

```

1  if (seec == 1) then      !***** Vaughan model *****!
2
3  Emax = sey%Emax          ! Emax(delta=max,theta=0) in eV
4  deltamax = sey%deltamax  ! Maximum secondary electron yield (at Emax) for normal
5  incidence (theta=0)
6  ReRr = sey%ReRr
7  delta_b = sey%delta_b
8  E_0 = sey%E_0
9  E_0p = sey%E_0p
10
11 if (penf1 <= E_0p) then
12     delta = delta_b
13
14 elseif (penf1 > E_0p .AND. penf1 < E_0) then
15
16     delta = -delta_b*(penf1 - E_0)
17
18 else
19
20     ! Unpack user inputs regarding particle-wall interaction and SEE
21     kse = sey%kse          ! Roughness factor for energy, =0 for rough, =1 for dull, =2 for
22     smooth and anything in between
23     ks = sey%ks           ! Roughness factor for angle, =0 for rough, =1 for dull, =2 for
24     smooth and anything in between
25
26     ! Determine secondary electron yields (Vaughan's theory (1993))
27     Emax = Emax * (1 + kse*theta*theta/(2*pi))
28     deltamax = deltamax * (1 + ks*theta*theta/(2*pi))
29
30     xi = real((penf1 - E_0)/(Emax - E_0))
31
32     if (xi <= 3.6) then
33
34         if (xi <= 1) then
35             ke = 0.56
36         elseif (xi > 1 .OR. xi <= 3.6) then
37             ke = 0.25
38         endif
39
40         delta = deltamax * ( (xi * exp(1-xi))**ke )
41
42     elseif (xi > 3.6) then
43
44         delta = deltamax * 1.125 / (xi**0.35)
45
46     endif
47
48 endif
49 !DEBUG
50 if (allocated(SEYvsE)) then
51     call add23Darray(SEYvsE,(/0,0,1/))
52     k = size(SEYvsE,3)
53     SEYvsE(4,1,k) = penf1
54     SEYvsE(4,2,k) = delta
55 else
56     allocate (SEYvsE(4,2,1))
57     SEYvsE(4,1,1) = penf1
58     SEYvsE(4,2,1) = delta
59 endif
60
61 if (delta > 10) then
62     write(*,*) 'high delta'

```

```

61     call exit
62 endif
63
64 !if (first) then
65 !  s = random_Poisson(real(delta,4),.TRUE.)           ! Number of secondary electrons produced
66 !  first = .FALSE.
67 !else
68 !  s = random_Poisson(real(delta,4),.FALSE.)
69 !endif
70
71 if (isNaN(real(delta,8))) then
72   write(*,*) 'theta = ',theta
73   write(*,*) 'penf1 = ',penf1
74   write(*,*) 'E_0 = ',E_0
75   write(*,*) 'E_0p = ',E_0p
76   write(*,*) 'delta = ',delta
77   write(*,*) 'E_0p = ',E_0p
78   write(*,*) 'E_0 = ',E_0
79 endif
80
81 if (atype == 9) then
82   read(17,'(F17.15,1X,I1)') delta0, s
83   if (delta0-delta>1.5e-3) then
84     write(*,*) 'discrepancy in delta'
85     write(*,*) 'penf1 = ',penf1
86     write(*,*) 'theta = ',theta
87     write(*,*) 'delta0 = ',delta0
88     write(*,*) 'delta = ',delta
89     write(*,*) 'diff = ',delta0-delta
90     call exit
91   endif
92 else
93   s = poisdev(real(delta,8),atype)
94 endif
95
96 if (s == 1) then
97
98   allocate (pvf(1,3)) ! Only one electron comes off the wall
99
100  if (ReRr == 1) then
101    ! Determine Re and Rr from de Lara's (2006) empirical fit
102    a_lara = sey%a_lara
103    z_lara = sey%z_lara
104    Eb = 300 + 175*z_lara
105    Rr = a_lara * (1 - (3e-5)*penf1) * (penf1**0.56) * exp(-((penf1/Eb)**0.70))
106
107    Ee1 = 50./sqrt(z_lara)
108    Ee2 = 0.25 * z_lara * z_lara
109    Re = 0.93/(1+penf1/Ee1) + 0.07/(1+penf1/Ee2)
110
111    ! Take into account angle dependence of Re and Rr
112    C1 = 0.89 * Rr / (Re + Rr)
113    Rr = (Rr**cos(theta)) * (C1**(1-cos(theta)))
114    Re = (Re**cos(theta)) * ((0.89-C1)**(1-cos(theta)))
115
116  elseif (ReRr == 2) then      ! Empirical fit from LHC
117    a0 = 20.69989
118    a1 = -7.07605
119    a2 = 0.483547
120    a3 = 0
121    e0 = 56.914686
122    f = exp(a0 + a1*log(penf1+e0) + a2*((log(penf1+e0))**2) + a3*((log(penf1+e0))**3))
123

```

```

124      ! Based on publications saying that 7% and 3% of secondaries are elastically and
125      ! inelastically reflected respectively
126      Re = f*0.3
127      Rr = f-Re
128      endif
129      !open(unit=63,file='ReRr.txt',status='unknown',position='append')
130      !write(63,*) Re,Rr
131      !close(unit=63)
132
133      ! Determine probability of events
134      Pe = Re          ! Prob that electron is elastically reflected
135      Pr = Rr          ! Prob that electron is inelastically reflected
136      Pe = 0.03
137      Pr = 0.07
138
139      ! Now generate a random number uniformly distributed between 0 and 1
140      if (atype == 8) then
141          read(21,*) r
142      elseif (atype == 9) then
143          read(18,*) r
144      else
145          r = taus88()
146      endif
147
148      if (r < Pe) then
149
150          ! Reflect electron elastically
151          pvf = reflect_electron(pcol,penf1*(-e),pvf1,1)
152      !DEBUG
153          if (allocated(SEYvsE)) then
154              call add23Darray(SEYvsE,(/0,0,1/))
155              k = size(SEYvsE,3)
156              SEYvsE(1,1,k) = penf1
157              SEYvsE(1,2,k) = delta
158          else
159              allocate (SEYvsE(4,2,1))
160              SEYvsE(1,1,1) = penf1
161              SEYvsE(1,2,1) = delta
162          endif
163
164      elseif (r >= Pe .AND. r < (Pe+Pr)) then
165
166          ! Electron is reflected inelastically
167          !pvf = reflect_electron(pcol,penf1*(-e),pvf1,2,atype)
168
169          allocate (thetas(1))
170          allocate (phis(1))
171          allocate (penf(1))
172          allocate (vs(1))
173
174          penf = taus88() * penf1
175          vs = sqrt(2*penf*(-eme))
176
177          ! Determine which is the normal component of the outgoing velocity
178          if (pcol(1)==1 .OR. pcol(3)==1) then
179              i = 1          ! Normal component
180              j = 2          ! Parallel component(not z)
181          elseif (pcol(2)==1 .OR. pcol(4)==1) then
182              i = 2          ! Normal component
183              j = 1          ! Parallel component (not z)
184          endif
185

```

```

186     rt = taus88()
187     phis = rt * 2 * pi           ! Calculate emission azimuthal angle of secondaries
188
189     rt = taus88()
190     thetas = asin(2*rt - 1)      ! Calculate emission angle of secondaries with respect to the
normal
191
192     pvf(1,i) = vs(1) * cos(thetas(1))  ! Normal component of secondary velocity
193
194     pvf(1,j) = sin(phis(1)) * vs(1) * sin(thetas(1))
195     pvf(1,3) = cos(phis(1)) * vs(1) * sin(thetas(1))
196
197     if (pcol(1) == 1 .OR. pcol(2) == 1) then
198         pvf(1,i) = - abs(pvf(1,i))      ! Normal velocity pointing into the inside of the
waveguide
199     elseif (pcol(3) == 1 .OR. pcol(4) == 1) then
200         pvf(1,i) = abs(pvf(1,i))      ! Normal velocity pointing into the inside of the waveguide
201     endif
202
203     deallocate (thetas)
204     deallocate (phis)
205     deallocate (penf)
206     deallocate (vs)
207 !DEBUG
208     if (allocated(SEYvsE)) then
209         call add23Darray(SEYvsE,(/0,0,1/))
210         k = size(SEYvsE,3)
211         SEYvsE(2,1,k) = penf1
212         SEYvsE(2,2,k) = delta
213     else
214         allocate (SEYvsE(4,2,1))
215         SEYvsE(2,1,1) = penf1
216         SEYvsE(2,2,1) = delta
217     endif
218
219     else
220 !DEBUG
221     if (allocated(SEYvsE)) then
222         call add23Darray(SEYvsE,(/0,0,1/))
223         k = size(SEYvsE,3)
224         SEYvsE(3,1,k) = penf1
225         SEYvsE(3,2,k) = delta
226     else
227         allocate (SEYvsE(4,2,1))
228         SEYvsE(3,1,1) = penf1
229         SEYvsE(3,2,1) = delta
230     endif
231
232     allocate (yn(1))
233     allocate (thetas(1))
234     allocate (phis(1))
235     allocate (penf(1))
236     allocate (vs(1))
237
238     ! Initialize variables
239     pvf = 0.          ! velocity of 1 secondary electron
240
241     ! Scalar parameters needed to obtain velocity components
242     call gratio(real(p_n*s,8), real(penf1/Eom,8), ans, qans, 0)
243     if (atype == 8) then
244         read(21,*) rt
245         ans = ans*rt
246     elseif (atype == 9) then

```

```

247     read(20,*) rt
248     ans = ans*rt
249   else
250     ans = ans*taus88()
251   endif
252   call gaminv(real(p_n*s,8), yn(1), real(0,8), ans, 1-ans, ierr)
253
254   if (ierr < 0) then
255     write(*,*) 'error in calculating inverse incomplete gamma function. Terminating.'
256     call exit
257   endif
258
259   yn = sqrt( yn ) ! magnitude of normalized velocity of secondary electrons
260
261   penf = Eom * yn * yn
262   vs = sqrt(2*penf*(-eme))
263
264   ! Determine which is the normal component of the outgoing velocity
265   if (pcol(1)==1 .OR. pcol(3)==1) then
266     i = 1 ! Normal component
267     j = 2 ! Parallel component(not z)
268   elseif (pcol(2)==1 .OR. pcol(4)==1) then
269     i = 2 ! Normal component
270     j = 1 ! Parallel component (not z)
271   endif
272
273   if (atype == 8) then
274     read(21,*) rt
275   elseif (atype == 9) then
276     read(20,*) rt
277   else
278     rt = taus88()
279   endif
280   phis = rt * 2 * pi ! Calculate emission azimuthal angle of secondaries
281
282   if (atype == 8) then
283     read(21,*) rt
284   elseif (atype == 9) then
285     read(20,*) rt
286   else
287     rt = taus88()
288   endif
289   thetas = asin(2*rt - 1) ! Calculate emission angle of secondaries with respect to the
290   normal
291
292   pvf(1,i) = vs(1) * cos(thetas(1)) ! Normal component of secondary velocity
293
294   pvf(1,j) = sin(phis(1)) * vs(1) * sin(thetas(1))
295   pvf(1,3) = cos(phis(1)) * vs(1) * sin(thetas(1))
296
297   if (pcol(1) == 1 .OR. pcol(2) == 1) then
298     pvf(1,i) = - abs(pvf(1,i)) ! Normal velocity pointing into the inside of the
299     waveguide
300   elseif (pcol(3) == 1 .OR. pcol(4) == 1) then
301     pvf(1,i) = abs(pvf(1,i)) ! Normal velocity pointing into the inside of the waveguide
302   endif
303
304   deallocate (yn)
305   deallocate (thetas)
306   deallocate (phis)
307   deallocate (penf)
308   deallocate (vs)

```

```

308     endif
309     elseif (s > 1) then
310     !DEBUG
311         if (allocated(SEYvsE)) then
312             call add23Darray(SEYvsE, (/0,0,1/))
313             k = size(SEYvsE,3)
314             SEYvsE(3,1,k) = penf1
315             SEYvsE(3,2,k) = delta
316         else
317             allocate(SEYvsE(4,2,1))
318             SEYvsE(3,1,1) = penf1
319             SEYvsE(3,2,1) = delta
320         endif
321
322     allocate (pvf(s,3))
323     allocate (yn(s))
324     allocate (thetas(s))
325     allocate (phis(s))
326     allocate (penf(s))
327     allocate (vs(s))
328
329     ! Initialize variables
330     pvf = 0.          ! m x 3 array of 3D velocities of m secondary electrons
331     yn = 0.           ! magnitude of velocity of secondary electrons
332     thetas = 0.       ! Emission angle of secondaries with respect to the normal
333     phis = 0.         ! Azimuthal emission angle of secondaries
334     penf = 0.
335     vs = 0.
336
337     call gratio(real(p_n*s,8), real(penf1/Eom,8), ans, qans, 0)
338     if (atype == 8) then
339         read(21,*) rt
340     elseif (atype == 9) then
341         read(20,*) rt
342     else
343         rt = taus88()
344     endif
345     ans = ans*rt
346     call gaminv(real(p_n*s,8), y, real(0,8), ans, 1-ans, ierr)
347
348     Y = sqrt( Y )      ! v^tilde
349
350     !! Scalar parameters needed to obtain velocity components
351     sint = 1.
352     do k = 1,s-1
353         lnbeta = betaln(real(p_n*(s-k),4), real(p_n,4))
354
355         if (atype == 8) then
356             read(21,*) rt
357         elseif (atype == 9) then
358             read(20,*) rt
359         else
360             rt = taus88()
361         endif
362         inbeta = betain( real(rt,8), real(p_n*(s-k),8), real(p_n,8), lnbeta, ifault )
363         if (ifault /= 0) then
364             write(*,*) 'error calculating incomplete beta function. Terminating.'
365             call exit
366         endif
367
368         alpha = xinbta( real(p_n*(s-k),8), real(p_n,8), lnbeta, inbeta, ifault )
369         if (ifault /= 0) then
370             write(*,*) 'error calculating inverse incomplete beta function. Terminating.'

```

```

371     call exit
372 endif
373
374     alpha = asin( sqrt( alpha ) )      ! Alpha angles used to calculate the magnitude of
velocity
375
376     yn(k) = Y * sint * cos(alpha)      ! magnitude of outgoing velocities
377
378     sint = sint * sin(alpha)           ! Spherical coordinates factor
379 enddo
380 yn(s) = Y * sint
381
382 ! Determine which is the normal component of the outgoing velocity
383 if (pcol(1)==1 .OR. pcol(3)==1) then
384     i = 1                             ! Normal component
385     j = 2                             ! Parallel component(not z)
386 elseif (pcol(2)==1 .OR. pcol(4)==1) then
387     i = 2                             ! Normal component
388     j = 1                             ! Parallel component (not z)
389 endif
390
391 do k = 1,s
392     penf(k) = Eom * (yn(k))**2
393     vs(k) = sqrt(2*penf(k)*(-eme))
394
395     if (atype == 8) then
396         read(21,*) rt
397     elseif (atype == 9) then
398         read(20,*) rt
399     else
400         rt = taus88()
401     endif
402     phis(k) = rt * 2 * pi              ! Calculate emission azimuthal angle of secondaries
403
404     if (atype == 8) then
405         read(21,*) rt
406     elseif (atype == 9) then
407         read(20,*) rt
408     else
409         rt = taus88()
410     endif
411     thetas(k) = asin(2*rt - 1)         ! Calculate emission angle of secondaries with respect to
the normal
412
413     pvf(k,i) = vs(k) * cos(thetas(k)) ! Normal component of secondary velocity
414     pvf(k,j) = sin(phis(k)) * vs(k) * sin(thetas(k))
415     pvf(k,3) = cos(phis(k)) * vs(k) * sin(thetas(k))
416 enddo
417
418 if (pcol(1) == 1 .OR. pcol(2) == 1) then
419     do k = 1,s
420         pvf(k,i) = - abs(pvf(k,i))    ! Normal velocity pointing into the inside of the
waveguide
421     enddo
422 elseif (pcol(3) == 1 .OR. pcol(4) == 1) then
423     do k = 1,s
424         pvf(k,i) = abs(pvf(k,i))     ! Normal velocity pointing into the inside of the waveguide
425     enddo
426 endif
427
428 deallocate (yn)
429 deallocate (thetas)
430 deallocate (phis)

```

```
431    deallocate (penf)
432    deallocate (vs)
433
434    elseif (s == 0) then
435        allocate (pvf(1,3))
436        pvf = 0.
437    endif
```