

```

1 elseif (seec == 4) then ! New model as of Jul12 2013. FEST3Dish?
2
3 Emax = sey%Emax      ! Emax(delta=max,theta=0) in eV
4 deltamax = sey%deltamax ! Maximum secondary electron yield (at Emax) for normal
   incidence (theta=0)
5 E_0 = sey%E_0
6 E_0p = sey%E_0p
7
8 if (penf1 <= E_0*4) then      ! Reflect electron
9
10    ! Reflect electron inelastically
11    !   pvf = reflect_electron(pcol,penf1*(-e),pvf1,2)
12
13    allocate (thetas(1))
14    allocate (phis(1))
15    allocate (penf(1))
16    allocate (vs(1))
17
18    penf = taus88() * penf1
19    vs = sqrt(2*penf*(-eme))
20
21    ! Determine which is the normal component of the outgoing velocity
22    if (pcol(1)==1 .OR. pcol(3)==1) then
23        i = 1      ! Normal component
24        j = 2      ! Parallel component(not z)
25    elseif (pcol(2)==1 .OR. pcol(4)==1) then
26        i = 2      ! Normal component
27        j = 1      ! Parallel component (not z)
28    endif
29
30    rt = taus88()
31    phis = rt * 2 * pi      ! Calculate emission azimuthal angle of secondaries
32
33    rt = taus88()
34    thetas = asin(2*rt - 1) ! Calculate emission angle of secondaries with respect to
   the normal
35
36    pvf(1,i) = vs(1) * cos(thetas(1)) ! Normal component of secondary velocity
37
38    pvf(1,j) = sin(phis(1)) * vs(1) * sin(thetas(1))
39    pvf(1,3) = cos(phis(1)) * vs(1) * sin(thetas(1))
40
41    if (pcol(1) == 1 .OR. pcol(2) == 1) then
42        pvf(1,i) = - abs(pvf(1,i)) ! Normal velocity pointing into the inside of the
   waveguide
43    elseif (pcol(3) == 1 .OR. pcol(4) == 1) then
44        pvf(1,i) = abs(pvf(1,i)) ! Normal velocity pointing into the inside of the
   waveguide
45    endif
46
47    deallocate (thetas)
48    deallocate (phis)
49    deallocate (penf)
50    deallocate (vs)
51
52 else      ! True secondary(ies)
53
54    ! First calculate yield according to modified Vaughan
55
56    ! Unpack user inputs regarding particle-wall interaction and SEE
57    kse = sey%kse      ! Roughness factor for energy, =0 for rough, =1 for dull, =2

```

```

57 for smooth and anything in between
58 ks = sey%ks ! Roughness factor for angle, =0 for rough, =1 for dull, =2 for
   smooth and anything in between
59
60 ! Determine secondary electron yields (Vaughan's theory (1993))
61 Emax = Emax * (1 + kse*theta*theta/(2*pi))
62 deltamax = deltamax * (1 + ks*theta*theta/(2*pi))
63
64 xi = real((penf1 - E_0))/(Emax - E_0)
65
66 if (xi <= 3.6) then
67
68     if (xi <= 1) then
69         ke = 0.56
70     elseif (xi > 1 .OR. xi <= 3.6) then
71         ke = 0.25
72     endif
73
74     delta = deltamax * ( (xi * exp(1-xi))**ke )
75
76 elseif (xi > 3.6) then
77
78     delta = deltamax * 1.125 / (xi**0.35)
79
80 endif
81
82 if (delta > 10) then
83     write(*,*) 'high delta'
84     call exit
85 endif
86
87 if (isNaN(real(delta,8))) then
88     write(*,*) 'theta = ',theta
89     write(*,*) 'penf1 = ',penf1
90     write(*,*) 'E_0 = ',E_0
91     write(*,*) 'E_0p = ',E_0p
92     write(*,*) 'delta = ',delta
93     write(*,*) 'E_0p = ',E_0p
94     write(*,*) 'E_0 = ',E_0
95 endif
96
97 s = poisdev(real(delta,8),atype)
98
99 if (s > 0) then
100
101     allocate (pvf(s,3))
102     allocate (yn(s))
103     allocate (thetas(s))
104     allocate (phis(s))
105     allocate (penf(s))
106     allocate (vs(s))
107
108     ! Initialize variables
109     pvf = 0. ! m x 3 array of 3D velocities of m secondary electrons
110     yn = 0. ! magnitude of velocity of secondary electrons
111     thetas = 0. ! Emission angle of secondaries with respect to the normal
112     phis = 0. ! Azimuthal emission angle of secondaries
113     penf = 0.
114     vs = 0.
115

```

```

116      ! Calculate cummulative energy of secondary electrons from a Gaussian distribution
117      !      Y = -1.
118      !      do while (Y < 0.)
119      !          Y = penf1 - Eom * abs(random_normal())
120      !      enddo
121      !      Y = sqrt( Y/Eom )      ! v^tilde
122
123      call gratio(real(p_n*s,8), real(penf1/Eom,8), ans, qans, 0)
124      rt = taus88()
125      ans = ans*rt
126      call gaminv(real(p_n*s,8), y, real(0,8), ans, 1-ans, ierr)
127
128      Y = sqrt( Y )      ! v^tilde
129
130      !! Scalar parameters needed to obtain velocity components
131      sint = 1.
132      do k = 1,s-1
133          lnbeta = betaln(real(p_n*(s-k),4),real(p_n,4))
134
135          rt = taus88()
136          inbeta = betain( real(rt,8), real(p_n*(s-k),8), real(p_n,8), lnbeta, ifault )
137          if (ifault /= 0) then
138              write(*,*) 'error calculating incomplete beta function. Terminating.'
139              call exit
140          endif
141
142          alpha = xinbta( real(p_n*(s-k),8), real(p_n,8), lnbeta, inbeta, ifault )
143          if (ifault /= 0) then
144              write(*,*) 'error calculating inverse incomplete beta function. Terminating.'
145              call exit
146          endif
147
148          alpha = asin( sqrt( alpha ) )      ! Alpha angles used to calculate the magnitude of
149          velocity
150          yn(k) = Y * sint * cos(alpha)      ! magnitude of outgoing velocities
151
152          sint = sint * sin(alpha)            ! Spherical coordinates factor
153      enddo
154      yn(s) = Y * sint
155
156      ! Determine which is the normal component of the outgoing velocity
157      if (pcol(1)==1 .OR. pcol(3)==1) then
158          i = 1      ! Normal component
159          j = 2      ! Parallel component(not z)
160      elseif (pcol(2)==1 .OR. pcol(4)==1) then
161          i = 2      ! Normal component
162          j = 1      ! Parallel component (not z)
163      endif
164
165      do k = 1,s
166          penf(k) = Eom * (yn(k))**2
167          vs(k) = sqrt(2*penf(k)*(-eme))
168
169          rt = taus88()
170          phis(k) = rt * 2 * pi      ! Calculate emission azimuthal angle of secondaries
171
172          rt = taus88()
173          thetas(k) = asin(2*rt - 1)      ! Calculate emission angle of secondaries with
174          respect to the normal

```

```
175     pvf(k,i) = vs(k) * cos(thetas(k))    ! Normal component of secondary velocity
176     pvf(k,j) = sin(phis(k)) * vs(k) * sin(thetas(k))
177     pvf(k,3) = cos(phis(k)) * vs(k) * sin(thetas(k))
178     enddo
179
180 !     write(*,*) 's = ',s
181 !     write(*,*) 'penf1 = ', penf1
182 !     write(*,*) 'penf(:) = ', penf
183 !     write(*,*) 'vs(:) = ', vs
184 !     stop
185
186     if (pcol(1) == 1 .OR. pcol(2) == 1) then
187         do k = 1,s
188             pvf(k,i) = - abs(pvf(k,i))    ! Normal velocity pointing into the inside of the
!             waveguide
189         enddo
190     elseif (pcol(3) == 1 .OR. pcol(4) == 1) then
191         do k = 1,s
192             pvf(k,i) = abs(pvf(k,i))    ! Normal velocity pointing into the inside of the
!             waveguide
193         enddo
194     endif
195
196     deallocate (yn)
197     deallocate (thetas)
198     deallocate (phis)
199     deallocate (penf)
200     deallocate (vs)
201
202     elseif (s == 0) then
203         allocate (pvf(1,3))
204         pvf = 0.
205     endif
206
207 endif
208
209 endif
```