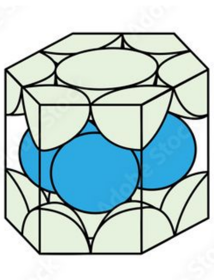
Diffusion Simulation

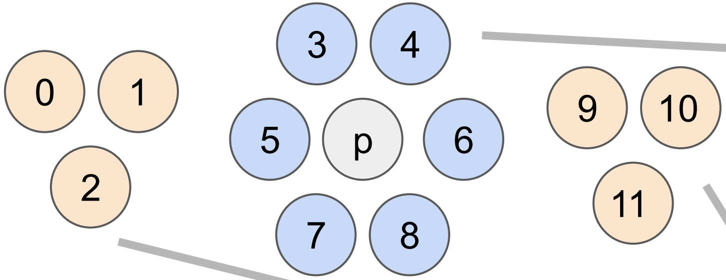
Mike Dillender

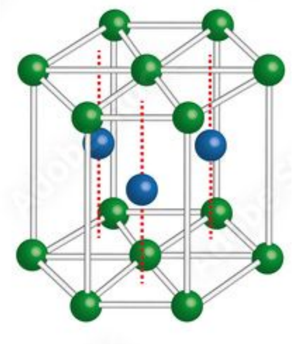
Assumption of Perfect Order

To make this easier, I assume we initially have a perfect hexagonal lattice



This hexagonal lattice can be mapped to a simple 3D array, using a set of simple rules to connect the QDs to their 12 nearest neighbors:

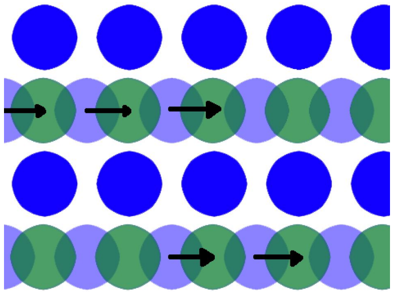




NNs on same y-layer:

```
y % 2 == 0      y % 2 == 1
  3  4  -        -  3  4
  5  p  6        5  p  6
  7  8  -        -  7  8
```

NNs on layer underneath:



```
        y % 2 = 0                            y % 2 = 1
z % 2 == 0     z % 2 == 1        z % 2 == 0     z % 2 == 1
 -  -  -          -  2  -         -  -  -         -  -  2
 1  0  -          -  0  1         1  0  -         -  0  1
 2  -  -          -  -  -         -  2  -         -  -  -
```

Scale axes:

$x_h = x$

$y_h = y*\mathrm{sqrt}(3)/2$

$z_h = z*\mathrm{sqrt}(6)/3$

FRET Rates

Assign random energies and dipole orientations, then use:

$$k_{i \to j} = \frac{1}{\tau}\left(\frac{R_0}{d_{ij}}\right)^6$$

$$R_0^{\ 6} = C\kappa^2 \frac{1}{\left[\frac{1}{2}\left(\varepsilon_D + \varepsilon_A - \Delta_{ss}\right)\right]^4} \exp\left[-\frac{\left(\varepsilon_D - \varepsilon_A - \Delta_{ss}\right)^2}{4\sigma_h^{\ 2}}\right]$$

$$\kappa = \hat{\mu}_D \cdot \hat{\mu}_A - 3\left(\hat{\mu}_A \cdot \hat{d}_{DA}\right)\left(\hat{\mu}_D \cdot \hat{d}_{DA}\right)$$

$$C \equiv \frac{9}{128}\frac{c^4 h^4}{\pi^5 n^4}\eta_{PL}\frac{\sigma_A\left(\varepsilon_A\right)}{\sqrt{2}}$$

For each dot in *[x][y][z]*, compute the transfer to the QD in each direction *i*:

$$rates[x][y][z][i] = \tau k_{i \to j} = (C/d_{ij}^6)[\tfrac{1}{2}(E_D + E_A - \Delta_{ss})]^4 exp\{-(E_D - E_A - \Delta_{ss})^2/4\sigma_h^2\}$$

The total rate at which the dot transfers or radiates =

$$\tau k_{tr/rad}[x][y][z] = 1 + \sum_{i=0}^{11} rates[x][y][z][i]$$

Simulation of Spectral Diffusion

Simulate excitation of a QD by calling **simulate_exciton(random_position, 0)**

```
void simulate_exciton(position, time)
        time_before_transfer = random sample from exponential distribution with λ = k_tr/rad
        random_value = (random decimal from 0-1) * ( k_tr/rad)
        If ( random_value < 1 )                                        (radiative recombination)
                APD[energy][time+time_before_transfer] ++              (Add a detection at this energy and time)
        else if( random_value < 1 + rates[x][y][z][0] )                (FRET in direction 0)
                position.move_in_direction(0)
                simulate_exciton(position, time + time_before_transfer)   (transfer the exciton to direction)
        else if( random_value < 1 + rates[x][y][z][0] + rates[x][y][z][1] )
                position.move_in_direction(1)
                simulate_exciton(position, time + time_before_transfer)
        else if( random_value < 1 + rates[x][y][z][0] + rates[x][y][z][1] + rates[x][y][z][2] )...
```