# Advanced stuff

VNiVERSiDAD Đ SALAMANCA

Unidad de Excelencia en

Luz y Materia Estructuradas (LUMES)

ALF USAL

# Representing the field polarization

$$\mathbf{E}(\mathbf{r},t) = \left[U_x(x,y)\mathbf{e}_x + U_y(x,y)\mathbf{e}_y\right] e^{i(kx-\omega t)}$$

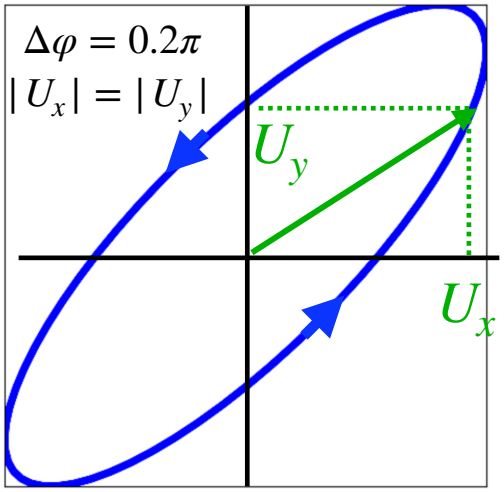
$$U_x = |U_x| e^{i\varphi_x}$$

$$U_y = |U_y| e^{i\varphi_y}$$

**We define the phase difference as** $\Delta\varphi = \varphi_y - \varphi_x$

**left polarizarion** $\Delta\varphi < 0$

**right polarizarion** $\Delta\varphi > 0$



# Representing the field polarization

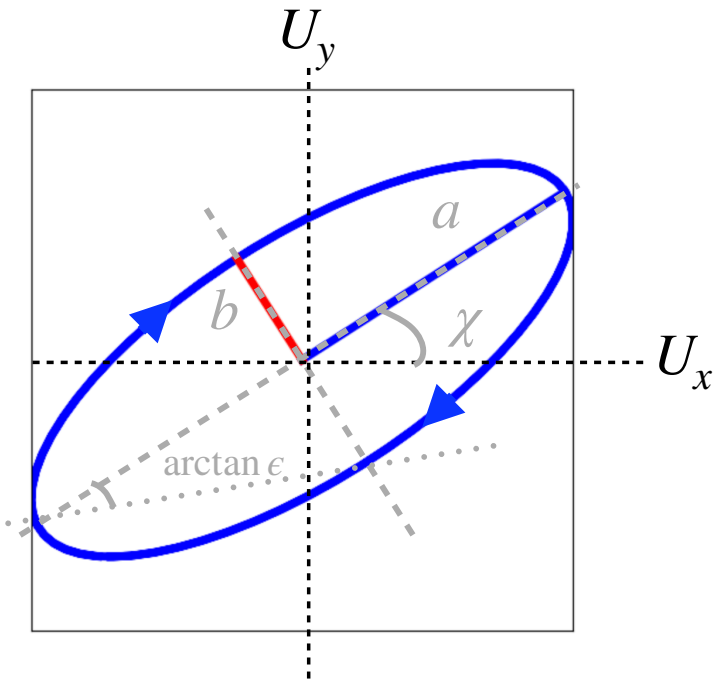
## The polarization ellipse

- Tilt angle

$$\chi = \frac{1}{2}\arctan\left(\frac{2|U_x||U_y|\cos\Delta\varphi}{|U_x|^2+|U_y|^2}\right)$$

- Ellipticity

$$\epsilon = \frac{2|U_x||U_y|\sin\Delta\varphi}{|U_x|^2+|U_y|^2}$$



# Python implementation

- Define a subgrid (coarse grid)

```
N = 11
# For each axis, pick indices that split the range evenly
ny, nx = X.shape
row_idx = np.linspace(0, ny - 1, N, dtype=int)
col_idx = np.linspace(0, nx - 1, N, dtype=int)
X_coarse = X[np.ix_(row_idx, col_idx)]
Y_coarse = Y[np.ix_(row_idx, col_idx)]
```

- Define maximum size of ellipses (not to everflow the coarse grid)

```
dx =  X_coarse[0, 1] -  X_coarse[0, 0]
dy =  Y_coarse[1, 0] -  Y_coarse[0, 0]
#dx=dx/XYratio

ellipse_width = 0.45 * dx
ellipse_height = 0.45 * dy
```

- Draw the ellipses at each point of the coarse grid

```
for i in range(N):
    for j in range(N):
        cx, cy = X_coarse[i, j], Y_coarse[i, j]
        angle = orientation_coarse[i, j]
        major = ellipse_width
        minor = ellipse_height * abs(ellipticity_coarse[i, j])
        if np.abs( ellipticity_coarse[i, j])<0.05:
            edgecolor='white'
        elif np.sign(ellipticity_coarse[i, j])==1:
            edgecolor='black'
        else:
            edgecolor='white'

        ell = Ellipse((cx, cy),
                      width=major,
                      height=minor,
                      angle=angle/np.pi*180,
                      edgecolor=edgecolor,
                      facecolor='none',
                      lw=0.5)
        ax.add_patch(ell)
```