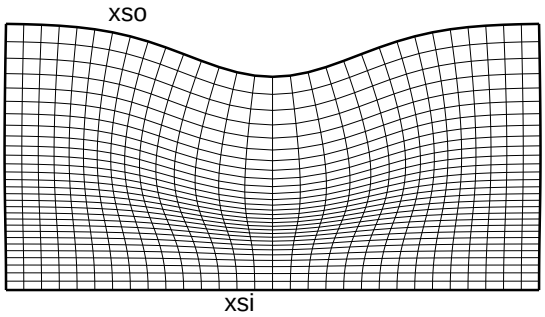
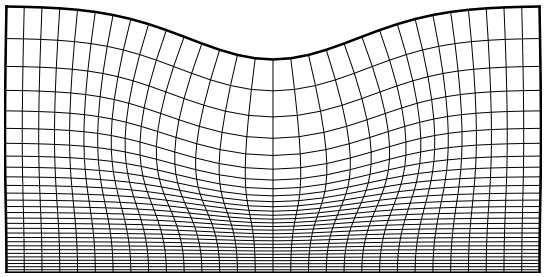
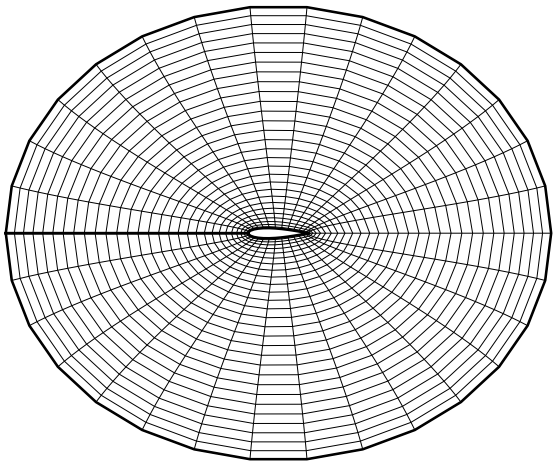
Two surface method for grid generation



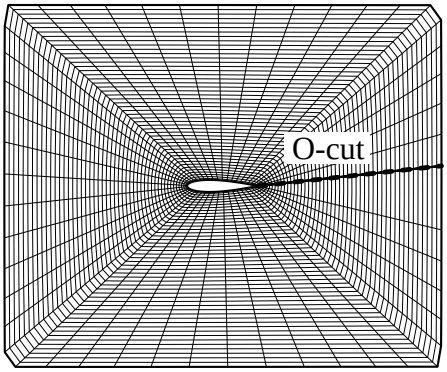
H-grid



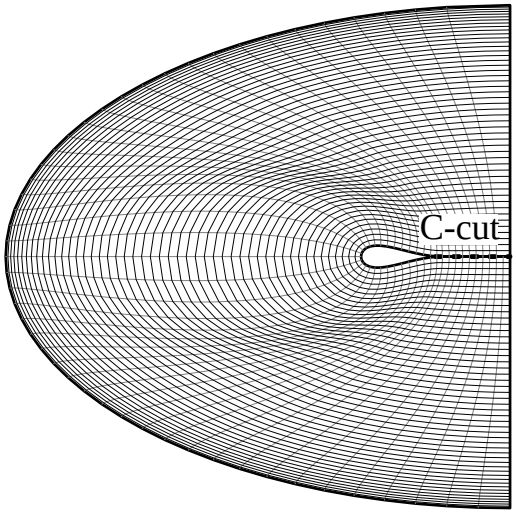
H-grid, with strecthing



O-grid (periodic)



O grid in rectangular domain



C-grid

```
!*****   Two Surface Method, Pseudo code  *********************
!  Arrays (xso,yso) and (xsi,ysi) define outer and inner walls
!  NB: Assumes that normal direction is (-dy,dx)
!        If grid goes into wall, sign of normal is reversed
!***********************************************************************
        READ xo(:) , yo(:)    ! coordinates of outer wall
        READ xi(:) , yi(:)    !   inner wall

xloop: DO i = 1,NX

!*********
! Evaluate unit normal at inner and outer walls
!*********
     (xni,yni)  =  (-dyi,dxi)/sqrt(dxi^2+dyi^2)  = normal to inner wall
     (xno,yno)  =  (-dyo,dxo)/sqrt(dxo^2+dyo^2)  = normal to outer wall
          E.G. dxi = xi(i+1)-xi(i-1)

!*********
! Hermite interpolation: define cubic polynomials H_i(s) such that
!
!  H1(0)=1, H1(1)=0, H1'(0)=0, H1'(1)=0
!  H2(0)=0, H2(1)=1, H2'(0)=0, H2'(1)=0
!  H3(0)=0, H3(1)=0, H3'(0)=1, H3'(1)=0
!  H4(0)=0, H4(1)=0, H4'(0)=0, H4'(1)=1
!
!*********

      Pp = 1.0   ! adjustable parameter for inner wall
      Qq = 2.0   ! parameter for outer wall

yloop: DO j=1,NY
          s  =  float(j-1)/float(NY-1)
          H1 =  1-3.*s^2+2.*s^3
          H2 =  3.*s^2-2.*s^3
          H3 = (s^3-2.*s^2+s)
          H4 = (s^3-s^2)
          xx(i,j) = xi*H1+xo*H2+Pp*xni*H3+Qq*xno*H4   ! grid
          yy(i,j) = yi*H1+yo*H2+Pp*yni*H3+Qq*yno*H4   ! grid
   ENDDO yloop
ENDDO  xloop
```