```
Sk_cos(:) = 0.0_wp
Sk_sin(:) = 0.0_wp

num_blocks = (num_atoms-1) / block_vector_size + 1

! Compute Sk_cos/Sk_sin for each k-point
do iblock = 1, num_blocks
  jstart_block = (iblock-1) * block_vector_size + 1
  jend_block = max(jstart_block + block_vector_size - 1, num_atoms)

  ! Loop on all k-points : num_kpoints = (2*n_max+1)*(2*l_max*m_max+l_max+m_max)
  do imode = 1, num_kpoints
    (...)
    if (knorm2 <= knorm2_max) then

      ! Compute block contribution to Sk_cos/Sk_sin for this (l,m,n) k-point
      do j = jstart_block, jend_block
        ! load precomputed cos/sin values
        (...)
        ! Compute cos/sin values using trigonometric rules
        (...)
        Sk_cos(imode) = Sk_cos(imode) + q_elec(j)*cos_kxkykz
        Sk_sin(imode) = Sk_sin(imode) + q_elec(j)*sin_kxkykz
      end do
    end if
  end do
end do

! Compute potential on each electrode atom
do iblock = 1, num_blocks
  istart_block = (iblock-1) * block_vector_size + 1
  iend_block = max(jstart_block + block_vector_size - 1, num_atoms)

  ! Loop on all k-points : num_kpoints = (2*n_max+1)*(2*l_max*m_max+l_max+m_max)
  do imode = 1, num_kpoints
    (...)
    if (knorm2 <= knorm2_max) then
      ! For each atom, compute the contribution of this k-point to the potential
      (...)
      do i = istart_block, iend_block
        ! Load precomputed cos/sin values
        (...)
        ! Compute cos/sin values using trigonometric rules
        (...)
        V(i) = V(i) + Sk_alpha * (Sk_cos(imode)*cos_kxkykz &
                                 + Sk_sin(imode)*sin_kxkykz)

      end do
    end if
  end do
end do
```