Computing PCO components.

In accordance with the document "PCC definition" PCO components (e,n,u) are the solution of equation (3 pd):

$$\begin{pmatrix} \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\sin\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\cos\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\cos\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\; w\sin\theta d\theta d\varphi \end{pmatrix} = N\begin{pmatrix} e \\ n \\ u \\ \rho \end{pmatrix} \qquad (3\text{ pd})$$

For three types of weight function given in the following table

Weight function	enum OffsetMode
$w=\cos(\theta)$	eSinAndCos
$w=1$	eOnlyCos
$w=1/\sin(\theta)$	eNoWeight

Where θ_0 is the zenith angle elevation mask and matrix N is given by equation (4 pd):

$$w(\theta)=\cos\theta: \qquad N=\begin{pmatrix} \pi\sin^4\theta_0/4 & 0 & 0 & 0 \\ 0 & \pi\sin^4\theta_0/4 & 0 & 0 \\ 0 & 0 & \pi\left(1-\cos^4\theta_0\right)/2 & 2\pi\left(1-\cos^3\theta_0\right)/3 \\ 0 & 0 & 2\pi\left(1-\cos^3\theta_0\right)/3 & \pi\sin^2\theta_0 \end{pmatrix}$$

$$w(\theta)=1: \; N=\begin{pmatrix} \frac{\pi(\cos\theta_0-1)^2(\cos\theta_0+2)}{3} & 0 & 0 & 0 \\ 0 & \frac{\pi(\cos\theta_0-1)^2(\cos\theta_0+2)}{3} & 0 & 0 \\ 0 & 0 & \frac{2\pi\left(1-\cos^3\theta_0\right)}{3} & \pi\sin^2\theta_0 \\ 0 & 0 & \pi\sin^2\theta_0 & 2\pi\left(1-\cos\theta_0\right) \end{pmatrix}$$

$$w(\theta)=1/\sin(\theta): \; N=\begin{pmatrix} \frac{\pi(2\theta_0-\sin 2\theta_0)}{4} & 0 & 0 & 0 \\ 0 & \frac{\pi(2\theta_0-\sin 2\theta_0)}{4} & 0 & 0 \\ 0 & 0 & \frac{\pi(2\theta_0+\sin 2\theta_0)}{2} & 2\pi\sin\theta_0 \\ 0 & 0 & 2\pi\sin\theta_0 & 2\pi\theta_0 \end{pmatrix} \qquad (4\text{ pd})$$

So, to solve (3 pd) we have to compute integrals in the left part of it:

$$\begin{pmatrix} \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\sin\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\cos\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\cos\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\; w\sin\theta d\theta d\varphi \end{pmatrix} \tag{1}$$

$PCC(\theta,\varphi)$ in equation (1) is loaded from ANTEX file and consists of two components:

$$\begin{aligned} PCC(\theta,\varphi) &= PCO_a(\theta,\varphi) + PCV_a(\theta,\varphi) \\ &= e_a\sin\varphi\cos\theta + n_a\cos\varphi\cos\theta + u_a\sin\theta + PCV_a(\theta,\varphi) \end{aligned} \tag{2}$$

Where (e_a, n_a, u_a) are PCO components which are given in ANTEX file. So, integrals (1) can be split into two components also:

$$\begin{pmatrix} \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\sin\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\cos\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\cos\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCC(\theta,\varphi)\; w\sin\theta d\theta d\varphi \end{pmatrix} =$$

$$\begin{pmatrix} \int_0^{2\pi}\int_0^{\theta_0} PCO_a(\theta,\varphi)\sin\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCO_a(\theta,\varphi)\cos\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCO_a(\theta,\varphi)\cos\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCO_a(\theta,\varphi)\; w\sin\theta d\theta d\varphi \end{pmatrix} + \begin{pmatrix} \int_0^{2\pi}\int_0^{\theta_0} PCV_a(\theta,\varphi)\sin\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCV_a(\theta,\varphi)\cos\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCV_a(\theta,\varphi)\cos\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCV_a(\theta,\varphi)\; w\sin\theta d\theta d\varphi \end{pmatrix} \tag{3}$$

First component is computed analytically:

$$\begin{pmatrix} \int_0^{2\pi}\int_0^{\theta_0} PCO_a(\theta,\varphi)\sin\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCO_a(\theta,\varphi)\cos\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCO_a(\theta,\varphi)\cos\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCO_a(\theta,\varphi)\, w\sin\theta d\theta d\varphi \end{pmatrix} = N \begin{pmatrix} e_a \\ n_a \\ u_a \\ 0 \end{pmatrix} \tag{4}$$

Where N is defined by (4 pd), that is it is the same as in equation (3 pd). Using this we can rewrite the equation (3 pd) as follows:

$$\begin{pmatrix} \int_0^{2\pi}\int_0^{\theta_0} PCV_a(\theta,\varphi)\sin\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCV_a(\theta,\varphi)\cos\varphi\sin\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCV_a(\theta,\varphi)\cos\theta\, w\sin\theta d\theta d\varphi \\ \int_0^{2\pi}\int_0^{\theta_0} PCV_a(\theta,\varphi)\, w\sin\theta d\theta d\varphi \end{pmatrix} = N \begin{pmatrix} e-e_a \\ n-n_a \\ u-u_a \\ \rho \end{pmatrix} \tag{5}$$

In contrast to (3 pd) in (5) integrals are taken not from PCC but from PCV and the resulting PCO components are differences between sought for ones and taken from ANTEX file. These integrals can be computed numerically, but as they are two-dimensional, it is rather time consuming, so we do the following. As $PCV_a(\theta,\varphi)$ are given on the grid points (usually with 5-degree steps), we first compute integral by θ considering that PCV is linearly interpolated between given points. First two upper integrals by θ in (5) are equal to each other, we denote them as $ien(\varphi,\theta_0)$, two lower integrals in (5) we denote as $iu(\varphi,\theta_0)$ and $i\rho(\varphi,\theta_0)$. Taking into account linear interpolation of PCV between grid points these integrals read as follows:

$$\begin{aligned} ien(\varphi,\theta_0) &= \int_0^{\theta_0} PCV_a(\theta,\varphi)\sin\theta w\sin\theta d\theta = \sum_{i=1}^{M}\int_{\theta_i}^{\theta_{i+1}} PCV_i(\theta,\varphi)\sin\theta w\sin\theta d\theta \\ iu(\varphi,\theta_0) &= \int_0^{\theta_0} PCV_a(\theta,\varphi)\cos\theta w\sin\theta d\theta = \sum_{i=1}^{M}\int_{\theta_i}^{\theta_{i+1}} PCV_i(\theta,\varphi)\cos\theta w\sin\theta d\theta \\ i\rho(\varphi,\theta_0) &= \int_0^{\theta_0} PCV_a(\theta,\varphi) w\sin\theta d\theta = \sum_{i=1}^{M}\int_{\theta_i}^{\theta_{i+1}} PCV_i(\theta,\varphi) w\sin\theta d\theta \\ where\ \ PCV_i(\theta,\varphi) &= \left(\frac{PCV_a(\theta_{i+1},\varphi)-PCV_a(\theta_i,\varphi)}{\theta_{i+1}-\theta_i}(\theta-\theta_i)+PCV_a(\theta_i,\varphi)\right) \end{aligned} \tag{6}$$

So we split the integration area into M intervals (for an example for $\theta_0 = 90\deg$ and angle step of 5 degrees, $M = 18$) and on each of these intervals $PCV_i(\theta,\varphi)$ is a linear function and all integrals in (6) are computed analytically. The result of the integration depends on the weight function. In the AntexAntenna::calcOffset() method I use lambda-functions `intCosSinEN`, `intSinEN`, `intNoneEN`, `intCosSinU`, `intSinU`, `intNoneU`, `intCosSinRo`, `intSinRo`, `intNoneRo` to compute them. These lambda-functions take as parameters zenith angles θ_i, θ_{i+1} (variables t0 and t1) and corresponding PCV values $PCV(\theta_i, \varphi), PCV(\theta_{i+1}, \varphi)$ (variables p0 and p1) and return the value of corresponding integral. The whole integral by $\int_0^{\theta_0} ...d\theta$ is the sum of M such integrals and it is computed by the lambda-function `intTheta`. As one can see from the C++ code this function takes as arguments the value of azimuth angle φ for which the integral by θ is computed (variable az), the step for zenith angle θ which is usually 5 degrees (variable dt), output array intEle for 4 computed integrals from (6) and elevation mask eleMaskDeg which is related to the mask by zenith angle by the expression zenMaskDeg = 90 – eleMaskDeg.

To move further in calculating integrals from (5) we just need to calculate the integrals over the azimuth angle. We compute them numerically, here is the corresponding piece of code from calcOffset:

```
double intAz[4] = { 0, 0, 0, 0 };
double dt = zenStep * PI / 180;
double dfi = azStep * PI / 180;

double sq3 = sqrt(3);
for (double az = 0; az < 360 - azStep / 2; az += azStep)
{
	double az0 = az + azStep / 2 - azStep / (sq3 * 2);
	double az1 = az + azStep / 2 + azStep / (sq3 * 2);
	double ca0 = cos(az0 * PI / 180);
	double sa0 = sin(az0 * PI / 180);
	double ca1 = cos(az1 * PI / 180);
	double sa1 = sin(az1 * PI / 180);

	double intEle0[4] = { 0, 0, 0, 0 };
	double intEle1[4] = { 0, 0, 0, 0 };
	intTheta(az0, dt, intEle0, eleMask);
	intTheta(az1, dt, intEle1, eleMask);

	intAz[0] += (intEle0[0] * sa0 + intEle1[0] * sa1) / 2;
	intAz[1] += (intEle0[1] * ca0 + intEle1[1] * ca1) / 2;
	intAz[2] += (intEle0[2] + intEle1[2]) / 2;
	intAz[3] += (intEle0[3] + intEle1[3]) / 2;
}

for (int i = 0; i < 4; i++)
	intAz[i] *= dfi;
```

Here we use Gauss quadrature 2-points formula (see, f.e. https://en.wikipedia.org/wiki/Gaussian_quadrature).

The rest of the code of calcOffset is the solution of linear equation (5) and restoring the sought for PCO using PCO from ANTEX file.

One more issue I have to note. The method calcOffset takes elevation mask instead of zenith angle mask for historical reasons. I always used the elevation mask before, because of that it seems more suitable to me and does not rise possible compatibility difficulties. Of course, it may be changed to zenith angle mask, using formula above.