

```

; -----
; Generation of intersecting tunnels of circular shape with flat floor
; Grid generation is based upon brick primitives that have surfaces
; adjusted to circular shape
; -----
;
;
def input_parameters ; interactive input
; zcentuna = in('Height of Center of Tunnel A :')
; zradtuna = in('Radius of Tunnel A :')
; zcentunb = in('Height of Center of Tunnel B :')
; zradtunb = in('Radius of Tunnel B :')
; ff_height = in('Far field height above crown :')
; ff_depth = in('Far field depth below base :')
; ff_len_a = in('Length of Tunnel A from intersection :')
; ff_len_b = in('Length of Tunnel B from intersection :')
; zone_x_a = in('Number of Zones for Tunnel A Width :')
; zone_x_b = in('Number of Zones for Tunnel B Width :')
; zone_z_a = in('Number of Zones for Tunnel A Length :')
; zone_z_b = in('Number of Zones for Tunnel B Length :')
; zone_y_tun = in('Number of Zones for Height of Tunnels :')
; zone_ff_a = in('Number of Zones above Tunnels :')
; zone_ff_b = in('Number of Zones below Tunnels :')
; zone_rat_a = in('Zone Ratio along length of Tunnel A :')
; zone_rat_b = in('Zone Ratio along length of Tunnel B :')
; zone_rat_z = in('Zone Ratio above and below Tunnels :')
;end

def input_parameters ; manual input
    zcentuna = 1.05
    zradtuna = 1.55
    zcentunb = 0.80
    zradtunb = 1.20
    ff_height = 10.0
    ff_depth = 5.0
    ff_len_a = 10.0
    ff_len_b = 5.0
    zone_x_a = 5
    zone_x_b = 5
    zone_z_a = 10
    zone_z_b = 5
    zone_y_tun = 5
    zone_ff_a = 10
    zone_ff_b = 10
    zone_rat_a = 1.2
    zone_rat_b = 1.2

```

```

    zone_rat_z = 1.2
end
;
; This routine calculates the position and creates 12 bricks
; that subdivide the tunnel intersection geometry.
; (Three levels of four bricks)
;
def calculate_bricks
    x0brick = 0.0
    x1brick = zradtuna
    x2brick = x1brick + ff_len_b
    y0brick = 0.0
    y1brick = zradtunb
    y2brick = y1brick + ff_len_a
    z0brick = -ff_depth
    z1brick = 0.0
    z2brick = zradtuna + zcentuna
    z3brick = z2brick + ff_height
    zone_rat_z_inv = 1.0 / zone_rat_z
    tunb_len = 2.0 * (ff_len_b + zradtuna)
    tunc_min = tunb_len - zradtuna
    tunc_plus = tunb_len + zradtuna
command
    ;; Brick 1
    gen zone brick size (zone_x_a,zone_x_b,zone_y_tun) &
        p0 (x0brick,y0brick,z1brick) p1(x1brick,y0brick,z1brick) &
        p2 (x0brick,y1brick,z1brick) p3(x0brick,y0brick,z2brick) &
        group Brick_1
end_command
command
    ;; Brick 2
    gen zone brick size (zone_x_a,zone_z_a,zone_y_tun) &
        p0 (x0brick,y1brick,z1brick) p1(x1brick,y1brick,z1brick) &
        p2 (x0brick,y2brick,z1brick) p3(x0brick,y1brick,z2brick) &
        ratio (1.0,zone_rat_a,1.0) &
        group Brick_2
end_command
command
    ;; Brick 3
    gen zone brick size (zone_z_b,zone_x_b,zone_y_tun) &
        p0 (x1brick,y0brick,z1brick) p1(x2brick,y0brick,z1brick) &
        p2 (x1brick,y1brick,z1brick) p3(x1brick,y0brick,z2brick) &
        ratio (zone_rat_b,1.0,1.0) &
        group Brick_3
end_command
command
    ;; Brick 4
    gen zone brick size (zone_z_b,zone_z_a,zone_y_tun) &

```

```

        p0 (x1brick,y1brick,z1brick) p1(x2brick,y1brick,z1brick) &
        p2 (x1brick,y2brick,z1brick) p3(x1brick,y1brick,z2brick) &
        ratio (zone_rat_b,zone_rat_a,1.0) &
        group Brick_4
end_command
command                                ;; Brick 5
    gen zone brick size (zone_x_a,zone_x_b,zone_ff_a) &
        p0 (x0brick,y0brick,z2brick) p1(x1brick,y0brick,z2brick) &
        p2 (x0brick,y1brick,z2brick) p3(x0brick,y0brick,z3brick) &
        ratio (1.0,1.0,zone_rat_z) &
        group Brick_5
end_command
command                                ;; Brick 6
    gen zone brick size (zone_x_a,zone_z_a,zone_ff_a) &
        p0 (x0brick,y1brick,z2brick) p1(x1brick,y1brick,z2brick) &
        p2 (x0brick,y2brick,z2brick) p3(x0brick,y1brick,z3brick) &
        ratio (1.0,zone_rat_a,zone_rat_z) &
        group Brick_6
end_command
command                                ;; Brick 7
    gen zone brick size (zone_z_b,zone_x_b,zone_ff_a) &
        p0 (x1brick,y0brick,z2brick) p1(x2brick,y0brick,z2brick) &
        p2 (x1brick,y1brick,z2brick) p3(x1brick,y0brick,z3brick) &
        ratio (zone_rat_b,1.0,zone_rat_z) &
        group Brick_7
end_command
command                                ;; Brick 8
    gen zone brick size (zone_z_b,zone_z_a,zone_ff_a) &
        p0 (x1brick,y1brick,z2brick) p1(x2brick,y1brick,z2brick) &
        p2 (x1brick,y2brick,z2brick) p3(x1brick,y1brick,z3brick) &
        ratio (zone_rat_b,zone_rat_a,zone_rat_z) &
        group Brick_8
end_command
command                                ;; Brick 9
    gen zone brick size (zone_x_a,zone_x_b,zone_ff_b) &
        p0 (x0brick,y0brick,z0brick) p1(x1brick,y0brick,z0brick) &
        p2 (x0brick,y1brick,z0brick) p3(x0brick,y0brick,z1brick) &
        ratio (1.0,1.0,zone_rat_z_inv) &
        group Brick_9
end_command
command                                ;; Brick 10
    gen zone brick size (zone_x_a,zone_z_a,zone_ff_b) &
        p0 (x0brick,y1brick,z0brick) p1(x1brick,y1brick,z0brick) &
        p2 (x0brick,y2brick,z0brick) p3(x0brick,y1brick,z1brick) &

```

```

        ratio (1.0, zone_rat_a, zone_rat_z_inv) &
        group Brick_10
end_command
command
    ;; Brick 11
    gen zone brick size (zone_z_b, zone_x_b, zone_ff_b) &
    p0 (x1brick, y0brick, z0brick) p1(x2brick, y0brick, z0brick) &
    p2 (x1brick, y1brick, z0brick) p3(x1brick, y0brick, z1brick) &
    ratio (zone_rat_b, 1.0, zone_rat_z_inv) &
    group Brick_11
end_command
command
    ;; Brick 12
    gen zone brick size (zone_z_b, zone_z_a, zone_ff_b) &
    p0 (x1brick, y1brick, z0brick) p1(x2brick, y1brick, z0brick) &
    p2 (x1brick, y2brick, z0brick) p3(x1brick, y1brick, z1brick) &
    ratio (zone_rat_b, zone_rat_a, zone_rat_z_inv) &
    group Brick_12
end_command
end

set logfile tunint.log
set log on
input_parameters
set log off

calculate_bricks

save tunint1.sav

call shape.fis

save tunint2.sav

; Reflects the geometry across symmetry planes

def reflect_tunnels
command
    gen zone reflect normal (1,0,0) origin (0,0,0)
end_command
x3brick = x2brick * 2.0
command
    gen zone reflect normal (1,0,0) origin (x2brick,0,0)
end_command
; NOTE! Final reflection is commented out. Use if reflecting
;     model about x-z plane

```

```

y3brick = -y2brick
command
    gen zone reflect normal (0,1,0) origin (0,0,0)
end_command
end

reflect_tunnels

save tunint3.sav

; Build some convenient named ranges

group TunA range group Brick_1 any group Brick_2 any
group TunB range group Brick_3 x 0 tunb_len
group TunC range group TunA      x tunc_min tunc_plus

plot create Tunnels
plot set cap size 25
plot set rot 20,0,20
plot set center 8.218,-.1459,1.811
plot set dist 83.64

plot add block group shade off yellow orange lblue &
        range group TunA any group TunB any group TunC any
plot add sketch
plot add axes

ret

```