Models for sputtering yield

Markus Brenneis, Vladislav Kotov, FZJ

October, 2016

Inhaltsverzeichnis

1 1993-model

1. Normal incidence

Equation from [W. Eckstein et al., Atomic and plasma-material interaction data for fusion, vol. 7, Part B (2001)] Chapter 2, first Equation on Page 18:

$$Y(E,0) = Q\left(1-\left(\frac{E_{th}}{E}\right)^{2/3}\right)\left(1-\frac{E_{th}}{E}\right)^{2}\frac{0.5\log\left(1+1.2288\frac{E}{E_{tf}}\right)}{\frac{E}{E_{tf}}+0.1728\sqrt{\frac{E}{E_{tf}}}+0.008\left(\frac{E}{E_{tf}}\right)^{0.1504}} \tag{1}$$

Here E is the kinetic energy of the incident particle, E_{th} is the threshold energy, E_{tf} is the Thomas-Fermi energy, Q is the yield factor. Parameters E_{th}, E_{tf} and Q are defined for each projectile-target combination. For $E < E_{th}$:

$Y(E) = 0$. Same model was applied in subroutine `SPUTER` of the EIRENE code.

2. Angular dependence

Equation from [W. Eckstein et al., Atomic and plasma-material interaction data for fusion, vol. 7, Part B (2001)] Chapter 2, first Equation on Page 19:

$$Y(E,\theta) = Y(E,0) \cdot [\cos(\theta)]^{-f} \exp\left\{ f \left[1 - \frac{1}{\cos\theta} \right] \cos(\theta_{opt}) \right\} \qquad (2)$$

Here $Y(E,0)$ is Equation (1), θ is the incident angle - angle between the incident velocity and external normal to the surface (θ=0 for normal incidence). The values of parameters f and θ_{opt} are fixed at: $f = 2$, θ_{opt}=75^o (that is, $[Y(E,\theta)/Y(E,0)]_{max}$= 3.3902). Same model was applied in subroutine `SPUTER` of the EIRENE code.

2 2001-model

In some places (e.g. web-interface) the model described below which uses fitting coefficients from [W. Eckstein, "Sputtering Yields", Top. Appl. Phys., vol. 110 (2007) p. 33] is called "2007-model". In this case "2001-model" is the model which uses same fit-formulas but fitting coefficients from [W. Eckstein et al., Atomic and plasma-material interaction data for fusion, vol. 7, Part B (2001)]

1. Normal incidence

Equation from [W. Eckstein et al., Atomic and plasma-material interaction data for fusion, vol. 7, Part B (2001)] Chapter 2, last Equation on Page 18:

$$Y(E,0) = 0.5Q \frac{\left(\frac{E}{E_{th}} - 1\right)^{\mu} \ln(1 + 1.2288\epsilon)}{\lambda + \left(\frac{E}{E_{th}} - 1\right)^{\mu} \left[\frac{E}{\epsilon} + 0.1728\sqrt{\frac{E}{\epsilon}} + 0.008 \left(\epsilon \frac{E}{\epsilon}\right)^{0.1504} \right]} \qquad (3)$$

where λ, Q, μ, ϵ, and E_{th} are fitting parameters defined for each target-projectile combination, see [W. Eckstein, "Sputtering Yields", Top. Appl. Phys., vol. 110 (2007) p. 33], Tables 2-9. Note that in this more recent report Equation (3) is written with a mistake. E_{th} is the threshold energy, for $E < E_{th}$: $Y(E) = 0$.

2. Angular dependence

Equation from [W. Eckstein et al., Atomic and plasma-material interaction data for fusion, vol. 7, Part B (2001)] Chapter 2, second Equation on Page 19:

$$Y(E,\theta) = Y(E,0)\left\{\cos\left[\left(\frac{\theta}{\theta_0}\frac{\pi}{2}\right)^c\right]\right\}^{-f}\exp\left\{b\left(1-\frac{1}{\cos\left[\left(\frac{\theta}{\theta_0}\frac{\pi}{2}\right)^c\right]}\right)\right\} \quad (4)$$

Here $Y(E,0)$ is Equation (3), θ is the incident angle - angle between the incident velocity and external normal to the surface (θ=0 for normal incidence), angle θ_0 is calculated as follows:

$$\theta_0 = \pi - \arccos\sqrt{\frac{E_{sp}}{E_{sp}+E_0}} \geq \frac{\pi}{2} \quad (5)$$

E_{sp}, c, f, b are fitting parameters defined for each incident energy, see [W. Eckstein, "Sputtering Yields", Top. Appl. Phys., vol. 110 (2007) p. 33], Tables 10-24.

3 2007-model

Same as "2001-model" with fitting coefficients from [W. Eckstein, "Sputtering Yields", Top. Appl. Phys., vol. 110 (2007) p. 33]. See Section 2 above.

4 Example of usage

1. Makefile

```
SHELL=/bin/sh
FC=ifort -fPIC -O -i4 -r8 -g -traceback -save \
        -fpconstant -implicitnone -I./

run : esputr_example.exe
        ./esputr_example.exe

esputr_example.exe: esputr_example.f
        @echo ESPUTR must be installed in '../bin'
        $(FC) -o $@ esputr_example.f -L../bin -lesputr -I../bin
```

```
clean:
	rm *.o *.exe
```

2. Program

```
C> @file esputr_example.f90
C> Example of usage for ESPUTR
C> @author Vladislav Kotov, v.kotov@fz-juelich.de
      program esputr_example
      use esputr
      use esputr1993
      use esputr2001

      integer,parameter :: N=5,M=6
      integer :: err,itrg,iprj,icomb,i,p,i1st,iLst
      real(esputr_dp) :: E0(N)=(/10,50,100,500,1000/),
     r                   Y1993(N),Y2007(N),Eth1993,Eth2007,
     r                   theta(M)=(/0,40,60,70,80,90/),
     r                   F1993(M),F2007(M),Emin,Emax

C     Initialize the 1993-model
      call esputr1993_init('../data/SPUTER',err)
      if(err.ne.0) STOP "ERROR while initializing the 1993-model"
C     Initialize the 2001-model
      call esputr2001_init('../data/ECKSTEIN2007N',
     i                     '../data/ECKSTEIN2007TH',err)
      if(err.ne.0) STOP "ERROR while initializing the 2001-model"

C
C     Calculate sputtering yield for normal incidence, D on Be
C

C     ... 1993-model
      iprj=esputr1993_getProjectileId('D',err)
      if(err.ne.0) STOP "ERROR: could not find projectile"
      itrg=esputr1993_getTargetId('Beryllium',err)
      if(err.ne.0) STOP "ERROR: could not find target"
      do i=1,N
       Y1993(i)=esputr1993_yn(E0(i),iprj,itrg,err)
       if(err.ne.0)
     s    STOP "ERROR: error while calculating the 1993-yield"
```

```
      end do

C     ... 2001-model (with 2007 data)
      icomb=esputr2001_getProjectileTargetIdN('D','Be',err)
      if(err.ne.0) STOP "ERROR: could not find the "//
     /                  "projectile-target combination"
      do i=1,N
       Y2007(i)=esputr2001_yn(E0(i),icomb,err)
       if(err.ne.0)
     s    STOP "ERROR: error while calculating the 2007-yield"
      end do

C     Get threshold energy
      Eth1993=esputr1993_Eth(iprj,itrg,err)
      if(err.ne.0) STOP "ERROR: cannot read Eth (1993) "
      Eth2007=esputr2001_Eth(icomb,err)
      if(err.ne.0) STOP "ERROR: cannot read Eth (2007) "

C     Print results
      print *
      print *, "E_th (1993) = ",Eth1993," [eV]"
      print *, "E_th (2007) = ",Eth2007," [eV]"
      print *, "E [eV], Y-1993, Y-2007"
      do i=1,N
       print *,E0(i),Y1993(i),Y2007(i)
      end do

C
C     Calculate the angular dependence factor Y(theta,E)/Y(0,E)
C     D on Be, E=100 eV
C
       p=3

C     ... 1993-model
      do i=1,M
C      ESPUTR_PI/180. is to convert degree into radian
       F1993(i)=esputr1993_yth(theta(i)*ESPUTR_PI/180.,err)
      if(err.ne.0)
     f    STOP "ERROR: error while calculating"//
     /         "the 1993 angular dependence"
      end do
```

```
C      ... 2001-model (with 2007-data)
       call esputr2001_getProjectileTargetIdsTH('D','Be',i1st,iLst,err)
       if(err.ne.0) STOP "ERROR: could not find the "//
      /                  "projectile-target combination"

C     Get minimum and maximum enetgy for which the angular
C     dependence data are available
       call esputr2001_getAvailableEnergyRange(i1st,iLst,Emin,Emax,err)
       if(err.ne.0) STOP "ERROR: could not read energy range"
C
C     Use this switch if you want ESPUTR to extrapolate the
C     angular dependence to energies smaller that Emin and
C     larger than Emax. Default value = .false.
C
C     Be careful with this option !!! Do not extrapolate too far !!!
C
       esputr2001_extrapolate_angular=.true.

       do i=1,M
        if(Y2007(p)>0.) then
         F2007(i)=esputr2001_yth(E0(p),theta(i)*ESPUTR_PI/180.,
     ,                           i1st,iLst,err)
         if(err.ne.0) STOP "ERROR: error while calculating"//
      /                    "the 2001 angular dependence"
        else
C        we do not have to calculate F if Y is zero
         F2007(i)=0.
        end if
       end do

C      Print results
       print *
       print *,"Emin = ",Emin," [eV]"
       print *,"Emax = ",Emax," [eV]"
       print *,
      p  "theta [degree], Y(theta)/Y(0)-1993, Y(theta)/Y(0)-2007"
       do i=1,M
        print *, theta(i),F1993(i),F2007(i)
       end do
```

```
      print *

C     Deallocate modules
      call esputr1993_deallocate(err)
      call esputr2001_deallocate(err)

      STOP "ESPUTR_EXAMPLE COMPLETED"

      end program esputr_example
```