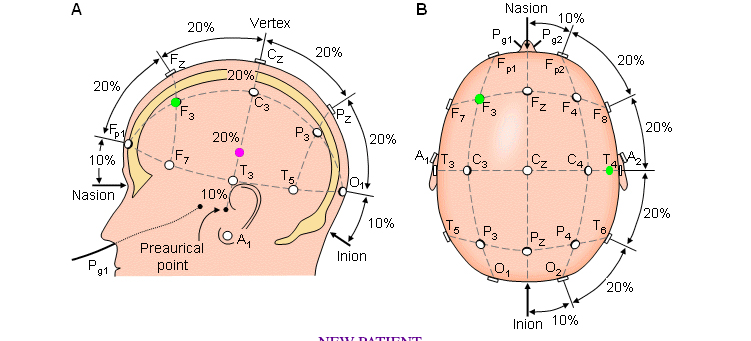
Variables for Beam project



**Nasion to Inion input = nasion\_to\_inion**

**Tragus to Tragus input = tragus\_to\_tragus**

**Circumference input = circumference**

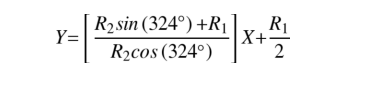
**R =** distance from the vertex to the point Fpz = ½(nasion\_to\_inion) – (10/100) \* ½(nasion\_to\_inion)

**R2 =** distance from the vertex to the point T3 = ½(tragus\_to\_tragus) – (10/100) \* ½(tragus\_to\_tragus)

**R1 =** radius of points

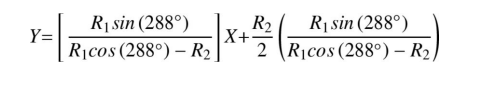
The coordinates for the points Fz , F7, Fp1, and C3 are now intersected by an imaginary circle with radius R\_1.

**Import math module**

****

Equation of the line containing points f7 and fz

Y = [(R2 \* degrees(math.sin(324)) + R1) / (R2 \* degrees(math.cos(324)))] \* x + (R1/2)

****

Equation of the line containing the points C3 and Fp1

Y = [(R1 \* degrees(math.sin(288))) / (R1 \* degrees(math.cos(288)) – R2))] \* X + (R2/2) \* ((R1 \* degrees(math.sin(288))) / R1 \* (degrees(math.cos(288)) – R2))

These equations are the lines intersecting point F3

**Find x**

x = (1/2) \* ((R1 + (R2 \* R1 \* degrees(math.sin(288))) / ((R1 \* degrees(math.sin(288))) / ((R1 \* degrees(math.cos(288))) – R2) – ((R2 \* degrees(math.sin(324)) + R1) / R2 \* degrees(math.cos(324))))

x = y \* (1/2) \*(((R2 \* degrees(math.sin(324)) + R1) / (R2 \* degrees(math.cos(324)))) \* (R1 + (R2 \* R1 \* degrees(math.sin(288))) / ((R1 \* degrees(math.cos(288))) – R2) / ((R1 \* degrees(math.cos(288))) – R2) – (R2 \* degrees(math.sin(324) + R1) / (R2 \* degrees(math.cos(324))))

(x,y) is the cartesian coordinate of point F3

r = radius of point x, line to point x

r\_corrected = r \* 0.9

angle = angle from the line from tragus to tragus of F3

(r\_corrected, angle) is the polar coordinate of point F3 from vertex

midline\_angle = angle from the midline from tragus to tragus of F3 = Φ = 90 - angle

distance(u) to F3 along the circumference(c) beginning at the midline (u/c) = Φ/360

u = Φc/360

Thus we multiply the radius coordinate of the polar coordinates for F3 by 0.9 in order to account for the head not being a sphere.

**Output:**

Once this value has been input into the program, it produces two output values

The first is the distance to a point (we'll call point-x) along the circumference from the centerline (in cm's) = (F3) Distance along circumference from midline (X) = (midline\_angle \* ((1/2) \* circumference)) / 360

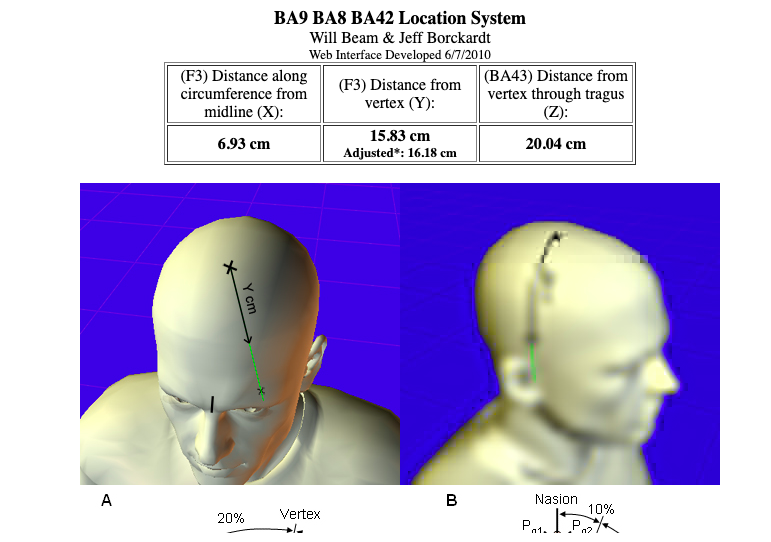
the second is the distance (in cm's) from the vertex along a line intersecting point-x = (F3) Distance from vertex (Y) = r\_corrected \*\* 2 + 0 – 2 \* r\_corrected \* 0 \* cos(0 - r\_corrected) = r\_corrected \*\*2

The distance from the vertex specified by the computer program along a ray beginning at the vertex and intersecting point-x, will be the F3 location from the 10–20 system. = (BA43) Distance from vertex through tragus (Z) = (R2/2),0 = (R2/2) \*\* 2

Class

Init takes circum\_point\_x, vertex\_point\_x, tragus\_vertex

Set\_circum\_point\_x(



Point (0,0) is the midline point