How SHADOW manipulates electric vectors

M. Sanchez del Rio

March 15, 2013

Abstract

This document explains how the electric fields and related parameters (intensity, polarization) are defined and modified in SHADOW. It also describes a bug fixed in March 2013.

1 Definitions

Most Monte-Carlo codes store a parameter "intensity" or weight for each ray that allow to track the energy or intensity of each traced event. As most of electromagnetic interactions are better described by the change in the electric fieldit is more appropriated to store directly the electric fields and not the intensity. The intensity is the squarred-modulus of the electric fields, The electric fields permit not only to compute intensity or each ray, but also the polarization, phase changes, and coherence properties. In order to extract usable and accurate information on these important parameters, the electric fields must be correctly simulated and propagated. I try to summarize here some of the assumptions and manipulations of the electric fields in SHADOW.

A single ray $\vec{R}$ in SHADOW is an array of 18 parameters (called "columns" in SHADOW's jargon) stored in the following order:

$$\vec{R} = (\vec{x}, \vec{v}, \vec{E}_\sigma, f, k, n, opd, \phi_\sigma, \phi_\pi, \vec{E}_\pi) \tag{1}$$

where $\vec{x}$ are the spatial coordinates, $\vec{v}$ the director cosines (ray direction), f is a flag (good or lost ray), k is the wavevector modulus in cm^{-1} ($k = 2\pi/\lambda$, with λ the photon wavelength), n is a counter (from one to the total number of rays), *opd* is the optical path (the travelled distance times the refraction index). The complex electric vector is expressed as a function of the parallel (σ) and perpendicular (π) components:

$$\vec{E} = \vec{E}_\sigma e^{i\phi_\sigma} + \vec{E}_\pi e^{i\phi_\pi} \tag{2}$$

.

Obviously the storage in SHADOW is redundant, for example $v_3 = \sqrt{v_1^2 + v_2^2}$ and regarding the electric fields:

- $\vec{E}_\sigma \perp \vec{E}_\pi$, $\vec{E}_\sigma \perp \vec{v}$, $\vec{E}_\pi \perp \vec{v}$
- $\vec{E}_\sigma$ and $\vec{E}_\pi$ are real.
- $|\vec{E}|^2 = 1$ at the source, and $|\vec{E}|$ is reduced as far as the ray interacts with the different optical elements.

The intensity of the ray is:

$$I = |\vec{E}|^2 = \vec{E}.\vec{E}^* = (\vec{E}_\sigma e^{i\phi_\sigma} + \vec{E}_\pi e^{i\phi_\pi}).(\vec{E}_\sigma e^{-i\phi_\sigma} + \vec{E}_\pi e^{-i\phi_\pi}) = \\ |\vec{E}_\sigma|^2 + |\vec{E}_\pi|^2 + 2\vec{E}_\sigma.\vec{E}_\pi \cos(\phi_\sigma - \phi_\pi) = |\vec{E}_\sigma|^2 + |\vec{E}_\pi|^2 \quad (3)$$

.

2 A polarized source

When simulating a source, SHADOW first samples the ray energy, direction and positions and then define set electric vectors. A first step is to obtain orthogonal unitary vectors $\vec{u}_\sigma$ and $\vec{u}_\pi$ normal to $\vec{v}$. At the source level, there is not yet knowledge of any optical surface to define σ and π directions, so these directions are considered "mostly" horizontal and vertical, respectively. The unitary vectors are calculated as:

$$\vec{u}_\sigma = \vec{v} \wedge (\vec{u}_1 \wedge v) \\ \vec{u}_\pi = \vec{u}_\sigma \wedge v \quad (4)$$

where $\vec{u}_1 = (1, 0, 0)$ the unitary vector along the horizontal direction at the source plane. This is done with the following code in the `sourceGeom` subroutine of the `shadow_kernel.F90` module:

```
A_VEC(1)              =    1.0D0
A_VEC(2)              =    0.0D0
A_VEC(3)              =    0.0D0
! C
! C   Rotate A_VEC so that it will be perpendicular to DIREC and
    with the
! C   right components on the plane.
! C
CALL          CROSS    (A_VEC,DIREC,A_TEMP)
CALL          CROSS    (DIREC,A_TEMP,A_VEC)
CALL          NORM     (A_VEC,A_VEC)
CALL          CROSS    (A_VEC,DIREC,AP_VEC)
CALL          NORM     (AP_VEC,AP_VEC)
```

Then the electric vectors are these versors times the corresponding (σ or π) modulus, in such a way that the intensity is one. The mudulus of each component is a function of a "particular" polarization degree defined as:

$$P = \frac{\cos\chi}{\cos\chi + \sin\chi} \quad (5)$$

where χ is the angle between the $\vec{E}$ vector and $\vec{u}_1$ when the beam is linearly polarized. The electric vector components are:

$$\vec{E}_\sigma = \frac{P}{\sqrt{1 - 2P + 2P^2}}\vec{u}_\sigma \\ \vec{E}_\pi = \frac{1 - P}{\sqrt{1 - 2P + 2P^2}}\vec{u}_\pi. \quad (6)$$

The phase ϕ_σ is set to zero for coherent beams or to a random angle in $[0, 180°)$ for an incoherent beam, and $\phi_\pi = \phi_\sigma + \Phi$ with:

- $\Phi = 0$ for linearly polarizad light

- $\Phi = 90°$ for right (CW) elliptical polarization
- $\Phi = -90°$ for left (CCW) elliptical polarization.

A few examples:

- $P = 1$,$\Phi = 0$ for horizontal linearly polarizad light
- $P = 0$,$\Phi = 0$ for vertical linearly polarizad light
- $P = 0.5$,$\Phi = 0$ for $45°$ linearly polarizad light
- $P = 0.5$ $\Phi = 90°$ for right circular polarization
- $P < 0.5$,$\Phi = -90°$ for left elliptical polarization mostly horizontal

3 Change of reference frame

The beam created in the source reference frame must be converted into the optical element reference frame for performing the ray tracing. The rotation of the vectors from the source (or *previous* o.e. to the o.e. under study is done in the subroutine `RESTART`. It applies sequential rotations of the vectors using the mirror orientation angle α, incident angle θ_i, etc. An example of code is the following, for changing the $\vec{E}_\pi$:

```
TEMP_1P(1) =   AP(1,ITIME)*COSAL + AP(3,ITIME)*SINAL
TEMP_1P(2) =   AP(2,ITIME)
TEMP_1P(3) = - AP(1,ITIME)*SINAL + AP(3,ITIME)*COSAL

TEMP_2P(1) =   TEMP_1P(1)
TEMP_2P(2) =   TEMP_1P(2)*SINTHR + TEMP_1P(3)*COSTHR
TEMP_2P(3) = - TEMP_1P(2)*COSTHR + TEMP_1P(3)*SINTHR

TEMP_1P(1) =   TEMP_2P(1)*COSAL_S - TEMP_2P(2)*SINAL_S
TEMP_1P(2) =   TEMP_2P(1)*SINAL_S + TEMP_2P(2)*COSAL_S
TEMP_1P(3) =   TEMP_2P(3)

AP(1,ITIME)      = TEMP_1P(1)
AP(2,ITIME)      = TEMP_1P(2)
AP(3,ITIME)      = TEMP_1P(3)
```

Then the subroutine `MIRROR1` performs the reflection/refraction/diffraction by the optical element (o.e.), changing the beam direction and electric vectors. This will be discussed in detail in the next paragraphs. Last, after the reflection, the rays are projected into the image plane. For that it projects the vectors vectors onto the versors of the new reference. For example, for $\vec{E}_\pi$

```
AP_VEC(1) = AP(1,J)
AP_VEC(2) = AP(2,J)
AP_VEC(3) = AP(3,J)
CALL DOT (AP_VEC,UXIM,A_1)
CALL DOT (AP_VEC,VNIMAG,A_2)
CALL DOT (AP_VEC,VZIM,A_3)
AP(1,J) = A_1 * rr_reflectivity
AP(2,J) = A_2 * rr_reflectivity
AP(3,J) = A_3 * rr_reflectivity
```

It is important that these changes of refecence system (before and after interaction with the o.e.) are not affecting the phase ϕ_σ and ϕ_π and conserve

the modulus of each electric field component $\vec{E}_\sigma$ and $\vec{E}_\pi$. Also, they do not alter the orthogonality relationships:

$$\begin{aligned} \vec{E}_\sigma &\perp \vec{E}_\pi \\ \vec{E}_\sigma &\perp \vec{v} \\ \vec{E}_\pi &\perp \vec{v} \end{aligned} \tag{7}$$

4 Modifications of electric fields in "local" o.e.

The physics of the X-ray reflection and diffraction in an optical surface affects in a different way the *local* parallel (σ) and perpendicular (π) components. The "local" parallel component is a vector that sits on the o.e. surface and it is not coincident with the incident $\vec{E}_\sigma$. Therefore, one must calculate the "local" $\vec{u'}_\sigma$ and $\vec{u'}_\pi$ vectors that verify 1) they are orthogonal to $\vec{v}$ and ii) $\vec{u'}_\sigma$ is contained in the o.e. surface. This is done here (`MIRROR1` subroutine):

```
! * Check for reflectivity. If this mode is "on", we have to compute
! * some angles, namely the sine of the incidence angle and the sine
! * of the A vector with the normal. Also, the polarized light is
! * treated as a superposition of two orthogonal A vectors with the
    appropriate
! * phase relation. These two incoming vectors have to be resolved into
    the
! * local S- and P- component with a new phase relation.
! * A_VEC will be rotated later, once the amplitude will have been
    determined.
      CALL     CROSS_M_FLAG     (VVIN,VNOR,AS_TEMP,M_FLAG)      ! vector
          pp. to inc.pl.
!print *,'M_FLAG: ',M_FLAG

      IF (M_FLAG.EQ.1) THEN
      CALL     DOT      (AS_VEC,AS_VEC,AS2)
      CALL     DOT      (AP_VEC,AP_VEC,AP2)
      IF (AS2.NE.0)   THEN
       DO 499 I=1,3
499      AS_TEMP(I) = AS_VEC(I)
      ELSE
       DO 599 I=1,3
599      AS_TEMP(I) = AP_VEC(I)
      END IF
      END IF


      CALL     NORM     (AS_TEMP,AS_TEMP)        ! Local unit As vector
      CALL     CROSS    (AS_TEMP,VVIN,AP_TEMP)
      CALL     NORM     (AP_TEMP,AP_TEMP)        ! Local unit Ap vector
```

The electric vector can thus be expressed in two orthonormal vectors $\vec{E'}_\sigma$ and $\vec{E'}_\pi$ along these new directions $\vec{u'}_\sigma$ and $\vec{u'}_\pi$. Physically, it is a rotation of the old electric vectors around the $\vec{v}$ direction to put the σ component on top of the surface, but the rotation affect not only the moduli, but also the phases. The new (complex) electric vectors are build from the projection of the old ones onto the new axes, thus originating a transformation in both moduli and phases:

$$\vec{E'}_\sigma e^{i\phi'_\sigma} = [(\vec{E}_\sigma e^{i\phi_\sigma}).\vec{u'}_\sigma + (\vec{E}_\pi e^{i\phi_\pi}).\vec{u'}_\sigma] \;\; \vec{u'}_\sigma \tag{8}$$

$$\vec{E'}_\pi e^{i\phi'_\pi} = [(\vec{E}_\sigma e^{i\phi_\sigma}).\vec{u'}_\pi + (\vec{E}_\pi e^{i\phi_\pi}).\vec{u'}_\pi] \;\; \vec{u'}_\pi \tag{9}$$

Defining the projection of the (real) electric field components onto these versors as:

$$\begin{aligned} a_{11} &= \vec{E}_\sigma.\vec{u'}_\sigma, \quad & a_{12} &= \vec{E}_\sigma.\vec{u'}_\pi \\ a_{21} &= \vec{E}_\pi.\vec{u'}_\sigma, \quad & a_{22} &= \vec{E}_\pi.\vec{u'}_\pi \end{aligned} \tag{10}$$

we get:

$$\begin{aligned} \vec{E'}_\sigma e^{i\phi'_\sigma} &= (a_{11}e^{i\phi_\sigma} + a_{12}e^{i\phi_\pi})\vec{u'}_\sigma \\ \vec{E'}_\pi e^{i\phi'_\pi} &= (a_{21}e^{i\phi_\sigma} + a_{22}e^{i\phi_\pi})\vec{u'}_\pi \end{aligned} \tag{11}$$

And the moduli are:

$$\begin{aligned} |\vec{E'}_\sigma|^2 = \vec{E'}_\sigma.\vec{E'}^*_\sigma &= a_{11}^2 + a_{12}^2 + 2a_{11}a_{12}\cos(\phi_\sigma - \phi_\pi) \equiv M_\sigma^2 \\ |\vec{E'}_\pi|^2 = \vec{E'}_\pi.\vec{E'}^*_\pi &= a_{21}^2 + a_{22}^2 + 2a_{21}a_{22}\cos(\phi_\sigma - \phi_\pi) \equiv M_\pi^2 \end{aligned} \tag{12}$$

Therefore we can construct the new local (real) electric fields as:

$$\begin{aligned} \vec{E'}_\sigma &= M_\sigma \vec{u'}_\sigma \\ \vec{E'}_\pi &= M_\pi \vec{u'}_\pi \end{aligned} \tag{13}$$

To compute the new phases, we insert Eq. 13 in Eq. 11 and get:

$$\begin{aligned} e^{i\phi'_\sigma} &= M_\sigma^{-1}(a_{11}e^{i\phi_\sigma} + a_{12}e^{i\phi_\pi}) \\ e^{i\phi'_\pi} &= M_\pi^{-1}(a_{21}e^{i\phi_\sigma} + a_{22}e^{i\phi_\pi}) \end{aligned} \tag{14}$$

Therefore:

$$\begin{aligned} \tan\phi'_\sigma &= \frac{Im(a_{11}e^{i\phi_\sigma} + a_{12}e^{i\phi_\pi})}{Re(a_{11}e^{i\phi_\sigma} + a_{12}e^{i\phi_\pi})} = \frac{a_{11}\sin\phi_\sigma + a_{12}\sin\phi_\pi}{a_{11}\cos\phi_\sigma + a_{12}\cos\phi_\pi} \\ \tan\phi'_\pi &= \frac{Im(a_{21}e^{i\phi_\sigma} + a_{22}e^{i\phi_\pi})}{Re(a_{21}e^{i\phi_\sigma} + a_{22}e^{i\phi_\pi})} = \frac{a_{21}\sin\phi_\sigma + a_{22}\sin\phi_\pi}{a_{21}\cos\phi_\sigma + a_{22}\cos\phi_\pi} \end{aligned} \tag{15}$$

The code that implements Eq. 13 and Eq. 15 is in `MIRROR1`:

```
CALL    DOT       (AS_VEC,AS_TEMP,A11)
CALL    DOT       (AP_VEC,AS_TEMP,A12)
CALL    DOT       (AS_VEC,AP_TEMP,A21)
CALL    DOT       (AP_VEC,AP_TEMP,A22)
! ** Now recompute the ampltitude and phase of the local S- and P-
    component.
AS_NEW  = SQRT(ABS(A11**2 + A12**2 + 2.0D0*A11*A12*COS(PHS-PHP)))
AP_NEW  = SQRT(ABS(A21**2 + A22**2 + 2.0D0*A21*A22*COS(PHS-PHP)))
CALL DOT(AP_VEC,AS_VEC,TMP11)

CALL    SCALAR  (AS_TEMP,AS_NEW,AS_VEC) ! Local As vector
CALL    SCALAR  (AP_TEMP,AP_NEW,AP_VEC) ! Local Ap vector

PHTS    = A11*SIN(PHS) + A12*SIN(PHP)
PHBS    = A11*COS(PHS) + A12*COS(PHP)
PHTP    = A21*SIN(PHS) + A22*SIN(PHP)
PHBP    = A21*COS(PHS) + A22*COS(PHP)
CALL    ATAN_2  (PHTS,PHBS,PHS)          ! Phase of local As vector
CALL    ATAN_2  (PHTP,PHBP,PHP)          ! Phase of local Ap vector
```

Note that these transformations imply a polarization mixing, or not conservation of the intensity of each component during the projection ($|\vec{E'}_\sigma| \neq |\vec{E}_\sigma|$), but the total intensity is conserved ($I = |\vec{E}_\sigma|^2 + |\vec{E}_\pi|^2 = |\vec{E'}_\sigma|^2 + |\vec{E'}_\pi|^2$). The phases also change, but the difference of thase is conserved $\Phi = \phi_\sigma - \pi_\pi = \phi'_\sigma - \phi'_\pi$ (I think, but I have not demostrated it). The new electric vectors are orthogonal to $\vec{v}$ by constructions.

5 Reflection/refraction in the o.e. and subsequent changes in electric vectors

Once the electric vectors are expressed in the o.e. local coordinates ($\vec{E'}_\sigma$ and $\vec{E'}_\pi$) they are multiplied by the mirror/crystal reflectivities and the phases also affected: :

$$\begin{aligned}
\vec{E'}_\sigma^{new} &= \vec{E'}_\sigma R_\sigma \\
\vec{E'}_\pi^{new} &= \vec{E'}_\pi R_\pi \\
\phi'_\sigma^{new} &= \phi'_\sigma \Sigma_\sigma \\
\phi'_\pi^{new} &= \phi'_\pi \Sigma_\pi
\end{aligned} \tag{16}$$

where R_σ and R_π are the o.e. reflectivities and Σ_σ and Σ_π are the phases added in the reflection/refraction/diffraction. This process is accompanied by a change of direction of the ray from the incident direction $\vec{v}$ to a new output direction $\vec{v'}$. All these operations are done in `MIRROR1` subroutine.

The resulting electric vectors and phases after applying Eq. 16 must now hold the orthogonality relations (Eq. 7) with respect to $\vec{v'}$ so they must be changed. This part was incorrectly done in the following code, which assumes conservation of the $\vec{E'}_\sigma$ component (except for gratings) and the π component is "mirrored". These operations do not garantee that the resulting vectors are normal to $\vec{v'}$. In some cases, in particular for σ polarizad light onto o.e.'s with mirror orientation angle 90° can result in a loss of beam intensity in subsequent changes of frames. The **wrong** code was (in `MIRROR1`):

```
! ** So far we have the new amplitude of the two components. We have now
! ** to 'reflect' A_VEC onto the mirror. For this, notice that the s-
    comp
! ** is geometrically unchanged, while the p-comp is changed. The angles
! ** are exchanged with respect to VVIN. Things are more complicated in
! ** the case of a grating, due to the vectorial nature of the
    diffraction,
! ** not treated here. We make the simplifying assumption that the
! ** diffraction will not change the degree of polarization. This mean
    that
! ** A_VEC will have the same components referred to the ray as before
    the
! ** diffraction.

VVOUT(1)          =    RAY(4,ITIK)
VVOUT(2)          =    RAY(5,ITIK)
VVOUT(3)          =    RAY(6,ITIK)
! C
! C The following IF block applies only to the GRATING case.
! C The binormal is redefined in terms of the diffraction
! C plane.
! C
        IF (F_GRATING.NE.0.OR.F_BRAGG_A.EQ.1) THEN
          CALL  PROJ       (VVOUT,VNOR,VTEMP)
          CALL  SCALAR     (VTEMP,-2.0D0,VTEMP)
          CALL  SUM        (VTEMP,VVOUT,VTEMP)
          !CALL CROSS      (VTEMP,VNOR,AS_TEMP)
          CALL  CROSS_M_FLAG      (VTEMP,VNOR,AS_TEMP,M_FLAG)
         IF (M_FLAG.EQ.1) THEN
          CALL DOT         (AS_VEC,AS_VEC,AS2)
          CALL DOT         (AP_VEC,AP_VEC,AP2)
         IF (AS2.NE.0) THEN
           DO 899 I=1,3
899           AS_TEMP(I) = AS_VEC(I)
         ELSE
          DO 999 I=1,3
```

```
 999          AS_TEMP(I) = AP_TEMP(I)
            END IF
           END IF
            CALL  NORM      (AS_TEMP,AS_TEMP)          ! Local unit As vector
            CALL  CROSS     (AS_TEMP,VTEMP,AP_TEMP)
            CALL  NORM      (AP_TEMP,AP_TEMP)          ! Local unit Ap vector
            CALL  DOT       (AS_VEC,AS_VEC,RES)
            RES   =     SQRT (RES)

            CALL  SCALAR    (AS_TEMP,RES,AS_VEC)
            CALL  DOT       (AP_VEC,AP_VEC,RES)
            RES   =     SQRT (RES)
            CALL  SCALAR    (AP_TEMP,RES,AP_VEC)
          END IF


CALL      PROJ      (AP_VEC,VNOR,VTEMP)
CALL      VECTOR    (VTEMP,AP_VEC,VTEMP)
CALL      SCALAR    (VTEMP,-2.0D0,VTEMP)

CALL      SUM       (AP_VEC,VTEMP,AP_VEC)
```

This code has been replaced by a new one that simply calculates the new σ and π versors orthogonal to $\vec{v'}$ and affects them by the electric field moduli that do not change. Also, the phases are not changed. This guaratees that the intensity is conserved, as well as orthogonality. The new code is:

```
! Electric vectors are changed to assure orthogonality with the new
     direction VVOUT
! To conserve intensity, the moduli of Es and Ep must not change
! AS_VEC and VVOUT are not orthogonal so a projection of S and P
     coordinates into the
! new ones do not work as it may be a component of the electric field
     along VVOUT

CALL CROSS_M_FLAG   (VVOUT,VNOR,AS_TEMP,M_FLAG) ! vector pp. to inc.pl.
CALL DOT (AS_VEC,AS_VEC,AS2)
CALL DOT (AP_VEC,AP_VEC,AP2)

IF (M_FLAG.EQ.1) THEN
 IF (AS2.NE.0) THEN
   DO I=1,3
     AS_TEMP(I) = AS_VEC(I)
   END DO
 ELSE
  DO I=1,3
    AS_TEMP(I) = AP_VEC(I)
  END DO
 END IF
END IF

CALL NORM    (AS_TEMP,AS_TEMP) ! Local unit As vector perp to vvout
CALL CROSS (AS_TEMP,VVOUT,AP_TEMP)
CALL NORM (AP_TEMP,AP_TEMP) ! Local unit Ap vector perp to vvout

do i=1,3
  as_vec(i) = as_temp(i) * sqrt(as2)
  ap_vec(i) = ap_temp(i) * sqrt(ap2)
end do
```