

KiMonETSim documentation

Macià Mut Sbert

August 2019

Resum

S'ha desenvolupat i implementat una metodologia per simular la transferència d'energia en materials semiconductors orgànics moleculars. Al donar-se aquesta transferència en un règim incoherent, s'ha modelat amb l'algorisme de *Kinetic Monte-Carlo* amb llenguatge *python*. El programa rep per nom KiMonETSim (*Kinetic Monte-Carlo Energy Transfer Simulator*).

Aquest text presenta la documentació que un usuari necessita per córrer el script de llançament i la informació sobre quins processos es poden simular i com es podria estendre el programa per incloure'n més.

1 Launch script

Aquest primer fitxer és amb el que l'usuari pot llançar una simulació. Es podria dir que és el fitxer principal de llançament, doncs tot anàlisi posterior partirà de dades retornades d'aquest.

Aquest script consisteix en la implementació del programa KiMonETSim en sí.

1.1 Inputs

Per fer la simulació el programa necessita quatre conjunts de dades. Per una banda la informació sobre els paràmetres interns de la molècula. Per altra les condicions físiques externes en què es dona la simulació. I finalment definir la morfologia del sistema on hi ha la transferència d'energia. A part, es pot donar l'excitació inicial del sistema.

L'usuari també pot fixar el número de trajectòries que s'han de simular així com un fitar un màxim d'iteracions per trajectòria (s'eviten *loops* infinits).

1.1.1 Definició de la molècula

En el programa les molècules es tracten com una classe de *python*. Aquesta ja duu incorporats mètodes que actuen sobre les seves variables internes, per exemple pel càlcul dels rates de decaïment.

La construcció del sistema on estudiarem la transferència parteix de una instància genèrica d'aquesta classe. Per inicialitzar-la es necessita:

- Diccionari amb les energies d'excitació de cada estat electrònic considerat (donades en eV).
- Diccionari anàleg amb les energies de relaxació (donades en eV).
- El vector moment de transició dipolar, donat com un *num-py array* (donat en unitats atòmiques).

Les claus dels diccionaris seran les que caracteritzaran els estats al llarg de tot el programa i anàlisis posteriors. Per tant, és precís que, un cop definides, no es canviïn. Els estats implementats i les respectives claus són:

- Estat fonamental: 'gs'
- Primer estat singlet excitat: 's1'

1.1.2 Condicions físiques

Tota la informació relacionada amb les condicions físiques externes al sistema s'agruparan en forma de diccionari. A l'actual nivell de desenvolupament (Agost 2019) aquest necessita les següents entrades:

- Temperatura (donada en Kelvin)
- Índex de refracció del material

A més incluïm en aquest diccionari una **distància màxima d'interacció** entre molècules. Aquest actua com un *cut-off* de la interacció. Aquesta distància s'expressa en **nm**.

1.1.3 Morfologia

L'usuari pot definir també quina forma té el sistema on es transparen els excitons. Es poden donar dos conjunts de paràmetres depenent de si es vol un sistema ordenat (material cristal·lí) o un sistema desordenat (material amorf).

- **Lattice**. Aquest diccionari conté la informació necessària per inicialitzar una xarxa cristal·lina. Les entrades són:
 - La super cel·la de la xarxa (número de molècules per costat). Pitònicament es dóna com una llista i la longitud d'aquesta és la dimensionalitat del sistema.
 - El paràmetre de xarxa en cada direcció (nm). S'ha de donar com una llista, òptimament de la mateixa longitud que les dimensions.

Es podria expandir incloent els àngles que formen els enllaços entre dos veïns.

- **Amorphous.** Diccionari que recull la informació per inicialitzar un material amorf. Aquest diccionari inclou les dimensions físiques del material (volum total) en nm i el número total de molècules. El programa reparteix les molècules de manera aleatòria en aquest volum (tot assegurant que no es superposin). Aquesta opció no ha estat gaire desenvolupada.

El programa entén que donat un diccionari l'altre ha de ser **None**. Doncs només un dels dos serà usat.

Juntament amb les dimensions físiques del sistema i número de molècules l'usuari pot fixar l'orientació relativa entre molècules. Aquesta pot ser: *parallel*, *anti-parallel* o *random*. La primera indica que totes estan orientades en la mateixa direcció, la segona que cada molècula troba els seus veïns primers orientats antiparal·lelament i l'última indica que a cada molècula se li associa una orientació aleatòria. Cal comentar que no es contempla una situació amb orientació *anti-parallel* per un sistema amorf. En el cas de que ambdues comandes siguin donades l'usuari rebria per pantalla un avís de la seva incompatibilitat i no es generaria cap sistema.

1.1.4 Excitació inicial

El programa està pensat per simular la transferència d'un excitó en un possibles sistemes explicats anteriorment. L'usuari pot fixar quin és l'estat excitat inicial del sistema.

L'estructura computacional és un diccionari on la clau s'espera que sigui una de les etiquetes dels possibles estats excitats que s'han presentat en l'apartat 1.1.1 i els arguments són una llista amb les posicions dels excitons en el material. Aquesta posició es pot donar en diversos formats.

- **Posicions tipus *string*.** La posició de l'excitó ve donada per una paraula. Les possibles comandes són: *random*, *firsts*, *last*, *centre* i *furthest*. *First*, *last* fan referència a la posició dels excitons en la llista de instàncies de la classe genèrica que es genera i *furthest* a la posició més llunyana del centre (situat en el punt $\vec{0}$).
- **Posicions tipus *integer*.** S'entén com la líndex de la molècula en la llista de molècules esmentades.
- **Posicions tipus *list*.** El programa ho pren com les coordenades cartesianes de la molècula on ha d'estar localitzada l'excitació.

En qualsevol de les situacions esmentades en què s'especifica la molècula que s'ha d'excitar, si ja ho està es prendria una posició aleatòria (al nivell de desenvolupament d'Agost 2019 només es contempla que cada molècula estigui excitada a un sol estat).

Per les situacions en què es concreten unes coordenades per la molècula el programa buscarà primer si en un entorn, donat per un radi de tolerància, hi ha una molècula. Si no fos així es buscaria la més propera (en un radi de 200nm). Si no hi ha cap molècula en aquest radi es pren una posició aleatòria de la llista. En qualsevol cas l'usuari reb una alerta per pantalla.

1.1.5 Valors per defecte

Es dóna un conjunt de valors per aquestes variables per poder córrer el script directament. Per defecte es pren:

- Energies d'excitació: $E_0 = 0.0eV$, $E_{S1} = 2.5eV$
- Energies de relaxació: $E_0 = 0.0eV$, $E_{S1} = 0.7eV$
- Moment de transició dipolar: $(1, 0, 0)au$
- Temperatura: $T = 273.15K$
- Índex de refracció: $n = 1$
- Radi *cut-off*: $R_{max} = 1.1nm$
- Dimensions de la supercel·la: (1000) (1-dimensional)
- Paràmetre de xarxa: $1.0nm$
- Orientació relativa dels dipols: *parallel*
- Excitació inicial: La molècula en la posició *centre* a l'estat 's1'.

1.2 Procediment del programa

1.2.1 Inicialització del sistema

En primer lloc, a partir de la informació donada sobre el sistema i de la molècula genèrica inicialitzada, es crea el sistema. Aquesta variable, *system*, té estructura de diccionari. Aquest pas i totes les operacions posteriors es fan per cada trajectòria (*loop* sobre el número de trajectòries).

La principal entrada d'aquest diccionari consisteix en una llista de totes les instàncies de la classe molècula. A diferència de la genèrica passada com a *input*, aquestes contenen informació sobre la seva posició i orientació en l'espai. A més, *system* recull en altres dues entrades els diccionaris amb les condicions físiques i amb la morfologia. D'aquesta manera es tenen de manera compacta tots els paràmetres que es poden necessitar en futurs càlculs.

A part, incloem dues entrades més. Una llista amb els índexos de les molècules excitades en la llista i una etiqueta sobre el tipus de sistema. Aquestes tenen un paper enfocat cap a l'eficiència del programa. Tot i això, són imprescindibles pel funcionament del programa.

1.2.2 Actualització del sistema

S'entén un pas de la trajectòria com el succés d'un dels possibles processos considerats, ja sigui de transferència d'un excitó, o bé de decaïment. La tria d'aquest procés es farà per mitjà de l'algorisme *Kinetic Monte-Carlo (KMC)*.

En cada pas (una iteració d'un *loop* sobre el número total de passos permès per trajectòria) el sistema s'ha d'actualitzar. Per això es crida una funció que

sols reb per argument la variable *system* i retorna el procés triat i el temps que se li associa a aquest procés. Computacionalment, el procés triat és un diccionari amb 4 entrades. Per una banda l'índex del donor i l'índex de l'acceptor en la llista de molècules i per una altra l'etiqueta del procés. Finalment l'índex de l'excitó que ha patit el procés. Aquesta darrera variable està pensada pel cas en què hi hagi més d'un excitó propagant-se poder reconstruir les trajectòries individuals. Reb el nom d'índex perquè fa referència a la posició que ocupa l'excitó en la llista de les molècules excitades.

La funció que fa tot aquest procediment és el *core* del programa. Els processos interns d'aquesta funció inclouen la cerca de veïns, el càlcul dels *rates* de transferència i decaïment i l'aplicació del *KMC* per triar un procés. Finalment s'actualitza la variable *system* modificant la llista de molècules excitant i desexcitant les corresponents.

A la llista amb els índexs de les molècules excitades es modifica sols la posició de l'excitó que s'ha alterat. A la posició afectada s'hi guarda l'índex de l'acceptor si es tracta d'un procés de transferència o se la fa **None** en cas d'un decaïment. D'aquesta manera s'assegura que no es modifiquen la resta de posicions i es manté l'ordre de la llista. Les etiquetes dels índexos segueixen sent vàlides. Tot i que el nivell de desenvolupament actual (Agost 2019) no ho contempla, si es donàs un procés on es creen nous excitons aquests es guardarien en posicions que s'afegirien i la posició anterior es consideraria **None**.

Les dades obtingudes d'aquesta funció s'utilitzen per actualitzar la variable *trajectory*. Aquesta també és un diccionari i s'encarrega de arregar la informació del procés triat en cada pas amb cinc entrades:

- **Temps.** Variable acumulativa que va guardant l'instant de temps en què ocorre cada procés.
- **Procés.** Etiqueta del procés triat.
- **Número d'excitons** Quantitat d'excitons que hi ha propagant-se en cada pas. Pot anar canviant a mesura que decaiguin o bé se'n generin de nous.
- **Posicions.** Llista amb les posicions dels excitons. La posició d'un excitó inclou, a part de les coordenades cartesianes el seu estat excitat (etiqueta corresponent).
- **Índex de l'excitó.** Diu quin excitó ha sofrit el procés.

La simulació d'una trajectòria pararà quan tots els excitons han decaïgut. Computacionalment s'implementa comprovant si totes les posicions de la llista de molècules excitades són **None**.

1.3 Outputs

El programa retorna un fitxer format *json*. S'ha triat aquest format per què permet establir una correspondència entre les estructures de *python* i les seves estructures internes. Això facilitarà la seva lectura. En aquest s'hi escriu un diccionari separat en dues parts.

- **Capçalera.** Un diccionari que recull les condicions amb què s’ha duit a terme la simulació. Ha d’incloure els paràmetres que siguin necessaris per un anàlisi posterior i també ha de ser suficient per distingir-la d’una altra situació.
- **Trajectòries.** Llista amb totes les trajectòries simulades. És l’entrada que aporta la informació extreta de la trajectòria. Constitueix l’*output* en sí.

Donats dos fitxers, si tenen la mateixa *Capçalera* es poden ajuntar les dues llistes de trajectòries i així tenir un conjunt més nombrós alhora de fer una estadística.

2 *Package* processos

Aquest paquet recull l’implementació de tota la física del programa. Inclou l’estudi dels acoblaments electrònics i solapaments espectrals entre les molècules involucrades i les implementacions dels mètodes que permetran obtenir els *rates* dels diferents processos.

Aquest paquet no es crida directament en el *script* de llançament sino que forma part de la infraestructura del *core*. La seva funció és calcular els *rates* dels processos possibles on després s’hi aplicarà el *KMC*. Per un bon funcionament de l’algorisme, la funció principal d’aquest paquet ha de retornar dues llistes. Una llista amb els processos i una segona llista amb els *rates* associats a cada procés (sols el valor numèric). Els elements d’ambdues llistes referits a un mateix succés coincideixen en el mateix índex. El format amb què es donarà un procés serà de diccionari amb quatre entrades: índex del donor, índex de l’acceptor, etiqueta del procés i índex de l’excitó (és el mateix format amb què es rebia el procés triat en el *script* de llançament).

2.1 Decaïment

Els processos de decaïment venen donats per un mètode intern de la classe molècula. Donada una molècula en un determinat estat excitat, el mètode estudia quins possibles decaïments hi ha per aquest estat. La crida del mètode retorna un diccionari amb l’etiqueta del procés com a clau i el *rate* de decaïment com argument (en ns^{-1}).

Aquest mètode es complementa amb una funció del *package*. Aquesta s’encarrega de construir el format desitjat: llista amb els valors dels *rates* i llista amb les variables *process*. A part, a aquesta funció se li dóna memòria sobre els *rates* calculats. Cada cop que es crida el mètode per una molècula en un estat determinat es guarda el diccionari que retorna. Així si es torna trobar una molècula igual en el mateix estat es poden usar els valors guardats en memòria.

A l’actual nivell de desenvolupament (Agost 2019) sols es contempla un estat excitat, el primer singlet ‘s1’. Per tant el mètode tendrà una única entrada.

Per aquest estat es considera un decaïment radiatiu donat per l'expressió:

$$k_{rad} = \frac{4E_S^3 \mu^2}{3c^3}$$

S'inclou una entrada addicional de control per l'estat fonamental. Si es crida el mètode per una molècula no excitada es retorna un *rate* 0 i un avís que indica que una molècula a l'estat fonamental no pot decaure.

El programa està pensat per ser extensible i poder introduir nous estats excitats. Per estudiar el seu decaïment s'hauria de modificar el mètode intern de la molècula incloent noves entrades per cada estat afegit i implementar l'expressió física corresponent pel *rate* de decaïment. És precís que el format de sortida sigui conseqüent amb el proposat.

2.2 Transfarències

S'inclouen dins la categoria de processos de transfarència aquells que impliquen la participació de més d'una molècula.

La funció principal que tractarà aquests processos parteix d'un donor (molècula inicialment excitada) i el conjunt de possibles acceptors (molècules situades a una distància menor que el *cut-off*). Cerca les possibles interaccions entre les parelles donor-acceptor i calcula el *rate* del procés mitjançant la *Fermi's Golden Rule*.

$$k_{DA} = \frac{2\pi}{\hbar} |V_{DA}|^2 J_{DA}$$

- V_{DA} és l'acoblament electrònic entre el donor i l'acceptor.
- J_{DA} és el solapament entre l'espectre d'emissió del donor i el d'absorció de l'acceptor.

La funció principal esmentada, anomenada *get_transfer_rates*, sols fa el producte d'aquests dos factors. Ambdós es calculen de manera externa i independent.

2.2.1 Solapament espectral

Pel solapament espectral se suposa que totes les simulacions es realitzaran en unes condicions on es pugui usar la mateixa aproximació per qualsevol procés / acoblament electrònic. D'entrada s'aproxima per la fórmula de Marcus:

$$J_{DA} = \frac{1}{\sqrt{4\pi k_B T \lambda}} \exp\left(-\frac{(\Delta G + \lambda)^2}{4k_B T \lambda}\right)$$

En aquesta T actua com la temperatura, λ com l'energia de reorganització de l'acceptor i ΔG com la diferència energètica entre les posicions d'equilibri de l'estat excitat del donor i de l'acceptor. Finalment k_B és la constant de Boltzmann.

La funció que fa aquest càlcul requereix per *inputs* les instàncies de la classe molècula del donor i l'acceptor i les condicions físiques de la simulació. Aquesta funció conté la implementació de la fórmula presentada i retorna el valor d'aquest solapament en unitats atòmiques (no eV).

Per evitar cost computacional s'ha dotat de certa memòria a aquesta funció de tal manera que si es troba una situació amb el mateix tipus de donor, mateix tipus d'acceptor i a la mateixa temperatura que una situació ja calculada reutilitzarà el càlcul fet. El format d'aquesta memòria és un diccionari on la clau és la contracció *hash* de $T, \Delta G, \lambda$ i l'argument el valor de J_{DA} .

2.2.2 Acoblament electrònic

L'acoblament electrònic es tracta en un fitxer a part on es defineixen funcions per cada aproximació considerada. Aquestes funcions s'ajunten en un diccionari, *couplings* que es crida en la funció *get_transfer_rates*. La clau d'aquest diccionari serà l'etiqueta del procés de transfarència i l'argument la funció que retorna l'acoblament en unitats atòmiques.

Característiques i restriccions de *couplings*:

- Si dos processos es relacionen amb el mateix acoblament s'introduïran dues entrades amb el mateix argument.
- Els inputs de totes les funcions ajuntades ha de ser el mateix: instàncies de la classe molècula del donor i l'acceptor i les condicions físiques de la simulació.
- D'igual manera l'output ha de ser el mateix: el valor en unitats atòmiques d'aquest acoblament.
- Per ser coherents amb el procediment de la funció que retorna els *rates*, és precís que les claus tinguin un mateix format: {etiqueta de l'estat inicial del donor} - {etiqueta de l'estat inicial de l'acceptor} - {possible informació extra del procés}. La correcta definició d'aquestes claus és capital doncs actuaran en altres parts del programa com les etiquetes què distingeixen els processos.

D'igual manera que en el càlcul del solapament espectral i dels *rates* de decaïment aquest fitxer està dotat de certa memòria per evitar repetir càlculs. El format és anàleg: la clau del diccionari és la compressió *hash* dels paràmetres característics de l'acoblament i l'argument el valor d'aquest en eV.

L'únic procés de transfarència considerat és la transmissió d'un excitó del primer singlet (únic estat excitat possible). Per aquest procés s'usa l'aproximació de Förster per l'acoblament electrònic. Aquesta es basa en la interacció clàssica dipol-dipol.

$$V_{DA} = V_{dip-dip} = \frac{\kappa\mu^2}{n^2R^3}$$

En aquesta expressió μ és el moment de transició dipolar, R la distància entre el donor i l'acceptor n l'índex de refracció del material i κ el factor d'orientació entre les dues molècules ¹. Aquest es defineix com:

$$\kappa = n_d \cdot n_a - 3(n_d \cdot e)(n_a \cdot e)$$

On les tres magnituds que apareixen són els vectors unitaris del moment de transició del donor (n_d), de l'acceptor (n_a) i del vector desplaçament entre els dos (e).

2.2.3 Procediment de *get_transfer_rates*

La funció *get_transfer_rates* és on convergeixen les dues estructures abans descrites. Requereix com a *inputs* els índexs del donor i una llista amb els dels seus veïns en la llista de molècules, el diccionari *system* i l'índex (etiqueta) que associam a l'excitació.

La funció itera sobre la llista de veïns (possibles acceptors) calculant per una banda el solapament espectral i per altra l'acoblament electrònic. Per aquesta segona part es construeix una etiqueta prefix com: {etiqueta de l'estat inicial del donor} _ {etiqueta de l'estat inicial de l'acceptor}. Aleshores es prenen del diccionari *couplings* totes les entrades que comencen igual i es calcula per la FGR el *rate* del procés (en ua). En el cas de que no hi hagi cap entrada amb l'etiqueta prefix definida es pren el *rate* com a nul.

A part de la part de càlcul, en aquesta funció també es defineix la variable *process*. És un diccionari amb 4 entrades: índex del donor, índex de l'acceptor, etiqueta del procés (clau del diccionari *couplings*) i índex de l'excitació.

Finalment la funció retorna dues llistes: llista dels *rates* dels processos (en ns^{-1}) i una llista amb els processos. Els índexs coincideixen.

¹Expressió per dues molècules químicament iguals.