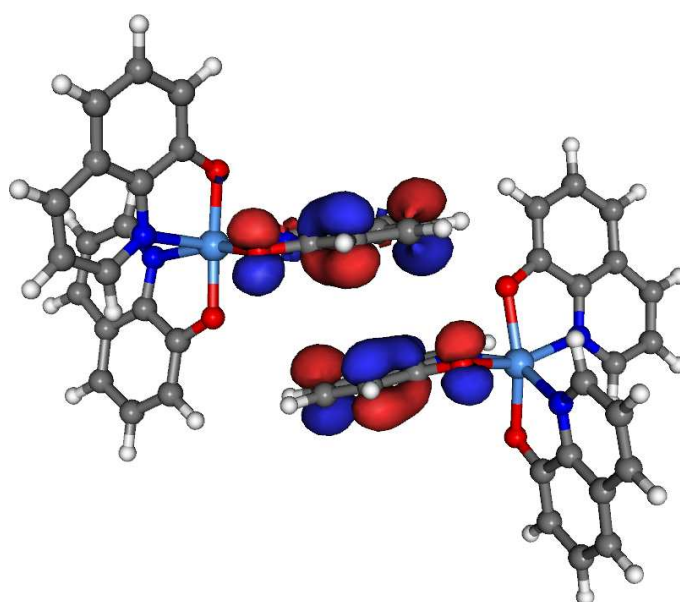


# VOTCA-XTP

## EXCITON TRANSPORT SIMULATIONS

### USER MANUAL



compiled from: 1.4.1

September 2, 2017

[www.votca.org](http://www.votca.org)

## Disclaimer

This manual is not complete. The best way to start using the software is to look at provided tutorials. The reference section is generated automatically from the source code, so please make sure that your software and manual versions match.

## Citations

Development of this software depends on academic research grants. If you are using the package, please cite the following papers

[1] Microscopic simulations of charge transport in disordered organic semiconductors, Victor Rühle, Alexander Lukyanov, Falk May, Manuel Schrader, Thorsten Vehoff, James Kirkpatrick, Björn Baumeier and Denis Andrienko  
*J. Chem. Theor. Comp.* **7**, 3335, 2011

[2] Versatile Object-oriented Toolkit for Coarse-graining Applications  
Victor Rühle, Christoph Junghans, Alexander Lukyanov, Kurt Kremer and Denis Andrienko  
*J. Chem. Theor. Comp.* **5**, 3211, 2009

## Development

The core development is currently taking place at the Max Planck Institute for Polymer Research, Mainz, Germany and TU/e Eindhoven.

## Copyright

VOTCA-XTP is free software. The entire package is available under the Apache License. For details, check the LICENSE file in the source code. The VOTCA-XTP source code is available on our homepage, [www.votca.org](http://www.votca.org).

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Theoretical background</b>	<b>5</b>
2.1	Workflow	5
2.2	Material morphology	5
2.3	Conjugated segments and rigid fragments	6
2.4	Neighbor list	8
2.5	Reorganization energy	8
2.5.1	Intramolecular reorganization energy	8
2.5.2	Outersphere reorganization energy	9
2.6	Site energies	10
2.6.1	Externally applied electric field	10
2.6.2	Internal energy	11
2.6.3	Electrostatic interaction energy	11
2.6.4	Induction energy - the Thole model	12
2.7	Transfer integrals	14
2.7.1	Projection of monomer orbitals on dimer orbitals (DIPRO)	15
2.7.2	DFT-based transfer integrals using DIPRO	16
2.7.3	ZINDO-based transfer integrals using MOO	19
2.8	Charge transfer rate	19
2.8.1	Classical charge transfer rate	19
2.8.2	Semi-classical bimolecular rate	20
2.8.3	Semi-classical rate	20
2.9	Master equation	21
2.9.1	Extrapolation to nondispersive mobilities	21
2.10	Stochastic Networks	22
2.10.1	Coarse-grained morphology	22
2.10.2	Charge transport network	25
2.11	Macroscopic observables	28
2.11.1	Charge density	28
2.11.2	Current	29
2.11.3	Mobility and diffusion constant	29
2.11.4	Spatial correlations of energetic disorder	30
<b>3</b>	<b>Input and output files</b>	<b>33</b>
3.1	Atomistic topology	33
3.2	Mapping file	35
3.3	Molecular orbitals	36
3.4	Monomer calculations for DFT transfer integrals	37
3.5	Pair calculations for DFT transfer integrals	39
3.6	DFT transfer integrals	41
3.7	State file	42

<b>4</b>	<b>Reference</b>	<b>45</b>
4.1	Programs	45
4.1.1	xtp_testsuite	45
4.1.2	xtp_update	45
4.1.3	xtp_update_exciton	45
4.1.4	xtp_basisset	45
4.1.5	xtp_map	46
4.1.6	xtp_run	46
4.1.7	xtp_tools	46
4.1.8	xtp_parallel	46
4.1.9	xtp_dump	47
4.1.10	xtp_overlap	47
4.1.11	xtp_kmc_run	47
4.2	Calculators	47
4.2.1	coupling	48
4.2.2	excitonicoupling	48
4.2.3	gencube	49
4.2.4	log2mps	49
4.2.5	molpol	49
4.2.6	orb2isogwa	50
4.2.7	partialcharges	50
4.2.8	pdb2map	50
4.2.9	pdb2top	50
4.2.10	ptopreader	50
4.2.11	qmanalyze	51
4.2.12	eanalyze	51
4.2.13	eimport	51
4.2.14	einternal	51
4.2.15	emultipole	51
4.2.16	eoutersphere	52
4.2.17	ianalyze	53
4.2.18	iimport	53
4.2.19	izindo	53
4.2.20	jobwriter	53
4.2.21	pairedump	53
4.2.22	panalyze	54
4.2.23	profile	54
4.2.24	rates	54
4.2.25	sandbox	54
4.2.26	stateserver	55
4.2.27	tdump	55
4.2.28	vaverage	55
4.2.29	zmultipole	55
4.2.30	edft	56
4.2.31	idft	56
4.2.32	qmmm	56
4.2.33	xqmultipole	57
4.2.34	energy2xml	58
4.2.35	integrals2xml	58
4.2.36	occupations2xml	58
4.2.37	pairs2xml	58
4.2.38	rates2xml	58
4.2.39	segments2xml	59
4.2.40	trajectory2pdb	59

<i>CONTENTS</i>	v
4.3 Common options . . . . .	59
<b>Bibliography</b>	<b>61</b>



# Index

- charge transfer rate, 19
  - bimolecular, 20
  - classical, 19
  - semiclassical, 20
- conjugated segment, 6
- current, 29
  - local, 29
- diabatic states, 14
- distributed multipoles, 11
- electronic coupling, 14
  - DFT, 16
  - ZINDO, 19
- hopping site, 6
- kinetic Monte Carlo, 21
- mobility, 29
- neighbor list, 8
- Pekar factor, 10
- reorganization energy, 8
  - intramolecular, 8
  - outersphere, 9
- rigid fragment, 7
- site energy, 10
  - distributed multipoles, 11
  - external field, 10
  - internal, 11
  - polarization, 12
  - spatial correlation, 30
- Thole model, 12
- transfer integral, *see* electronic coupling





# Chapter 1

## Introduction

sec:introduction

Charge carrier dynamics in an organic semiconductor can often be described in terms of charge hopping between localized states. The hopping rates depend on **electronic coupling elements**, **reorganization energies**, and **site energies**, which vary as a function of position and orientation of the molecules. The purpose of the VOTCA-XTP package [1] is to simplify the workflow for charge transport simulations, provide a uniform error-control for the methods, flexible platform for their development, and eventually allow *in silico* prescreening of organic semiconductors for specific applications.

The toolkit is implemented using modular concepts introduced earlier in the Versatile Object-oriented Toolkit for Coarse-graining Applications (VOTCA) [2]. It contains different **programs**, which execute specific tasks implemented in **calculators** representing an individual step in the workflow. Figure 1.1 summarizes a typical chain of commands to perform a charge transport simulation: First, the VOTCA code structures are adapted to reading atomistic trajectories, mapping them onto conjugated segments and rigid fragments, and substituting (if needed) rigid fragments with the optimized copies (`xtp_map`). The programs `xtp_run` and `xtp_parallel` (for heavy-duty tasks) are then used to calculate all bimolecular charge hopping rates (via precalculation of all required ingredients). **Site energies (or energetic disorder)** can be determined as a combination of internal (ionization potentials/electron affinities of single molecules) as well as electrostatic and polarization contributions within the molecular environment. The calculation of **electronic coupling elements** between conjugated segments from the corresponding molecular orbitals can be performed using a **dimer-projection** technique based on **density-functional** theory (DFT). This requires explicit calculations using quantum-chemistry software for which we provide interfaces to `Gaussian`, `Turbomole`, and `NWChem`. Alternatively, the **molecular orbital overlap** module calculates electronic coupling elements relying on the semi-empirical INDO Hamiltonian and molecular orbitals in the format provided by the `Gaussian` package.

The **kinetic Monte Carlo** module reads in the neighbor list, site coordinates, and hopping rates and performs charge dynamics simulations using either periodic boundary conditions or charge sources and sinks.

The toolkit is written as a combination of modular C++ code and scripts. The data transfer between programs is implemented via a state file (sql database), which is also used to restart simulations. Analysis functions and most of the calculation routines are encapsulated by using the observer pattern [3] which allows the implementation of new functions as individual modules.

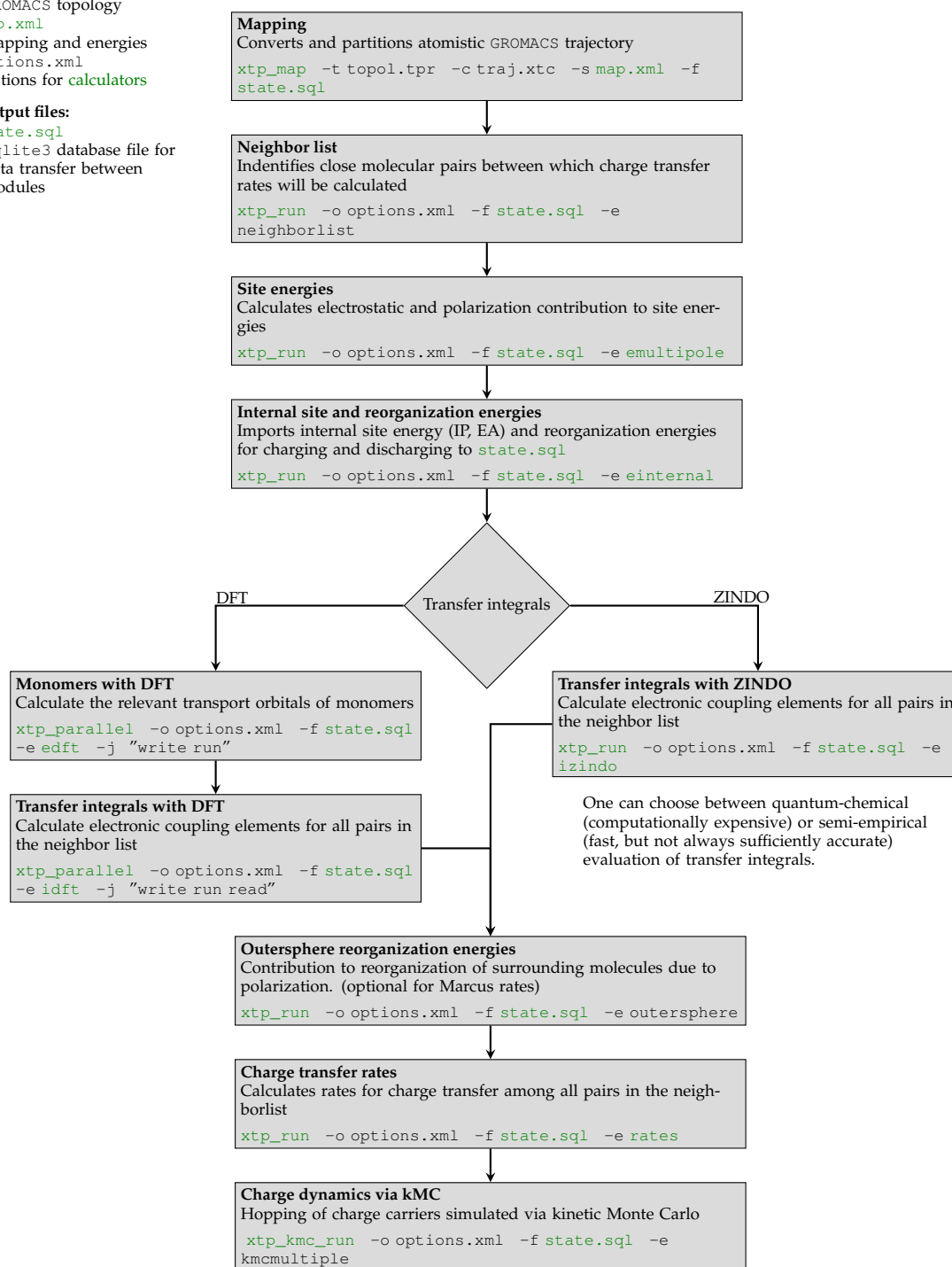
In the following chapter 2, we summarize the **theoretical background** of the workflow of charge transport simulations and in particular its individual steps. Chapter 3 describes the structure and content of input and output files, while a full reference of **programs** and **calculators** is available in chapter 4. For a hands-on tutorial, the reader is referred to the VOTCA-XTP project page at <http://code.google.com/p/votca-xtp/>.

**Input files:**

conf.gro  
GROMACS trajectory  
topol.tpr  
GROMACS topology  
map.xml  
mapping and energies  
options.xml  
options for calculators

**Output files:**

state.sql  
sqlite3 database file for  
data transfer between  
modules



Get list of available calculators: `xtp_run/xtp_parallel/xtp_kmc_run -l`

Get help and list of options for a calculator: `xtp_run/xtp_parallel/xtp_kmc_run -d neighborlist`

Figure 1.1: A practical workflow of charge transport simulations using VOTCA-XTP. The **theoretical background** of the individual steps is given in chapter 2. Chapter 3 describes the content of input and output files, while a full reference of **programs** and **calculators** is available in chapter 4. fig:summary

## 39 Chapter 2

# 40 Theoretical background

## 41 2.1 Workflow

42 A typical workflow of charge transport simulations is depicted in figure 2.1. The first step is  
43 the simulation of an atomistic morphology, which is then partitioned on hopping sites. The  
44 coordinates of the hopping sites are used to construct a list of pairs of molecules, or neighbor list.

fig:workflow

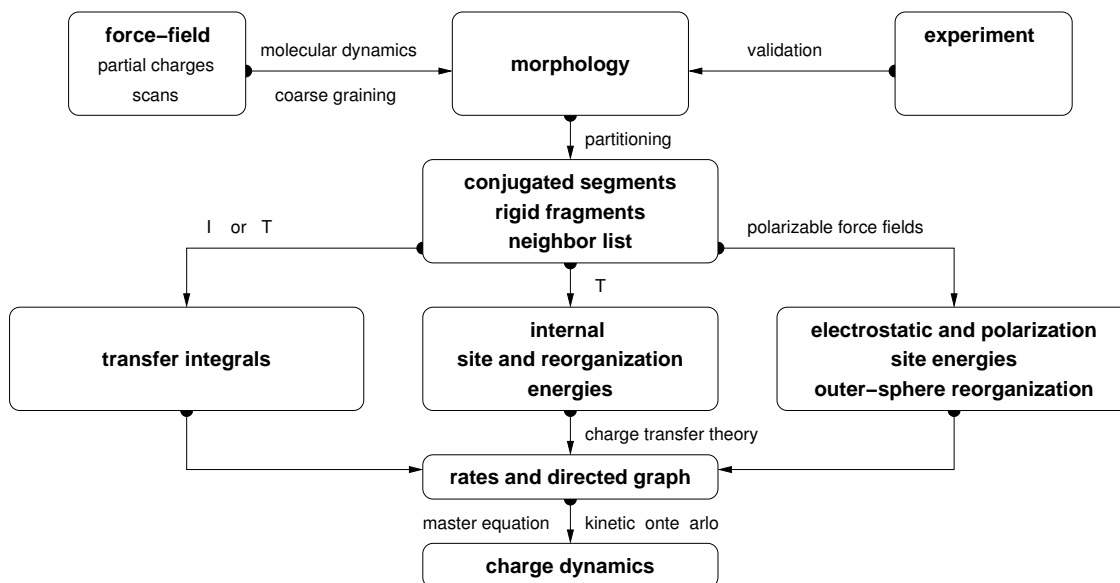


Figure 2.1: Workflow for microscopic simulations of charge transport.

45 For each pair an electronic coupling element, a reorganization energy, a driving force, and even-  
46 tually the hopping rate are evaluated. The neighbor list and hopping rates define a directed  
47 graph. The corresponding master equation is solved using the kinetic Monte Carlo method,  
48 which allows to explicitly monitor the charge dynamics in the system as well as to calculate time  
49 or ensemble averages of occupation probabilities, charge fluxes, correlation functions, and field-  
50 dependent mobilities.

## 51 2.2 Material morphology

52 There is no generic recipe on how to predict a large-scale atomistically-resolved morphology of  
53 an organic semiconductor. The required methods are system-specific: for ultra-pure crystals, for

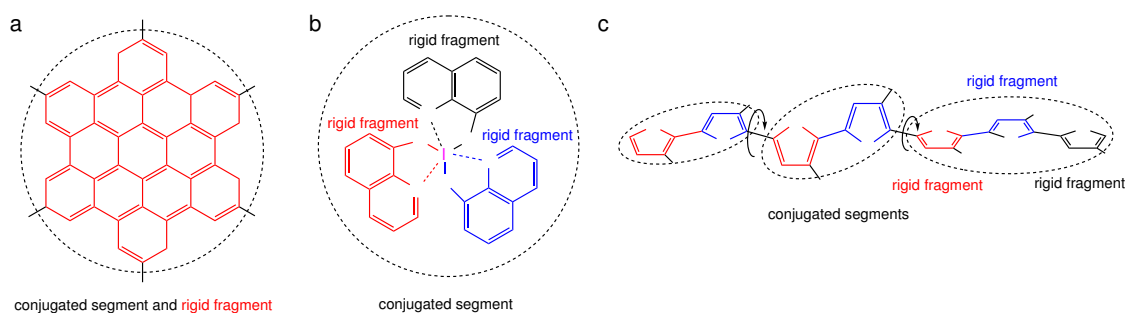


Figure 2.2: The concept of conjugated segments and rigid fragments. Dashed lines indicate conjugated segments while colors denote rigid fragments. (a) Hexabenzocoronene: the  $\pi$ -conjugated system is both a rigid fragment and a conjugated segment. (b)  $\text{Alq}_3$ : the Al atom and each ligand are rigid fragments while the whole molecule is a conjugated segment. (c) Polythiophene: each repeat unit is a rigid fragment. A conjugated segment consists of one or more rigid fragments. One molecule can have several conjugated segments.

fig:segment

54 example, density-functional methods can be used provided the crystal structure is known from  
 55 experiment. For partially disordered organic semiconductors, however, system sizes much larger  
 56 than a unit cell are required. Classical molecular dynamics or Monte Carlo techniques are then  
 57 the methods of choice.

58 In molecular dynamics, atoms are represented by point masses which interact via empirical po-  
 59 tentials prescribed by a force-field. Force-fields are parametrized for a limited set of compounds  
 60 and their refinement is often required for new molecules. In particular, special attention shall  
 61 be paid to torsion potentials between successive repeat units of conjugated polymers or between  
 62 functional groups and the  $\pi$ -conjugated system. First-principles methods can be used to charac-  
 63 terize the missing terms of the potential energy function.

64 Self-assembling materials, such as soluble oligomers, discotic liquid crystals, block copolymers,  
 65 partially crystalline polymers, etc., are the most complicated to study. The morphology of such  
 66 systems often has several characteristic length scales and can be kinetically arrested in a thermo-  
 67 dynamically non-equilibrium state. For such systems, the time- and length-scales of atomistic  
 68 simulations might be insufficient to equilibrate or sample desired morphologies. In this case,  
 69 systematic coarse-graining can be used to enhance sampling [2]. Note that the coarse-grained  
 70 representation must reflect the structure of the atomistic system and allow for back-mapping to  
 71 the atomistic resolution.

72 Here we assume that the morphology is already known, that is we know how the topology and  
 73 the coordinates of all atoms in the systems at a given time. VOTCA-XTP can read standard  
 74 GROMACS topology files. Custom definitions of **atomistic topology** via XML files are also possible.  
 75 Since the description of the atomistic topology is the first step in the charge transport simulations,  
 76 it is important to follow simple conventions on how the system is partitioned on molecules,  
 77 residues, and how atoms are named in the topology. Required input files are described in section  
 78 **atomistic topology**.

## 79 2.3 Conjugated segments and rigid fragments

sec:segments


80 With the morphology at hand, the next step is partitioning the system on hopping sites, or con-  
 81 jugated segments, and calculating charge transfer rates between them. Physically intuitive argu-  
 82 ments can be used for the partitioning, which reflects the localization of the wave function of  
 83 a charge. For most organic semiconductors, the molecular architecture includes relatively rigid,  
 84 planar  $\pi$ -conjugated systems, which we will refer to as rigid fragments. A conjugated segment  
 85 can contain one or more of such rigid fragments, which are linked by bonded degrees of freedom.

86 The dynamics of these degrees of freedom evolves on timescales much slower than the frequency  
 87 of the internal promoting mode. In some cases, e.g. glasses, it can be ‘frozen’ due to non-bonded  
 88 interactions with the surrounding molecules.

89 To illustrate the concept of conjugated segments and rigid fragments, three representative molec-  
 90 ular architectures are shown in figure 2.2. The first one is a typical discotic liquid crystal, hex-  
 91 abenzocoronene. It consists of a conjugated core to which side chains are attached to aid self-  
 92 assembly and solution processing. In this case the orbitals localized on side chains do not partic-  
 93 ipate in charge transport and the conjugated  $\pi$ -system is both, a rigid fragment and a conjugated  
 94 segment. In  $\text{Alq}_3$ , a metal-coordinated compound, a charge carrier is delocalized over all three  
 95 ligands. Hence, the whole molecule is one conjugated segment. Individual ligands are relatively  
 96 rigid, while energies of the order of  $k_B T$  are sufficient to reorient them with respect to each other.  
 97 Thus the Al atom and the three ligands are rigid fragments. In the case of a conjugated polymer,  
 98 one molecule can consist of several conjugated segments, while each backbone repeat unit is a  
 99 rigid fragment. Since the conjugation along the backbone can be broken due to large out-of-plane  
 100 twists between two repeat units, an empirical criterion, based on the dihedral angle, can be used  
 101 to partition the backbone on conjugated segments [4]. However, such intuitive partitioning is, to  
 102 some extent, arbitrary and shall be validated by other methods [5–7].

103 After partitioning, an additional step is often required to remove bond length fluctuations intro-  
 104 duced by molecular dynamics simulations, since they are already integrated out in the deriva-  
 105 tion of the rate expression. This is achieved by substituting respective molecular fragments with  
 106 rigid, planar  $\pi$ -systems optimized using first-principles methods. Centers of mass and gyration  
 107 tensors are used to align rigid fragments, though a custom definition of local axes is also possible.  
 108 Such a procedure also minimizes discrepancies between the force-field and first-principles-based  
 109 ground state geometries of conjugated segments, which might be important for calculations of  
 110 electronic couplings, reorganization energies, and intramolecular driving forces.

111 To partition the system on hopping sites and substitute rigid fragments with the corresponding  
 112 ground-state geometries `xtp_map` program is used:

 **Mapping the GROMACS trajectory**  
 | `xtp_map -t topol.tpr -c traj.xtc -s map.xml -f state.sql`

113 It reads in the GROMACS topology (`topol.tpr`) and trajectory (`traj.xtc`) files, definitions of  
 114 conjugated segments and rigid fragments (`map.xml`) and outputs coordinates of conjugated seg-  
 115 ments (hopping sites) and rigid fragments (as provided in the MD trajectory and after rigidi-  
 116 fication) to the state file (`state.sql`). In order to do this, a mapping file `map.xml` has to be  
 117 provided, which specifies the corresponding atoms in the different representations. After this  
 118 step, all information (frame number, dimensions of the simulation box, etc) are stored in the state  
 119 file and only this file is used for further calculations.



#### Be careful!


| VOTCA-XTP requires a wrapped trajectory for mapping the segments and fragments, so all molecules should be whole in the frame.

120 In order to visually check the mapping one can use either the `tdump` calculator or the programm  
 sec:xtp\_dump `xtp_dump` with the calculator `trajectory2pdb`.

 **Writing a mapped trajectory with `xtp_dump`**  
 | `xtp_dump -f state.sql -e trajectory2pdb`

122 It reads in the state file created by `xtp_map` and outputs two trajectory files corresponding to  
 123 the original and rigidified atom coordinates. To check the mapping, it is useful to superimpose  
 124 the three outputs (original atomistic, atomistic stored in the state file, and rigidified according to  
 125 ground state geometries), e.g., with VMD.

sec:tdump

 **Writing a mapped trajectory with `tdump`**  
 | `xtp_run -f state.sql -o options.xml -e tdump`

126 It also reads in the state file but appends the coordinates to a `pdb` file. So make sure to delete old  
 127 `QM.pdb` and `MD.pdb` if you want to create a new imagef


## 128 2.4 Neighbor list

sec:neighborlist

129 A list of neighboring conjugated segments, or neighbor list, contains all pairs of conjugated seg-  
 130 ments for which `coupling elements`, `reorganization energies`, `site energy differences`, and `rates`  
 131 are evaluated.

132 Two segments are added to this list if the distance between centers of mass of any of their rigid  
 133 fragments is below a certain cutoff. This allows neighbors to be selected on a criterion of min-  
 134 imum distance of approach rather than center of mass distance, which is useful for molecules  
 135 with anisotropic shapes.

136 The neighbor list can be generated from the atomistic trajectory by using the `neighborlist`  
 137 `calculator`. This calculator requires a cutoff, which can be specified in the `options.xml` file. The  
 138 list is saved to the `state.sql` file:

 **Generating a neighbor list**  
 | `xtp_run -o options.xml -f state.sql -e neighborlist`

## 139 2.5 Reorganization energy

sec:reorganization

140 The reorganization energy  $\lambda_{ij}$  takes into account the change in nuclear (and dielectric) degrees of  
 141 freedom as the charge moves from donor  $i$  to acceptor  $j$ . It has two contributions: intramolecular,  
 142  $\lambda_{ij}^{\text{int}}$ , which is due to reorganization of nuclear coordinates of the two molecules forming the  
 143 charge transfer complex, and intermolecular (outersphere),  $\lambda_{ij}^{\text{out}}$ , which is due to the relaxation of  
 144 the nuclear coordinates of the environment. In what follows we discuss how these contributions  
 145 can be calculated.

### 146 2.5.1 Intramolecular reorganization energy

sec:intramolecular

147 If intramolecular vibrational modes of the two molecules are treated classically, the rearrange-  
 148 ment of their nuclear coordinates after charge transfer results in the dissipation of the internal  
 149 reorganization energy,  $\lambda_{ij}^{\text{int}}$ . It can be computed from four points on the potential energy surfaces  
 150 (PES) of both molecules in neutral and charged states, as indicated in figure 2.3.

151 Adding the contributions due to discharging of molecule  $i$  and charging of molecule  $j$  yields [8]

$$\lambda_{ij}^{\text{int}} = \lambda_i^{cn} + \lambda_j^{nc} = U_i^{nC} - U_i^{nN} + U_j^{cN} - U_j^{cC}. \quad (2.1) \quad \text{equ:lambda}$$

152 Here  $U_i^{nC}$  is the internal energy of the neutral molecule  $i$  in the geometry of its charged state  
 153 (small  $n$  denotes the state and capital  $C$  the geometry). Similarly,  $U_j^{cN}$  is the energy of the charged  
 154 molecule  $j$  in the geometry of its neutral state. Note that the PES of the donor and acceptor are  
 155 not identical for chemically different compounds or for conformers of the same molecule. In this

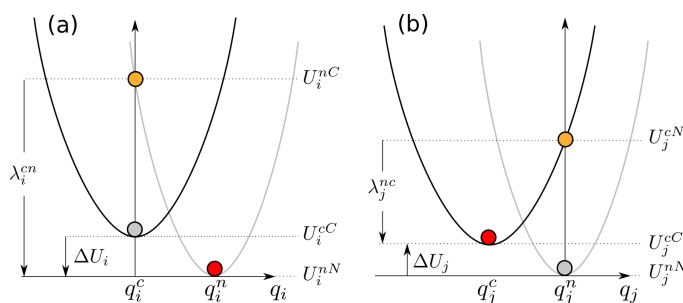


Figure 2.3: Potential energy surfaces of (a) donor and (b) acceptor in charged and neutral states. After the change of the charge state both molecules relax their nuclear coordinates. If all vibrational modes are treated classically, the total internal reorganization energy and the internal energy difference of the electron transfer reaction are  $\lambda_{ij}^{\text{int}} = \lambda_i^{\text{cn}} + \lambda_j^{\text{nc}}$  and  $\Delta E_{ij}^{\text{int}} = \Delta U_i - \Delta U_j$ , respectively.

fig.parabolas

156 case  $\lambda_i^{\text{cn}} \neq \lambda_j^{\text{cn}}$  and  $\lambda_i^{\text{nc}} \neq \lambda_j^{\text{nc}}$ . Thus  $\lambda_{ij}^{\text{int}}$  is a property of the charge transfer complex, and not of  
 157 a single molecule.

158 Intramolecular reorganization energies for discharging ( $\lambda^{\text{cn}}$ ) and charging ( $\lambda^{\text{nc}}$ ) of a molecule  
 159 need to be determined using quantum-chemistry and given in `map.xml`. The values are written  
 160 to the `state.sql` using the calculator `einternal` (see also internal energy):

```

  161 Intramolecular reorganization energies
  162 | xtp_run -o options.xml -f state.sql -e einternal
  
```

## 161 2.5.2 Outersphere reorganization energy

sec.outersphere

162 During the charge transfer reaction, also the molecules outside the charge transfer complex reori-  
 163 ent and polarize in order to adjust for changes in electric potential, resulting in the outersphere  
 164 contribution to the reorganization energy.  $\lambda_{ij}^{\text{out}}$  is particularly important if charge transfer occurs  
 165 in a polarizable environment. Assuming that charge transfer is much slower than electronic po-  
 166 larization but much faster than nuclear rearrangement of the environment,  $\lambda_{ij}^{\text{out}}$  can be calculated  
 167 from the electric displacement fields created by the charge transfer complex [9]

$$\lambda_{ij}^{\text{out}} = \frac{c_p}{2\epsilon_0} \int_{V^{\text{out}}} dV \left[ \vec{D}_I(\vec{r}) - \vec{D}_F(\vec{r}) \right]^2, \quad (2.2)$$

equ.lambda\_outer1

168 where  $\epsilon_0$  is the the permittivity of free space,  $\vec{D}_{I,F}(\vec{r})$  are the electric displacement fields created  
 169 by the charge transfer complex in the initial (charge on molecule  $i$ ) and final (charge transferred  
 170 to molecule  $j$ ) states,  $V^{\text{out}}$  is the volume outside the complex, and  $c_p = \frac{1}{\epsilon_{\text{opt}}} - \frac{1}{\epsilon_s}$  is the Pekar factor,  
 171 which is determined by the low ( $\epsilon_s$ ) and high ( $\epsilon_{\text{opt}}$ ) frequency dielectric permittivities.

172 Eq. (2.2) can be simplified by assuming spherically symmetric charge distributions on molecules  
 173  $i$  and  $j$  with total charge  $e$ . Integration over the volume  $V^{\text{out}}$  outside of the two spheres of radii  $R_i$   
 174 and  $R_j$  centered on molecules  $i$  and  $j$  leads to the classical Marcus expression for the outersphere  
 175 reorganization energy

$$\lambda_{ij}^{\text{out}} = \frac{c_p e^2}{4\pi\epsilon_0} \left( \frac{1}{2R_i} + \frac{1}{2R_j} - \frac{1}{r_{ij}} \right), \quad (2.3)$$

equ.lambda\_outer2

176 where  $r_{ij}$  is the molecular separation. While eq. (2.3) captures the main physics, e.g. predicts  
 177 smaller outer-sphere reorganization energies (higher rates) for molecules at smaller separations,  
 178 it often cannot provide quantitative estimates, since charge distributions are rarely spherically  
 179 symmetric.

180 Alternatively, the displacement fields can be constructed using the atomic partial charges. The  
 181 difference of the displacement fields at the position of an atom  $b_k$  outside the charge transfer  
 182 complex (molecule  $k \neq i, j$ ) can be expressed as

$$\vec{D}_I(\vec{r}_{b_k}) - \vec{D}_F(\vec{r}_{b_k}) = \sum_{a_i} \frac{q_{a_i}^c - q_{a_i}^n}{4\pi} \frac{(\vec{r}_{b_k} - \vec{r}_{a_i})}{|\vec{r}_{b_k} - \vec{r}_{a_i}|^3} + \sum_{a_j} \frac{q_{a_j}^n - q_{a_j}^c}{4\pi} \frac{(\vec{r}_{b_k} - \vec{r}_{a_j})}{|\vec{r}_{b_k} - \vec{r}_{a_j}|^3}, \quad (2.4)$$

183 where  $q_{a_i}^n$  ( $q_{a_i}^c$ ) is the partial charge of atom  $a$  of the neutral (charged) molecule  $i$  in vacuum. The  
 184 partial charges of neutral and charged molecules are obtained by fitting their values to reproduce  
 185 the electrostatic potential of a single molecule (charged or neutral) in vacuum. Assuming a uni-  
 186 form density of atoms, the integration in eq. (2.2) can be rewritten as a density-weighted sum  
 187 over all atoms excluding those of the charge transfer complex.

188 The remaining unknown needed to calculate  $\lambda_{ij}^{\text{out}}$  is the Pekar factor,  $c_p$ . In polar solvents  $\epsilon_s \gg$   
 189  $\epsilon_{\text{opt}} \sim 1$  and  $c_p$  is of the order of 1. In most organic semiconductors, however, molecular orien-  
 190 tations are fixed and therefore the low frequency dielectric permittivity is of the same order of  
 191 magnitude as  $\epsilon_{\text{opt}}$ . Hence,  $c_p$  is small and its value is very sensitive to differences in the permit-  
 192 tivities.

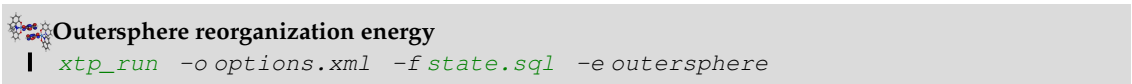
193 Outersphere reorganization energies for all pairs of molecules in the **neighbor list** can be com-  
 194 puted from the atomistic trajectory by using the **eoutersphere calculator**.

195 Two methods can be used to compute  $\lambda_{ij}^{\text{out}}$ . The first method uses the atomistic partial charges of  
 196 neutral and charged molecules from files specified in `map.xml` and eq. (2.2). The Pekar factor  $c_p$   
 197 and a cutoff radius based on molecular centers of mass have to be specified in the `options.xml`  
 198 file.

199 If this method is computationally prohibitive,  $\lambda_{ij}^{\text{out}}$  can be computed using eq. (2.3), which as-  
 200 sumes spherical charge distributions on the molecules. In this case the radii of these spheres are  
 201 specified in `segments.xml`, while the Pekar factor  $c_p$  is given in the `options.xml` file and no  
 202 cutoff radius is needed.

203 The outer sphere reorganization energies are saved to the `state.sql` file:

```


Outersphere reorganization energy
  

  | xtp_run -o options.xml -f state.sql -e outersphere

```

## 204 2.6 Site energies

sec:site\_energies

205 A charge transfer reaction between molecules  $i$  and  $j$  is driven by the site energy difference,  
 206  $\Delta E_{ij} = E_i - E_j$ . Since the transfer rate,  $\omega_{ij}$ , depends exponentially on  $\Delta E_{ij}$  (see eq. (2.31)) it is  
 207 important to compute its distribution as accurately as possible. The total site energy difference  
 208 has contributions due to **externally applied electric field**, electrostatic interactions, polarization  
 209 effects, and **internal energy** differences. In what follows we discuss how to estimate these contri-  
 210 butions by making use of first-principles calculations and polarizable force-fields.

### 211 2.6.1 Externally applied electric field

sec:ext\_field

212 The contribution to the total site energy difference due to an external electric field  $\vec{F}$  is given by  
 213  $\Delta E_{ij}^{\text{ext}} = q\vec{F} \cdot \vec{r}_{ij}$ , where  $q = \pm e$  is the charge and  $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$  is a vector connecting molecules  
 214  $i$  and  $j$ . For typical distances between small molecules, which are of the order of 1 nm, and  
 215 moderate fields of  $F < 10^8$  V/m this term is always smaller than 0.1 eV.



## 2.6.2 Internal energy

The contribution to the site energy difference due to different internal energies (see figure 2.3) can be written as

$$\Delta E_{ij}^{\text{int}} = \Delta U_i - \Delta U_j = (U_i^{cC} - U_i^{nN}) - (U_j^{cC} - U_j^{nN}), \quad (2.5)$$

where  $U_i^{cC(nN)}$  is the total energy of molecule  $i$  in the charged (neutral) state and geometry.  $\Delta U_i$  corresponds to the adiabatic ionization potential (or electron affinity) of molecule  $i$ , as shown in figure 2.3. For one-component systems and negligible conformational changes  $\Delta E_{ij}^{\text{int}} = 0$ , while it is significant for donor-acceptor systems.

Internal energies determined using quantum-chemistry need to be specified in `map.xml`. The values are written to the `state.sql` using the calculator `einternal` (see also `intramolecular reorganization energy`):

```

Internal energies
| xtp_run -o options.xml -f state.sql -e einternal

```

## 2.6.3 Electrostatic interaction energy

We represent the molecular charge density by choosing multiple expansion sites (“polar sites”) per molecule in such a way as to accurately reproduce the molecular electrostatic potential (ESP), with a set of suitably chosen multipole moments  $\{Q_{lk}^a\}$  (in spherical-tensor notation) allocated to each site. The expression for the electrostatic interaction energy between two molecules  $A$  and  $B$  in the multi-point expansion includes an implicit sum over expansion sites  $a \in A$  and  $b \in B$ ,

$$U_{AB} = \sum_{a \in A} \sum_{b \in B} \hat{Q}_{l_1 k_1}^a T_{l_1 k_1 l_2 k_2}^{a,b} \hat{Q}_{l_2 k_2}^b \equiv \hat{Q}_{l_1 k_1}^a T_{l_1 k_1 l_2 k_2}^{a,b} \hat{Q}_{l_2 k_2}^b, \quad (2.6)$$

where we have used the Einstein sum convention for the site indices  $a$  and  $b$  on the right-hand side of the equation, in addition to the sum convention that is in place for the multipole-moment components  $t \equiv l_1 k_1$  and  $u \equiv l_2 k_2$ . The  $T_{l_1 k_1 l_2 k_2}^{a,b}$  are tensors that mediate the interaction between a multipole component  $l_1 k_1$  on site  $a$  with the moment  $l_2 k_2$  on site  $b$ . If we include the molecular environment into a perturbative term  $W$  to enter in the single-molecule Hamiltonian, the above expression is exactly the first-order correction to the energy where the quantum-mechanical detail has been absorbed in classical multipole moments.

There are a number of strategies how to arrive at such a collection of *distributed multipoles*. They can be classified according to whether the multipoles are derived (a) from the electrostatic potential generated by the SCF charge density or (b) from a decomposition of the wavefunction itself. Here, we will only draft two of those approaches, CHELPG [10] from category (a) and DMA [11] from category (b).

The CHELPG (CHarges from ELEctrostatic Potentials, Grid-based) method relies on performing a least-squares fit of atom-placed charges to reproduce the electrostatic potential as evaluated from the SCF density on a regularly spaced grid [10]. The fitted charges result from minimizing the Lagrangian function [12]

$$z(\{q_i\}) = \sum_{k=1}^M \left( \phi(\vec{r}_k) - \sum_{i=1}^N \frac{1}{4\pi\epsilon_0} \frac{q_i}{|\vec{r}_i - \vec{r}_k|} \right) + \lambda \left( q_{\text{mol}} - \sum_{i=1}^N q_i \right), \quad (2.7)$$

with  $M$  grid points,  $N$  atomic sites, the set of atomic partial charges  $\{q_i\}$  and the SCF potential  $\phi$ . The Lagrange multiplier  $\lambda$  constrains the sum of the fitted charges to the molecular charge  $q_{\text{mol}}$ . The main difference from other fitting schemes [13] is the algorithm that selects the positions

242 at which the potential is evaluated (we note that the choice of grid points can have substan-  
 243 tial effects especially for bulky molecules). Clearly, the CHELPG method can be (and has been)  
 244 extended to include higher atomic multipoles. It should be noted, however, how already the in-  
 245 clusion of atomic dipoles hardly improves the parametrization, and can in fact be harmful to its  
 246 conformational stability.

The Distributed-Multipole-Analysis (DMA) approach [11, 14], developed by A. Stone, operates directly on the quantum-mechanical density matrix, expanded in terms of atom- and bond-centered Gaussian functions  $\chi_\alpha = R_{LK}(\vec{x} - \vec{s}_\alpha) \exp[-\zeta(\vec{x} - \vec{s}_\alpha)^2]$ ,

$$\rho(\vec{x}) = \sum_{\alpha,\beta} \rho_{\alpha\beta} \chi_\alpha(\vec{x} - \vec{s}_\alpha) \chi_\beta(\vec{x} - \vec{s}_\beta). \quad (2.8)$$

The aim is to compute multipole moments according in a distributed fashion: If we use that the overlap product  $\chi_\alpha \chi_\beta$  of two Gaussian basis functions yields itself a Gaussian centered at  $\vec{P} = (\zeta_\alpha \vec{s}_\alpha + \zeta_\beta \vec{s}_\beta) / (\zeta_\alpha + \zeta_\beta)$ , it is possible to proceed in two steps: First, we compute the multipole moments associated with a specific summand in the density matrix, referred to the overlap center  $\vec{P}$ :

$$Q_{LK}[\vec{P}] = - \int R_{LK}(\vec{x} - \vec{P}) \rho_{\alpha\beta} \chi_\alpha \chi_\beta d^3x. \quad (2.9)$$

Second, we transfer the resulting  $Q_{lk}[\vec{P}]$  to the position  $\vec{S}$  of a polar site according to the rule [11]

$$Q_{nm}[\vec{S}] = \sum_{l=0}^L \sum_{k=-l}^l \left[ \binom{n+m}{l+k} \binom{n-m}{l-k} \right]^{1/2} R_{n-l,m-k}(\vec{S} - \vec{P}) \cdot Q_{lk}[\vec{P}]. \quad (2.10)$$

247 Note how this requires a rule for the choice of the expansion site to which the multipole moment  
 248 should be transferred. In the near past [14], the nearest-site algorithm, which allocates the mul-  
 249 tipole moments to the site closest to the overlap center, was replaced for diffuse functions by an  
 250 algorithm based on a sxtpth weighting function in conjunction with grid-based integration meth-  
 251 ods in order to decrease the basis-set dependence of the resulting set of distributed multipoles.  
 252 One important advantage of the DMA approach over fitting algorithms such as CHELPG or  
 253 Merz-Kollman (MK) is that higher-order moments can also be derived without too large an am-  
 254 biguity.

255 The ‘mps’ file format used by VOTCA for the definition of distributed multipoles (as well as  
 256 point polarizabilities, see subsequent section) is based on the GDMA punch format of A. Stone’s  
 257 GDMA program [14] (the punch output file can be immediately plugged into VOTCA without  
 258 any conversions to be applied). Furthermore the log-file of different QM packages (currently  
 259 Gaussian, Turbomole and NWChem) may be fed into the `log2mps` tool, which will subse-  
 260 quently generate the appropriate mps-file.



Read in ESP charges from a QM log file

```
| xtp_tools -o options.xml -e log2mps
```

## 261 2.6.4 Induction energy - the Thole model

sec:thole\_model

If we in addition to the permanent set of multipole moments  $\{Q_t^a\}$  allow for induced moments  $\{\Delta Q_t^a\}$  and penalize their generation with a bilinear form (giving rise to a strictly positive contribution to the energy),

$$U_{\text{int}} = \frac{1}{2} \sum_A \Delta Q_t^a \eta_{tt'}^{aa'} \Delta Q_{t'}^{a'}, \quad (2.11)$$

it can be shown that the induction contribution to the site energy evaluates to an expression where all interactions between induced moments have cancelled out, and interactions between permanent and induced moments are scaled down by 1/2 [15]:

$$U_{pu} = \frac{1}{2} \sum_A \sum_{B>A} [\Delta Q_t^a T_{tu}^{ab} Q_u^b + \Delta Q_t^b T_{tu}^{ab} Q_u^a]. \quad (2.12) \quad \text{equ:u\_pu}$$

This term can be viewed as the second-order (induction) correction to the molecular interaction energy. The sets of  $\{Q_t^a\}$  are solved for self-consistently via

$$\Delta Q_t^a = - \sum_{B \neq A} \alpha_{tt'}^{aa'} T_{t'u}^{a'b} (Q_u^b + \Delta Q_u^b), \quad (2.13) \quad \text{equ:self\_consistent\_dQ}$$

262 where the polarizability tensors  $\alpha_{tt'}^{aa'}$  are given by the inverse of  $\eta_{tt'}^{aa'}$ .  
 263 With eqs. 2.13 and 2.12 we have at hand expressions that allow us to compute the induction en-  
 264 ergy contribution to site energies in an iterative manner based on a set of molecular distributed  
 265 multipoles  $\{Q_t^a\}$  and polarizabilities  $\{\alpha_{tt'}^{aa'}\}$ . We have drafted in the previous section how to ob-  
 266 tain the former from a wavefunction decomposition or fitting scheme (GDMA, CHELPG). The  
 267  $\{\alpha_{tt'}^{aa'}\}$  can be derived formally (or rather: read off) from a perturbative expansion of the molec-  
 268 ular interaction. In this work we make use of the Thole model [16, 17] as a semi-empirical ap-  
 269 proach to obtain the sought-after point polarizabilities in the local dipole approximation, that is,  
 270  $[\alpha_{tt'}^{aa'}] = \alpha_{tt'}^{aa'} \delta_{t\beta} \delta_{t'\beta} \delta_{aa'}$ , where  $\beta \in \{x, y, z\}$  references the dipole-moment component.  
 271 The Thole model is based on a modified dipole-dipole interaction, which can be reformulated in  
 272 terms of the interaction of smeared charge densities. This has been shown to be necessary due  
 273 to the divergent head-to-tail dipole-dipole interaction that otherwise results at small intersepara-  
 274 tions on the Å scale [16–18]. Smearing out the charge distribution mimics the nature of the QM  
 275 wavefunction, which effectively guards against this unphysical polarization catastrophe. Since  
 276 the point dipoles however only react individually to the external field, any correlation effects as  
 277 were still accounted for in the  $\{\alpha_{tt'}^{aa'}\}$  are lost, except perhaps those correlations that are due to  
 278 the mere classical field interaction.

The smearing of the nuclei-centered multipole moments is obtained via a fractional charge den-  
 sity  $\rho_f(\vec{u})$  which should be normalized to unity and fall off rapidly as of a certain radius  $\vec{u} = \vec{u}(\vec{R})$ .  
 The latter is related to the physical distance vector  $\vec{R}$  connecting two interacting sites via a linear  
 scaling factor that takes into account the magnitude of the isotropic site polarizabilities  $\alpha^a$ . This  
 isotropic fractional charge density gives rise to a modified potential

$$\phi(u) = - \frac{1}{4\pi\epsilon_0} \int_0^u 4\pi u' \rho(u') du' \quad (2.14) \quad \text{equ:mod\_potential}$$

We can relate the multipole interaction tensor  $T_{ij\dots}$  (this time in Cartesian coordinates) to the  
 fractional charge density in two steps: First, we rewrite the tensor in terms of the scaled distance  
 vector  $\vec{u}$ ,

$$T_{ij\dots}(\vec{R}) = f(\alpha^a \alpha^b) t_{ij\dots}(\vec{u}(\vec{R}), \alpha^a \alpha^b), \quad (2.15)$$

where the specific form of  $f(\alpha^a \alpha^b)$  results from the choice of  $u(\vec{R}, \alpha^a \alpha^b)$ . Second, we demand  
 that the smeared interaction tensor  $t_{ij\dots}$  is given as usual by the appropriate derivative of the  
 potential in eq. 2.14,

$$t_{ij\dots}(\vec{u}) = -\partial_{u_i} \partial_{u_j} \dots \phi(\vec{u}). \quad (2.16)$$

279 It turns out that for a suitable choice of  $\rho_f(\vec{u})$ , the modified interaction tensors can be rewritten  
 280 in such a way that powers  $n$  of the distance  $R = |\vec{R}|$  are damped with a damping function  
 281  $\lambda_n(\vec{u}(\vec{R}))$  [19].

282 There is a large number of fractional charge densities  $\rho_f(\vec{u})$  that have been tested for the purpose  
 283 of giving best results for the molecular polarizability as well as interaction energies. Note how a  
 284 great advantage of the Thole model is the exceptional transferability of the atomic polarizabilities  
 285 to compounds not used for the fitting procedure [17]. In fact, for most organic molecules, a fixed  
 286 set of atomic polarizabilities ( $\alpha_C = 1.334$ ,  $\alpha_H = 0.496$ ,  $\alpha_N = 1.073$ ,  $\alpha_O = 0.873$ ,  $\alpha_S = 2.926 \text{ \AA}^3$ )  
 287 based on atomic elements yields satisfactory results.

VOTCA implements the Thole model with an exponentially-decaying fractional charge density

$$\rho(u) = \frac{3a}{4\pi} \exp(-au^3), \quad (2.17)$$

288 where  $\vec{u}(\vec{R}, \alpha^a \alpha^b) = \vec{R}/(\alpha^a \alpha^b)^{1/6}$  and the smearing exponent  $a = 0.39$  (which can however be  
 289 changed from the program options), as used in the AMOEBA force field [19].

290 Even though the Thole model performs very well for many organic compounds with only the  
 291 above small set of element-based polarizabilities, conjugated molecules may require a more in-  
 292 tricate parametrization. The simplest approach is to resort to scaled polarizabilities to match  
 293 the effective molecular polarizable volume  $V \sim \alpha_x \alpha_y \alpha_z$  as predicted by QM calculations (here  
 294  $\alpha_x, \alpha_y, \alpha_z$  are the eigenvalues of the molecular polarizability tensor). The `molpol` tool assists  
 295 with this task, it self-consistently calculates the Thole polarizability for an input `mps`-file and  
 296 optimizes (if desired) the atomic polarizabilities in the above simple manner.

#### Generate Thole-type polarizabilities for a segment

```
| xtp_tools -o options.xml -e molpol
```

297 The electrostatic and induction contribution to the site energy is evaluated by the `emultipole`  
 298 `calculator`. Atomistic partial charges for charged and neutral molecules are taken from `mps`-files  
 299 (extended GDMA format) specified in `map.xml`. Note that, in order to speed up calculations for  
 300 both methods, a cut-off radius (for the molecular centers of mass) can be given in `options.xml`.  
 301 Threaded execution is advised.

#### Electrostatic and induction corrections

```
| xtp_run -o options.xml -f state.sql -e emultipole
```

302 Furthermore available are `zmultipole`, which extends `emultipole` to allow for an electrostatic  
 303 buffer layer (loosely related to the z-buffer in OpenGL, hence the name) and anisotropic point  
 304 polarizabilities. For the interaction energy of charged clusters of any user-defined composition  
 305 (Frenkel states, CT states, ...), `xqmultipole` can be used.

#### Interaction energy of charged molecular clusters embedded in a molecular environment

```
| xtp_parallel -o options.xml -f state.sql -e xqmultipole
```

## 306 2.7 Transfer integrals

sec:transfer\_integrals

307 The electronic transfer integral element  $J_{ij}$  entering the Marcus rates in eq. (2.31) is defined as

$$J_{ij} = \langle \phi_i | \hat{H} | \phi_j \rangle, \quad (2.18) \quad \text{equ:TI}$$

308 where  $\phi_i$  and  $\phi_j$  are diabatic wavefunctions, localized on molecule  $i$  and  $j$  respectively, partici-  
 309 pating in the charge transfer, and  $\hat{H}$  is the Hamiltonian of the formed dimer. Within the frozen-  
 310 core approximation, the usual choice for the diabatic wavefunctions  $\phi_i$  is the highest occupied  
 311 molecular orbital (HOMO) in case of hole transport, and the lowest unoccupied molecular or-  
 312 bital (LUMO) in the case of electron transfer, while  $\hat{H}$  is an effective single particle Hamiltonian,

313 e.g. Fock or Kohn-Sham operator of the dimer. As such,  $J_{ij}$  is a measure of the strength of the  
 314 electronic coupling of the frontier orbitals of monomers mediated by the dimer interactions.  
 315 Intrinsically, the transfer integral is very sensitive to the molecular arrangement, i.e. the dis-  
 316 tance and the mutual orientation of the molecules participating in charge transport. Since this  
 317 arrangement can also be significantly influenced by static and/or dynamic disorder [20–24], it is  
 318 essential to calculate  $J_{ij}$  explicitly for each hopping pair within a realistic morphology. Consid-  
 319 ering that the number of dimers for which eq. (2.18) has to be evaluated is proportional to the  
 320 number of molecules times their coordination number, computationally efficient and at the same  
 321 time quantitatively reliable schemes are required.

### 322 2.7.1 Projection of monomer orbitals on dimer orbitals (DIPRO)

sec:dipro

323 An approach for the determination of the transfer integral that can be used for any single-particle  
 324 electronic structure method (Hartree-Fock, DFT, or semiempirical methods) is based on the pro-  
 325 jection of monomer orbitals on a manifold of explicitly calculated dimer orbitals. This dimer  
 326 projection (DIPRO) technique including an assessment of computational parameters such as  
 327 the basis set, exchange-correlation functionals, and convergence criteria is presented in detail  
 328 in ref. [25]. A brief summary of the concept is given below.

329 We start from an effective Hamiltonian <sup>1</sup>

$$\hat{H}^{\text{eff}} = \sum_i \epsilon_i \hat{a}_i^\dagger \hat{a}_i + \sum_{j \neq i} J_{ij} \hat{a}_i^\dagger \hat{a}_j + c.c. \quad (2.19) \quad \text{equ:dipro_eq1}$$

330 where  $\hat{a}_i^\dagger$  and  $\hat{a}_i$  are the creation and annihilation operators for a charge carrier located at the  
 331 molecular site  $i$ . The electron site energy is given by  $\epsilon_i$ , while  $J_{ij}$  is the transfer integral between  
 332 two sites  $i$  and  $j$ . We label their frontier orbitals (HOMO for hole transfer, LUMO for electron  
 333 transfer)  $\phi_i$  and  $\phi_j$ , respectively. Assuming that the frontier orbitals of a dimer (adiabatic energy  
 334 surfaces) result exclusively from the interaction of the frontier orbitals of monomers, and conse-  
 335 quently expand them in terms of  $\phi_i$  and  $\phi_j$ . The expansion coefficients,  $\bar{\mathbf{C}}$ , can be determined by  
 336 solving the secular equation

$$(\mathbf{H} - E\mathbf{S})\bar{\mathbf{C}} = 0 \quad (2.20) \quad \text{equ:dipro_eq2}$$

337 where  $\mathbf{H}$  and  $\mathbf{S}$  are the Hamiltonian and overlap matrices of the system, respectively. These  
 338 matrices can be written explicitly as

$$\mathbf{H} = \begin{pmatrix} e_i & H_{ij} \\ H_{ij}^* & e_j \end{pmatrix} \quad \mathbf{S} = \begin{pmatrix} 1 & S_{ij} \\ S_{ij}^* & 1 \end{pmatrix} \quad (2.21) \quad \text{equ:dipro_eq3}$$

339 with

$$\begin{aligned} e_i &= \langle \phi_i | \hat{H} | \phi_i \rangle & H_{ij} &= \langle \phi_i | \hat{H} | \phi_j \rangle \\ e_j &= \langle \phi_j | \hat{H} | \phi_j \rangle & S_{ij} &= \langle \phi_j | \phi_j \rangle \end{aligned} \quad (2.22) \quad \text{equ:dipro_eq4}$$

340 The matrix elements  $e_{i(j)}$ ,  $H_{ij}$ , and  $S_{ij}$  entering eq. (2.21) can be calculated via projections on the  
 341 dimer orbitals (eigenfunctions of  $\hat{H}$ )  $\{|\phi_n^{\text{D}}\rangle\}$  by inserting  $\hat{1} = \sum_n |\phi_n^{\text{D}}\rangle \langle \phi_n^{\text{D}}|$  twice. We exemplify  
 342 this explicitly for  $H_{ij}$  in the following

$$H_{ij} = \sum_{nm} \langle \phi_i | \phi_n^{\text{D}} \rangle \langle \phi_n^{\text{D}} | \hat{H} | \phi_m^{\text{D}} \rangle \langle \phi_m^{\text{D}} | \phi_j \rangle. \quad (2.23) \quad \text{equ:dipro_eq16}$$

343 The Hamiltonian is diagonal in its eigenfunctions,  $\langle \phi_n^{\text{D}} | \hat{H} | \phi_m^{\text{D}} \rangle = E_n \delta_{nm}$ . Collecting the projec-  
 344 tions of the frontier orbitals  $|\phi_{i(j)}\rangle$  on the  $n$ -th dimer state  $(\bar{\mathbf{V}}_{(i)})_n = \langle \phi_i | \phi_n^{\text{D}} \rangle$  and  $(\bar{\mathbf{V}}_{(j)})_n =$   
 345  $\langle \phi_j | \phi_n^{\text{D}} \rangle$  respectively, into vectors we obtain

$$H_{ij} = \bar{\mathbf{V}}_{(i)} \mathbf{E} \bar{\mathbf{V}}_{(j)}^\dagger. \quad (2.24) \quad \text{equ:dipro_eq17}$$

<sup>1</sup>we use following notations:  $a$  - number,  $\bar{\mathbf{a}}$  - vector,  $\mathbf{A}$  - matrix,  $\hat{A}$  - operator

346 What is left to do is determine these projections  $\bar{\mathbf{V}}_{(k)}$ . In all practical calculations the molecular  
 347 orbitals are expanded in basis sets of either plane waves or of localized atomic orbitals  $|\varphi_\alpha\rangle$ .  
 348 We will first consider the case that the calculations for the monomers are performed using a  
 349 counterpoise basis set that is commonly used to deal with the basis set superposition error (BSSE).  
 350 The basis set of atom-centered orbitals of a monomer is extended to the one of the dimer by  
 351 adding the respective atomic orbitals at virtual coordinates of the second monomer. We can then  
 352 write the respective expansions as

$$|\phi_k\rangle = \sum_{\alpha} \lambda_{\alpha}^{(k)} |\varphi_{\alpha}\rangle \quad \text{and} \quad |\phi_n^{\text{D}}\rangle = \sum_{\alpha} D_{\alpha}^{(n)} |\varphi_{\alpha}\rangle \quad (2.25) \quad \text{eq:dipro_eq18}$$

353 where  $k = i, j$ . The projections can then be determined within this common basis set as

$$(\bar{\mathbf{V}}_k)_n = \langle \phi_k | \phi_n^{\text{D}} \rangle = \sum_{\alpha} \lambda_{\alpha}^{(k)} \langle \alpha | \sum_{\beta} D_{\beta}^{(n)} |\beta\rangle = \bar{\lambda}_{(k)}^{\dagger} \mathcal{S} \bar{\mathbf{D}}_{(n)} \quad (2.26) \quad \text{eq:dipro_eq19}$$

354 where  $\mathcal{S}$  is the overlap matrix of the atomic basis functions. This allows us to finally write the  
 355 elements of the Hamiltonian and overlap matrices in eq. (2.21) as:

$$\begin{aligned} H_{ij} &= \bar{\lambda}_{(i)}^{\dagger} \mathcal{S} \mathbf{D} \mathbf{E} \mathbf{D}^{\dagger} \mathcal{S}^{\dagger} \bar{\lambda}_{(j)} \\ S_{ij} &= \bar{\lambda}_{(i)}^{\dagger} \mathcal{S} \mathbf{D} \mathbf{D}^{\dagger} \mathcal{S}^{\dagger} \bar{\lambda}_{(j)} \end{aligned} \quad (2.27) \quad \text{eq:dipro_eq20}$$

356 Since the two monomer frontier orbitals that form the basis of this expansion are not orthogonal  
 357 in general ( $\mathbf{S} \neq \mathbf{1}$ ), it is necessary to transform eq. (2.20) into a standard eigenvalue problem of  
 358 the form

$$\mathbf{H}^{\text{eff}} \bar{\mathbf{C}}^{\text{eff}} = E \bar{\mathbf{C}}^{\text{eff}} \quad (2.28) \quad \text{eq:dipro_eq7}$$

359 to make it correspond to eq. (2.19). According to Löwdin such a transformation can be achieved  
 360 by

$$\mathbf{H}^{\text{eff}} = \mathbf{S}^{-1/2} \mathbf{H} \mathbf{S}^{-1/2}. \quad (2.29) \quad \text{eq:dipro_eq9}$$

361 This then yields an effective Hamiltonian matrix in an orthogonal basis, and its entries can di-  
 362 rectly be identified with the site energies  $\epsilon_i$  and transfer integrals  $J_{ij}$ :

$$\mathbf{H}^{\text{eff}} = \begin{pmatrix} e_i^{\text{eff}} & H_{ij}^{\text{eff}} \\ H_{ij}^{*,\text{eff}} & e_j^{\text{eff}} \end{pmatrix} = \begin{pmatrix} \epsilon_i & J_{ij} \\ J_{ij}^* & \epsilon_j \end{pmatrix} \quad (2.30) \quad \text{eq:dipro_eq11}$$

## 363 2.7.2 DFT-based transfer integrals using DIPRO

sec:dft

364 The calculation of one electronic coupling element based on DFT using the DIPRO method re-  
 365 quires the overlap matrix of atomic orbitals  $\mathcal{S}$ , the expansion coefficients for monomer  $\bar{\lambda}_{(k)} =$   
 366  $\{\lambda_{\alpha}^{(k)}\}$  and dimer orbitals  $\bar{\mathbf{D}}_{(n)} = \{D_{\alpha}^{(n)}\}$ , as well as the orbital energies  $E_n$  of the dimer are  
 367 required as input. In practical situations, performing self-consistent quantum-chemical calcula-  
 368 tions for each individual monomer and one for the dimer to obtain this input data is extremely  
 369 demanding. Several simplifications can be made to reduce the computational effort, such as  
 370 using non-Counterpoise basis sets for the monomers (thereby decoupling the monomer calcula-  
 371 tions from the dimer run) and performing only a single SCF step in a dimer calculation starting  
 372 from an initial guess formed from a superposition of monomer orbitals. This "noCP+noSCF"  
 373 variant of DIPRO is shown in figure 2.4(a) and recommended for production runs. A detailed  
 374 comparative study of the different variants can be found in [25].

375 The code currently contains supports evaluation of transfer integrals from quantum-chemical  
 376 calculations performed with the Gaussian, Turbomole, and NWChem packages. The interfac-  
 377 ing procedure consists of three main steps: generation of input files for monomers and dimers,  
 378 performing the actual quantum-chemical calculations, and calculating the transfer integrals.

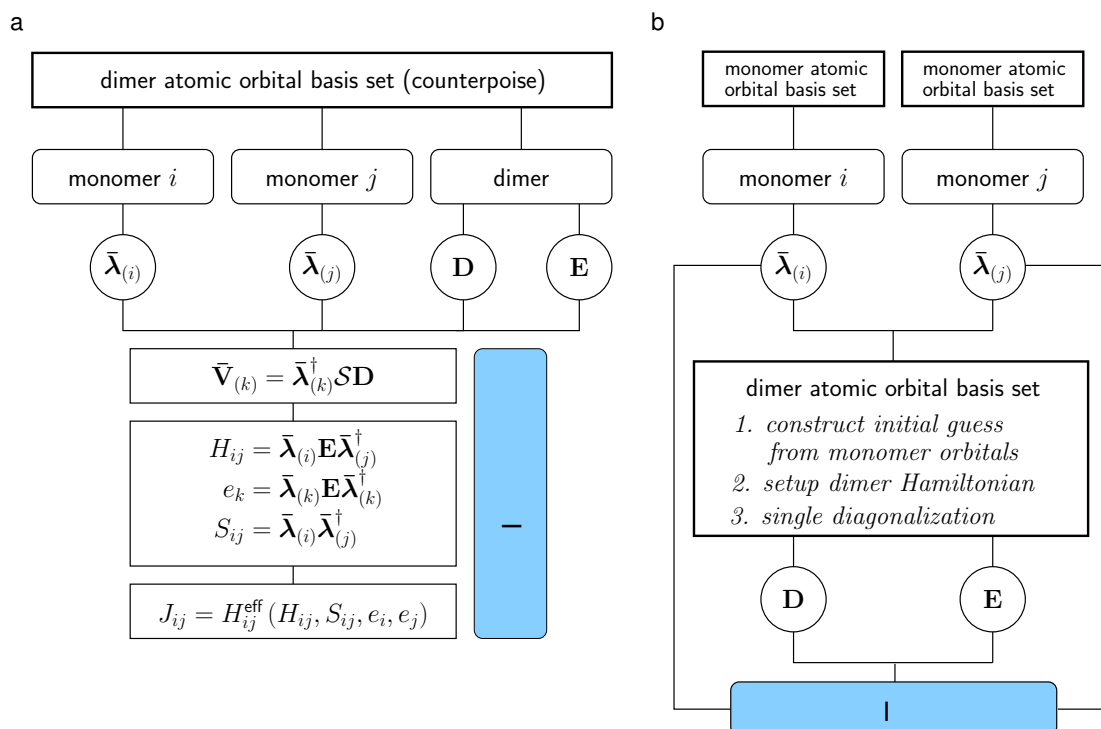


Figure 2.4: Schematics of the DIPRO method. (a) General workflow of the projection technique. (b) Strategy of the efficient noCP+noSCF implementation, in which the monomer calculations are performed independently form the dimer configurations (noCP), using the `edft` calculator. The dimer Hamiltonian is subsequently constructed based on an initial guess formed from monomer orbitals and only diagonalized once (noSCF) before the transfer integral is calculated by projection. This second step is performed by the `idft` calculator.

fig.dipro\_scheme


### 379 Monomer calculations

sec:edft

380 First, `hopping sites` and a `neighbor list` need to be generated from the atomistic topology and  
 381 trajectory and written to the `state.sql` file. Then the parallel `edft` calculator manages the  
 382 calculation of the monomer properties required for the determination of electronic coupling ele-  
 383 ments. Specifically, the individual steps it performs are:

- 384 1. Creation of a job file containing the list of molecules to be calculated with DFT


```

385  Writing job file for edft
| xtp_parallel -o options.xml -f state.sql -e edft -jwrite

```

- 385 2. Running of all jobs in job file

```

386  Running all edft jobs
| xtp_parallel -o options.xml -f state.sql -e edft -jrun

```

386 which includes

- 387 • creating the input files for the DFT calculation (using the package specified in `options.xml`)
- 388 in the directory

389 OR\_FILES/package/frame\_F/mol\_M

390 where F is the index of the frame in the trajectory, M is the index of a molecule in this  
391 frame,

- 392 • executing the DFT run, and
- 393 • after completion of this run, parsing the output (number of electrons, basis set, molec-  
394 ular orbital expansion coefficients), and saving it in compressed form to

395 OR\_FILES/molecules/frame\_F/molecule\_M.orb


### 396 Calculating the transfer integrals

sec:tdft

397 After the monomer calculations have been completed successfully, the respective runs for dimers  
398 from the neighborlist can be performed using the parallel **idft calculator**, which manages the  
399 DFT runs for the hopping pairs and determines the coupling element using DIPRO. Again, sev-  
400 eral steps are required:

- 401 1. Creation of a job file containing the list of pairs to be calculated with DFT


```

 Writing job file for idft
| xtp_parallel -o options.xml -f state.sql -e idft -jwrite

```

- 402 2. Running of all jobs in job file

```

 Running all idft jobs
| xtp_parallel -o options.xml -f state.sql -e idft -jrun

```

403 which includes

- 404 • creating the input files (including the merged guess for a noSCF calculation, if re-  
405 quested) for the DFT calculation (using the package specified in `options.xml`) in the  
406 directory

407 OR\_FILES/package/frame\_F/pair\_M\_N

408 where M and N are the indices of the molecules in this pair,

- 409 • executing the DFT run, and
- 410 • after completion of this run, parsing the output (number of electrons, basis set, molec-  
411 ular orbital expansion coefficients and energies, atomic orbital overlap matrix), and  
412 saving the pair information in compressed form to


413 OR\_FILES/pairs/frame\_F/pair\_M\_N.orb

- 414 • loading the monomer orbitals from the previously saved `*.orb` files.
- 415 • calculating the coupling elements and write them to the job file

- 416 3. Reading the coupling elements from the job file and saving them to the `state.sql` file

417

```

 Saving idft results from job file to state.sql
| xtp_parallel -o options.xml -f state.sql -e idft -jread

```



### 2.7.3 ZINDO-based transfer integrals using MOO

An approximate method based on Zerner's Intermediate Neglect of Differential Overlap (ZINDO) has been described in Ref. [26]. This semiempirical method is substantially faster than first-principles approaches, since it avoids the self-consistent calculations on each individual monomer and dimer. This allows to construct the matrix elements of the ZINDO Hamiltonian of the dimer from the weighted overlap of molecular orbitals of the two monomers. Together with the introduction of rigid segments, only a single self-consistent calculation on one isolated conjugated segment is required. All relevant molecular overlaps can then be constructed from the obtained molecular orbitals.

The main advantage of the molecular orbital overlap (MOO) library is *fast* evaluation of electronic coupling elements. Note that MOO is based on the ZINDO Hamiltonian which has limited applicability. The general advice is to first compare the accuracy of the MOO method to the DFT-based calculations.

MOO can be used both in a **standalone mode** and as an **izindo calculator** of VOTCA-XTP.

Since MOO constructs the Fock operator of a dimer from the molecular orbitals of monomers by translating and rotating the orbitals of **rigid fragments**, the optimized geometry of all **conjugated segments** and the coefficients of the molecular orbitals are required as its input in addition to the state file (`state.sql`) with the **neighbor list**. Coordinates are stored in `geometry.xyz` files with four columns, first being the atom type and the next three atom coordinates. This is a standard `xyz` format without a header. Note that the atom order in the `geometry.xyz` files can be different from that of the mapping files. The correspondence between the two is established in the `map.xml` file.



#### Be careful!

Izindo requires the specification of orbitals for hole and electron transport in `map.xml`. They are the HOMO and LUMO respectively and can be retrieved from the `log` file from which the `zindo.orb` file is generated. The number of alpha electrons is the HOMO, the LUMO is HOMO+1

The calculated transfer integrals are immediately saved to the `state.sql` file.



#### Transfer integrals from izindo

```
| xtp_run -o options.xml -f state.sql -e izindo
```

## 2.8 Charge transfer rate

Charge transfer rates can be postulated based on intuitive physical considerations, as it is done in the Gaussian disorder models [20, 27–29]. Alternatively, charge transfer theories can be used to evaluate rates from quantum chemical calculations [1, 8, 25, 30–32]. In spite of being significantly more computationally demanding, the latter approach allows to link the chemical and electronic structure, as well as the morphology, to charge dynamics.

### 2.8.1 Classical charge transfer rate

The high temperature limit of classical charge transfer theory [33, 34] is often used as a trade-off between theoretical rigor and computational complexity. It captures key parameters which influence charge transport while at the same time providing an analytical expression for the rate. Within this limit, the transfer rate for a charge to hop from a site  $i$  to a site  $j$  reads

$$\omega_{ij} = \frac{2\pi}{\hbar} \frac{J_{ij}^2}{\sqrt{4\pi\lambda_{ij}k_B T}} \exp\left[-\frac{(\Delta E_{ij} - \lambda_{ij})^2}{4\lambda_{ij}k_B T}\right], \quad (2.31) \quad \text{equ:marcus}$$

where  $T$  is the temperature,  $\lambda_{ij} = \lambda_{ij}^{\text{int}} + \lambda_{ij}^{\text{out}}$  is the **reorganization energy**, which is a sum of intra- and inter-molecular (outersphere) contributions,  $\Delta E_{ij}$  is the **site-energy difference**, or driving force, and  $J_{ij}$  is the **electronic coupling element**, or transfer integral.

## 2.8.2 Semi-classical bimolecular rate

The main assumptions in eq. (2.31) are non-adiabaticity (small electronic coupling and charge transfer between two diabatic, non-interacting states), and harmonic promoting modes, which are treated classically. At ambient conditions, however, the intramolecular promoting mode, which roughly corresponds to C-C bond stretching, has a vibrational energy of  $\hbar\omega \approx 0.2 \text{ eV} \gg k_{\text{B}}T$  and should be treated quantum-mechanically. The outer-sphere (slow) mode has much lower vibrational energy than the intramolecular promoting mode, and therefore can be treated classically. The weak interaction between molecules also implies that each molecule has its own, practically independent, set of quantum mechanical degrees of freedom.

A more general, quantum-classical expression for a bimolecular multi-channel rate is derived in the Supporting Information of ref. [1] and has the following form

$$\omega_{ij} = \frac{2\pi}{\hbar} \frac{|J_{ij}|^2}{\sqrt{4\pi\lambda_{ij}^{\text{out}}k_{\text{B}}T}} \sum_{l',m'=0}^{\infty} |\langle \chi_{i0}^c | \chi_{il'}^n \rangle|^2 |\langle \chi_{j0}^n | \chi_{jm'}^c \rangle|^2 \exp \left\{ -\frac{[\Delta E_{ij} - \hbar(l'\omega_i^n + m'\omega_j^c) - \lambda_{ij}^{\text{out}}]^2}{4\lambda_{ij}^{\text{out}}k_{\text{B}}T} \right\}. \quad (2.32)$$

If the curvatures of intramolecular PES of charged and neutral states of a molecule are different, that is  $\omega_i^c \neq \omega_i^n$ , the corresponding reorganization energies,  $\lambda_i^{cn} = \frac{1}{2}[\omega_i^n(q_i^n - q_i^c)]^2$  and  $\lambda_i^{nc} = \frac{1}{2}[\omega_i^c(q_i^n - q_i^c)]^2$ , will also differ. In this case the Franck-Condon (FC) factors for discharging of molecule  $i$  read [35]

$$|\langle \chi_{i0}^c | \chi_{il'}^n \rangle|^2 = \frac{2}{2^{l'}l'!} \frac{\sqrt{\omega_i^c\omega_i^n}}{(\omega_i^c + \omega_i^n)} \exp(-|s_i|) \left[ \sum_{\substack{k=0 \\ k \text{ even}}}^{l'} \binom{l'}{k} \left( \frac{2\omega_i^c}{\omega_i^c + \omega_i^n} \right)^{k/2} \frac{k!}{(k/2)!} H_{l'-k} \left( \frac{s_i}{\sqrt{2S_i^{cn}}} \right) \right]^2, \quad (2.33)$$

where  $H_n(x)$  is a Hermite polynomial,  $s_i = 2\sqrt{\lambda_i^{nc}\lambda_i^{cn}}/\hbar(\omega_i^c + \omega_i^n)$ , and  $S_i^{cn} = \lambda_i^{cn}/\hbar\omega_i^c$ . The FC factors for charging of molecule  $j$  can be obtained by substituting  $(s_i, S_i^{cn}, \omega_i^c)$  with  $(-s_j, S_j^{nc}, \omega_j^n)$ . In order to evaluate the FC factors, the **internal reorganization energy**  $\lambda_i^{cn}$  can be computed from the intramolecular PES.

## 2.8.3 Semi-classical rate


One can also use the quantum-classical rate with a common set of vibrational coordinates [9]

$$\omega_{ij} = \frac{2\pi}{\hbar} \frac{|J_{ij}|^2}{\sqrt{4\pi\lambda_{ij}^{\text{out}}k_{\text{B}}T}} \sum_{N=0}^{\infty} \frac{1}{N!} \left( \frac{\lambda_{ij}^{\text{int}}}{\hbar\omega^{\text{int}}} \right)^N \exp \left( -\frac{\lambda_{ij}^{\text{int}}}{\hbar\omega^{\text{int}}} \right) \exp \left\{ -\frac{[\Delta E_{ij} - \hbar N\omega^{\text{int}} - \lambda_{ij}^{\text{out}}]^2}{4\lambda_{ij}^{\text{out}}k_{\text{B}}T} \right\}. \quad (2.34)$$

Numerical estimates show that if  $\lambda_{ij}^{\text{int}} \approx \lambda_{ij}^{\text{out}}$  and  $|\Delta E_{ij}| \ll \lambda_{ij}^{\text{out}}$  the rates are similar to those of eq. (2.31). In general, there is no robust method to compute  $\lambda_{ij}^{\text{out}}$  [36] and both reorganization energies are often assumed to be of the same order of magnitude. In this case the second condition also holds, unless there are large differences in electron affinities or ionization potentials of neighboring molecules, e.g. in donor-acceptor blends.

476 To calculate rates of the type specified in `options.xml` for all pairs in the `neighbor list` and to  
 477 save them into the `state.sql` file, run the `rates calculator`. Note that all required ingredi-  
 478 ents (`reorganization energies`, `transfer integrals`, and `site energies` have to be calculated before).  
 479

```

  480  Calculation of transfer rates
  481 | xtp_run -o options.xml -f state.sql -e rates
  
```

## 480 2.9 Master equation

sec:kmc

481 Having determined the list of conjugated segments (hopping sites) and charge transfer rates be-  
 482 tween them, the next task is to solve the master equation which describes the time evolution of  
 483 the system


$$\frac{\partial P_\alpha}{\partial t} = \sum_\beta P_\beta \Omega_{\beta\alpha} - \sum_\beta P_\alpha \Omega_{\alpha\beta}, \quad (2.35) \quad \text{equ:master}$$

484 where  $P_\alpha$  is the probability of the system to be in a state  $\alpha$  at time  $t$  and  $\Omega_{\alpha\beta}$  is the transition rate  
 485 from state  $\alpha$  to state  $\beta$ . A state  $\alpha$  is specified by a set of site occupations,  $\{\alpha_i\}$ , where  $\alpha_i = 1(0)$   
 486 for an occupied (unoccupied) site  $i$ , and the matrix  $\hat{\Omega}$  can be constructed from rates  $\omega_{ij}$ .


487 The solution of eq. (2.35) is obtained by using kinetic Monte Carlo (KMC) methods. KMC  
 488 explicitly simulates the dynamics of charge carriers by constructing a Markov chain in state space  
 489 and can find both stationary and transient solutions of the master equation. The main advantage  
 490 of KMC is that only states with a direct link to the current state need to be considered at each step.  
 491 Since these can be constructed solely from current site occupations, extensions to multiple charge  
 492 carriers (without the mean-field approximation), site-occupation dependent rates (needed for  
 493 the explicit treatment of Coulomb interactions), and different types of interacting particles and  
 494 processes, are straightforward. To optimize memory usage and efficiency, a combination of the  
 495 variable step size method [37] and the first reaction method is implemented.

496 To obtain the dynamics of charges using KMC, the program `xtp_kmc_run` executes a specific  
 497 `calculator` after reading its options (charge carrier type, runtime, number of carriers etc.) from  
 498 `options.xml`.

```

   KMC for a single carrier in periodic boundary conditions
  | xtp_kmc_run -o options.xml -f state.sql -e kmcsingle
  
```

```

   KMC for multiple carriers of the same type in periodic boundary conditions
  | xtp_kmc_run -o options.xml -f state.sql -e kmcmultiple
  
```

### 499 2.9.1 Extrapolation to nondispersive mobilities

sec:nondispersive

500 Predictions of charge-carrier mobilities in partially disordered semiconductors rely on charge  
 501 transport simulations in systems which are only several nanometers thick. As a result, simu-  
 502 lated charge transport might be dispersive for materials with large energetic disorder [38, 39]  
 503 and simulated mobilities are system-size dependent. In time-of-flight (TOF) experiments, how-  
 504 ever, a typical sample thickness is in the micrometer range and transport is often nondispersive.  
 505 In order to link simulation and experiment, one needs to extract the nondispersive mobility from  
 506 simulations of small systems, where charge transport is dispersive at room temperature.

507 Such extrapolation is possible if the temperature dependence of the nondispersive mobility is  
 508 known in a wide temperature range. For example, one can use analytical results derived for one-  
 509 dimensional models [40–42]. The mobility-temperature dependence can then be parametrized by

510 simulating charge transport at elevated temperatures, for which transport is nondispersive even  
 511 for small system sizes. This dependence can then be used to extrapolate to the nondispersive  
 512 mobility at room temperature [43].

513 For Alq<sub>3</sub>, the charge carrier mobility of a periodic system of 512 molecules was shown to be  
 514 more than three orders of magnitude higher than the nondispersive mobility of an infinitely  
 515 large system [43]. Furthermore, it was shown that the transition between the dispersive and  
 516 nondispersive transport has a logarithmic dependence on the number of hopping sites  $N$ . Hence,  
 517 a brute-force increase of the system size cannot resolve the problem for compounds with large  
 518 energetic disorder  $\sigma$ , since  $N$  increases exponentially with  $\sigma^2$ .

## 519 2.10 Stochastic Networks

sec:stochastic

520 The VOTCA package contains the functionality of generating large, amorphous charge transport  
 521 networks ( $\sim 10^6$  molecules). This is done with a combined coarse-grained and stochastic ap-  
 522 proach. VOTCA::CSG is used to generate a coarse-grained morphology. The stochastic modeling  
 523 of VOTCA::CTP allows to make a charge transfer network out of this morphology by reproduc-  
 524 ing the neighbor list (connectivity), transfer integrals, correlated site energies. An overview is  
 525 given in Figure 2.5.

526 Throughout this section we will use two state files. One is the state file `state_ref.sql` of the  
 527 smaller reference system that can be generated as explained in the previous sections. The second  
 528 one is the state file `state_cg.sql` of the coarse-grained system, or the stochastic network, that  
 529 can be parametrized as explained in this section.

530 When using the stochastic functionalities, please cite the corresponding work:

- 531 1. B. Baumeier, O. Stenzel, C. Poelking, D. Andrienko, and V. Schmidt: Stochastic modeling of  
 532 molecular charge transport networks. *Phys. Rev. B* 86, 184202 (2012)
- 533 2. P. Kordt and D. Andrienko: Modeling of Spatially Correlated Energetic Disorder in Organic  
 534 Semiconductors. *Journal of Chemical Theory and Computation* 12, 36–40 (2016)
- 535 3. P. Kordt, J. J. M. van der Holst, M. Al Helwi, W. Kowalsky, F. May, A. Badinski, C. Lennartz,  
 536 and D. Andrienko: Modeling of Organic Light Emitting Diodes: From Molecular to Device  
 537 Properties. *Advanced Functional Materials* 25, 1955–1971 (2015)

### 538 2.10.1 Coarse-grained morphology

539 The first step is to generate a coarse-grained morphology. In this example, it is done by mapping  
 540 a DPBIC molecule (which consists of 103 atoms) to a single point, its center of mass and by using  
 541 the iterative Boltzmann inversion (IBI) method. Starting point is a smaller reference morphology,  
 542 generated with GROMACS. Using the command

```
543 g_rdf -f traj.xtc -s topol.tpr
```

544 you can extract the radial distribution function  $g(r)$  of your reference topology, outputted into the  
 545 file `rdf.xvg`. This file, together with `table.xvg`, `grompp.mdp`, `topol.top`, `index.ndx` and `confout.gro`  
 546 form your reference data.

547 For VOTCA::CSG you need a `setting.xml` file:

```
548
549 <cg>
550 <!-- example for a non-bonded interaction entry -->
551 <non-bonded>
552 <!-- name of the interaction -->
553 <name>IR-IR</name>
554 <!-- types involved in this interaction -->
555 <type1>IR</type1>
556 <type2>IR</type2>
```

```

557 <!-- dimension + grid spacing of tables for calculations -->
558 <min>0.5</min>
559 <max>5.0</max>
560 <step>0.01</step>
561 <inverse>
562   <!-- target distribution (rdf), just give gromacs rdf.svg -->
563   <target>rdf.svg</target>
564   <!-- update cycles -->
565   <do_potential>1</do_potential>
566   <!-- additional post processing of dU before added to potential -->
567   <post_update>scale smooth</post_update>
568   <post_update_options>
569     <scale>0.5</scale> <!--Scale the potential before updating it -->
570     <smooth>
571       <iterations>2</iterations>
572     </smooth>
573   </post_update_options>
574   <!-- additional post processing of U after dU added to potential -->
575   <post_add></post_add>
576   <!-- name of the table for gromacs run -->
577   <gromacs>
578     <table>table_IR_IR.svg</table>
579   </gromacs>
580 </inverse>
581 </non-bonded>
582
583 <!-- general options for inverse script -->
584 <inverse>
585   <!-- 300*0.00831451 gromacs units -->
586   <kBT>2.49435300</kBT>
587   <initial_configuration>maindir</initial_configuration>
588   <!-- use gromacs as simulation program -->
589   <program>gromacs</program>
590   <!-- gromacs specific options -->
591   <gromacs>
592     <!-- trash so many frames at the beginning -->
593     <equi_time>500</equi_time>
594     <!-- grid for table*.svg !-->
595     <table_bins>0.001</table_bins>
596     <!-- cut the potential at this value (gromacs bug) -->
597     <pot_max>1000000</pot_max>
598     <!-- extend the tables to this value -->
599     <table_end>6.0</table_end>
600   </gromacs>
601   <!-- these files are copied for each new run -->
602   <filelist>grompp.mdp topol.top index.ndx table.svg</filelist>
603   <!-- do so many iterations -->
604   <iterations_max>500</iterations_max>
605   <!-- Try to clean a bit -->
606   <cleanlist>traj.xtc</cleanlist>
607   <!-- ibm: inverse boltzmann imc: inverse monte carlo -->
608   <method>ibi</method>
609   <!-- write log to this file -->
610   <log_file>inverse.log</log_file>
611   <!-- write restart step to this file -->
612   <restart_file>restart_points.log</restart_file>
613   <!-- imc specific stuff -->
614 </inverse>

```

```
</cg>
```

You run IBI using the command

```
csg_inverse -options settings.xml
```

IBI intends to find a potential  $U(r)$  that reproduces your radial distribution function. It is stored in the file `table_IR_IR.xvg` in our example.

With the interaction potential at hand, a large topology can be generated using molecular dynamics simulations for the coarse grained model. Starting point is a box with equally distributed points, with each point representing one molecule and with the number of points chosen such that the density of the reference system is reproduced. A small python script can generate the `conf.gro` to start from, here shown to obtain a  $50 \times 50 \times 120 \text{ nm}^3$  starting morphology.

```

628 from pylab import *
629 import numpy as np
630
631 lenX      = 50
632 lenY      = 50
633 lenZ      = 120
634 originalV = 4704.339
635 originalN = 4000
636 spacing   = (originalV/originalN)**(1./3.)
637
638 molecule  = "DPBIC"
639 resname   = "IRI"
640 atomname  = "IR"
641
642 newV = lenX*lenY*lenZ
643 newN = int(newV/originalV*originalN)
644
645 nX = int(lenX/spacing)+1
646 nY = int(lenY/spacing)+1
647 nZ = int(lenZ/spacing)+1
648
649 print "max. molecules in X direction: "+str(nX)
650 print "max. molecules in Y direction: "+str(nY)
651 print "max. molecules in Z direction: "+str(nZ)
652 print "total number of molecules: "+str(newN)
653
654 file = open("box.gro", "w")
655 file.write(molecule+"\n")
656 file.write(str(newN)+"\n")
657
658 atomnumber = 1
659 for iX in range(nX):
660     for iY in range(nY):
661         for iZ in range(nZ):
662             if(atomnumber > newN):
663                 break
664             posX = spacing*iX
665             posY = spacing*iY
666             posZ = spacing*iZ
667             print >> file, "%5d%-5s%5s%5d%8.3f%8.3f%8.3f%8.4f%8.4f%8.4f" % \
668                 (1,resname,atomname,1,posX,posY,posZ,0,0,0)
669             atomnumber += 1
670
671 file.write(" "+str(lenX)+" "+str(lenY)+" "+str(lenZ))

```

```

673 file.close()
674
675 print "Note: for some obscure reason VMD will not be able to read this file\
676 properly unless you open it once in vi and save it."
677

```


678 Open the `box.gro` in `vi` and save it (`:wq`), afterwards you can have a look at it in VDM. Run your  
 679 MD simulations using the `mdrun` command. In the end you can compare the radial distribution  
 680 functions of your reference and coarse-grained system, as shown in figure 2.6(a) as an example.

## 681 2.10.2 Charge transport network

682 To generate a charge transport network you first need a reference system with neighbor list, site  
 683 energies and transfer integrals calculated and stored in a `state.sql` state file. The procedure for all  
 684 these three properties is always the same: first analyze the reference data, and second import the  
 685 analyzation files and reproduce the properties.

### 686 Neighbor list

687 In the atomistic reference system molecules are connected if their two closest segments are below  
 688 a certain cut-off radius. This finer picture of segments does not exist in the coarse-grained system,  
 689 where each molecule is represented by a point. To mimick the neighbor list, the probability of  
 690 two molecules to be connected is analyzed as a function of their center-of-mass distance. This  
 691 can be done by using the `panalyze` calculator

 **Analyze the pair connectivity (neighborlist) in the reference system**  
 | `xtp_run -o options.xml -f state_ref.sql -e analyze`

692 with the options defined as follows:

#### 693 options\_analyze.xml

```

694 <options>
695   <analyze>
696     <resolution_space>0.05</resolution_space>
697   </analyze>
698 </options>
699
700
701

```

702 The only parameter needed is the spacial resolution, i.e., the bin size for calculating the probabil-  
 703 ities. The `panalyze` calculator outputs a file `analyze.distanceprobability.out` with the respective  
 704 probabilities. Now this file has to be imported into the coarse-grained state file

 **Import the reference pair connectivity (neighborlist) and reproduce it in stochastic network**  
 | `xtp_run -o options.xml -f state_cg.sql -e neighborlist`

705 using the following options:

#### 706 options\_import.xml

```

707 <options>
708   <neighborlist>
709     <probabilityfile>analyze.distanceprobability.out</probabilityfile>
710   </neighborlist>
711 </options>
712
713

```


715 For testing purposes, you can run the `panalyze` calculator on your coarse-grained state file and  
 716 compare the probability function to the reference. An example is shown in figure 2.6(b). You  
 717 can also also look at the file `analyze.distanceprobability.out` for both state files, which has the  
 718 distribution of coordination numbers (number of neighbors) and its average in.

719 **Site energies**

720 Site energies in amorphous organic semiconductors are roughly Gaussian distributed, with the  
 721 width of the Gaussian,  $\sigma$ , called the energetic disorder. However, there are correlations between  
 722 sites if they are close enough to each other. The aim in this section is therefore to reproduce the  
 723 correlated energetic landscape. The first step is to get a spatial correlation function as well as the  
 724 mean energy and the energetic disorder from your reference state file:

725

```

726  Analyze the energy distribution and correlation in the reference system
727 | xtp_run -o options.xml -f state_ref.sql -e eanalyze
  
```

726 with the following options:

727

728 **options\_analyze.xml**

729

```

730 <options>
731   <eanalyze>
732     <resolution_sites>0.05</resolution_sites>
733     <resolution_pairs>0.05</resolution_pairs>
734     <resolution_space>0.3</resolution_space>
735     <states>1,-1</states> <!-- +1 for hole transport, -1 for electron transport -->
736     <distancemode>centreofmass</distancemode>
737   </eanalyze>
738 </options>
  
```

740 The first three parameters determine bin sizes, then you can choose to look at hole and/or elec-  
 741 tron energy. The keyword *centreofmass* means, that the correlation function is calculated as a  
 742 function of the centre-of-mass distance of molecules and not as a function of their nearest seg-  
 743 ments. For the stochastic simulations you always have to use the *centreofmass* mode!

744 The output files of this calculator that we need are **eanalyze.sitecorr\_e.out** (for electrons) and  
 745 **eanalyze.sitecorr\_h.out** (for holes). In the second line of this file, you find mean and sigma of the  
 746 energy distribution, as well as the mean of the static energies (without induction):

747

```

748 # EANALYZE: SPATIAL SITE-ENERGY CORRELATION
749 # AVG -0.4412655 STD 0.1739638 MIN_R 0.8365040 MAX_R 14.4771496 AVGESTATIC
750 -0.4730655
751 . . .
  
```

752 These values have to be inserted manually into the options file for importing to the coarse-  
 753 grained system (see below). Apart from that, the file contains the spatial correlation function.

754

755 You generate energies following this distribution and correlation by using the *eimport* calculator

```

756  Import the energy distribution and correlation and reproduce it in stochastic network
757 | xtp_run -o options.xml -f state_ref.sql -e eimport
  
```

756 with the options:

757

758 **options\_import.xml**

759

```

760 <options>
761   <eimport>
762     <probabilityfile_h>reference/eanalyze.sitecorr_h.out</probabilityfile_h>
763     <sigma_h>0.1763163</sigma_h>
764     <avgestatic_h>-0.5913265</avgestatic_h>
765     <probabilityfile_e>reference/eanalyze.sitecorr_e.out</probabilityfile_e>
  
```



```

766         <sigma_e>0.1739638</sigma_e>
767         <avgestatic_e>-0.4730655</avgestatic_e>
768         <cutoff>8.5</cutoff>
769         <seed>1</seed>
770     </eimport>
771 </options>
772

```

773 The *cutoff* keyword can be used to read in the correlation function only up to a certain distance,  
774 which can be useful if larger distances yield unphysical results.

### 775 Transfer Integrals

776 The last ingredient reproduced by the stochastic approach are transfer integrals  $J$ . The idea is that  
777  $\log_{10}(J^2/eV^2)$  is roughly Gaussian distributed, with mean and error of the distribution varying  
778 with distance (see figure 2.6 (d)). Use the calculator  
779

 **Analyze the distance-depend distribution of transfer integrals in the reference system**  
**|** `xtp_run -o options.xml -f state_ref.sql -e ianalyze`

780 with options

781

### 782 options\_analyze.xml

783

```

784 <options>
785     <ianalyze>
786         <resolution_logJ2>0.05</resolution_logJ2>
787         <resolution_space>0.05</resolution_space>
788         <states>1,-1</states> <!-- +1 for hole transport, -1 for electron transport -->
789     </ianalyze>
790 </options>
791

```

792 That will generate the files **ianalyze.ispatial\_e.out** and **ianalyze.ispatial\_h.out**, which contain  
793 means and errors as a function of centre-of-mass distance.

794

795 Now use the *iimport* calculator to generate transfer integrals in the coarse grained state file, fol-  
796 lowing the same statistics.

 **Import distance dependent distribution of transfer integrals and reproduce in stochastic network**  
**|** `xtp_run -o options.xml -f state_cg.sql -e iimport`

### 797 options\_import.xml

798

```

799 <options>
800     <iimport>
801         <TI_tag></TI_tag>
802         <TI_file></TI_file>
803         <idft_jobs_file></idft_jobs_file>
804         <probabilityfile_h>reference/ianalyze.ispatial_h.out</probabilityfile_h>
805         <probabilityfile_e>reference/ianalyze.ispatial_e.out</probabilityfile_e>
806     </iimport>
807 </options>
808


```

### 809 einternal

810 Run the *einternal* calculator, just as you do it for the reference system.

## 811 Rates

812 If you followed the steps in this section, you have everything at hand to calculate charge transfer  
813 rates for the coarse grained system from the stochastic ingredients:

```
814  Calculate rates in the stochastic network
815 | xtp_run -o options.xml -f state_cg.sql -e rates
```

814 Options are the same as for the reference file. You can check the result by comparing rates from  
815 your reference to the coarse-grained system, see figure 2.6(e) for an example. The resulting charge  
816 transport network can be used for kinetic Monte Carlo simulations with VOTCA. If everything  
817 goes well, mobilities for both systems should agree, as shown in figure 2.6(f).

## 818 2.11 Macroscopic observables

sec:analysis

819 Spatial distributions of charge and current densities can provide a better insight in the micro-  
820 scopic mechanisms of charge transport. If  $O$  is an observable which has a value  $O_\alpha$  in a state  $\alpha$ ,  
821 its ensemble average at time  $t$  is a sum over all states weighted by the probability  $P_\alpha$  to be in a  
822 state  $\alpha$  at time  $t$

$$\langle O \rangle = \sum_{\alpha} O_{\alpha} P_{\alpha}. \quad (2.36) \quad \text{equ:ensemble}$$

823 If  $O$  does not explicitly depend on time, the time evolution of  $\langle O \rangle$  can be calculated as

$$\frac{d\langle O \rangle}{dt} = \sum_{\alpha, \beta} [P_{\beta} \Omega_{\beta\alpha} - P_{\alpha} \Omega_{\alpha\beta}] O_{\alpha} = \sum_{\alpha, \beta} P_{\beta} \Omega_{\beta\alpha} [O_{\alpha} - O_{\beta}]. \quad (2.37)$$

824 If averages are obtained from KMC trajectories,  $P_{\alpha} = s_{\alpha}/s$ , where  $s_{\alpha}$  is the number of Markov  
825 chains ending in the state  $\alpha$  after time  $t$ , and  $s$  is the total number of chains.

Alternatively, one can calculate time averages by analyzing a single Markov chain. If the total occupation time of the state  $\alpha$  is  $\tau_{\alpha}$  then

$$\bar{O} = \frac{1}{\tau} \sum_{\alpha} O_{\alpha} \tau_{\alpha}, \quad (2.38) \quad \text{equ:time}$$

826 where  $\tau = \sum_{\alpha} \tau_{\alpha}$  is the total time used for time averaging.

827 For ergodic systems and sufficient sampling times, ensemble and time averages should give iden-  
828 tical results. In many cases, the averaging procedure reflects a specific experimental technique.  
829 For example, an ensemble average over several KMC trajectories with different starting condi-  
830 tions corresponds to averaging over injected charge carriers in a time-of-flight experiment. In  
831 what follows, we focus on the single charge carrier (low concentration of charges) case.

### 832 2.11.1 Charge density

sec:occupation

833 For a specific type of particles, the microscopic charge density of a site  $i$  is proportional to the  
834 occupation probability of the site,  $p_i$

$$\rho_i = e p_i / V_i, \quad (2.39)$$

835 where, for an irregular lattice, the effective volume  $V_i$  can be obtained from a Voronoi tessel-  
836 lation of space. For reasonably uniform lattices (uniform site densities) this volume is almost  
837 independent of the site and a constant volume per site,  $V_i = V/N$ , can be assumed. In the  
838 macroscopic limit, the charge density can be calculated using a sxtpting kernel function, i.e. a  
839 distance-weighted average over multiple sites. Site occupations  $p_i$  can be obtained from eq. (2.36)  
840 or eq. (2.38) by using the occupation of site  $i$  in state  $\alpha$  as an observable.

841 If the system is in thermodynamic equilibrium, that is without sources or sinks and without  
842 circular currents (and therefore no net flux) a condition, known as detailed balance, holds

$$p_j \omega_{ji} = p_i \omega_{ij}, \quad (2.40) \quad \text{equ:detailed\_balance}$$

843 It can be used to test whether the system is ergodic or not by correlating  $\log p_i$  and the site energy  
844  $E_i$ . Indeed, if  $\lambda_{ij} = \lambda_{ji}$  the ratios of forward and backward rates are determined solely by the  
845 energetic disorder,  $\omega_{ji}/\omega_{ij} = \exp(-\Delta E_{ij}/k_B T)$  (see eq. (2.31)).

### 846 2.11.2 Current

sec:vaverage

847 If the position of the charge,  $\vec{r}$ , is an observable, the time evolution of its average  $\langle \vec{r} \rangle$  is the total  
848 current in the system

$$\vec{J} = e \langle \dot{\vec{r}} \rangle = e \frac{d \langle \vec{r} \rangle}{dt} = e \sum_{i,j} p_j \omega_{ji} (\vec{r}_i - \vec{r}_j). \quad (2.41) \quad \text{equ:current\_def}$$

849 Symmetrizing this expression we obtain

$$\vec{J} = \frac{1}{2} e \sum_{i,j} (p_j \omega_{ji} - p_i \omega_{ij}) \vec{r}_{ij}, \quad (2.42) \quad \text{equ:current}$$

850 where  $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ . Symmetrization ensures equal flux splitting between neighboring sites and  
851 absence of local average fluxes in equilibrium. It allows to define a local current through site  $i$  as

$$\vec{J}_i = \frac{1}{2} e \sum_j (p_j \omega_{ji} - p_i \omega_{ij}) \vec{r}_{ij}. \quad (2.43) \quad \text{equ:site\_current}$$

852 A large value of the local current indicates that the site contributes considerably to the total cur-  
853 rent. A collection of such sites thus represents most favorable charge pathways [44].

### 854 2.11.3 Mobility and diffusion constant

sec:mobility

855 For a single particle, e.g. a charge or an exciton, a zero-field mobility can be determined by  
856 studying particle diffusion in the absence of external fields. Using the particle displacement  
857 squared,  $\Delta \mathbf{r}_i^2$ , as an observable we obtain

$$2dD_{\gamma\delta} = \frac{d \langle \Delta r_{i,\gamma} \Delta r_{i,\delta} \rangle}{dt} = \sum_{\substack{i,j \\ i \neq j}} p_j \omega_{ji} (\Delta r_{i,\gamma} \Delta r_{i,\delta} - \Delta r_{j,\gamma} \Delta r_{j,\delta}) = \sum_{\substack{i,j \\ i \neq j}} p_j \omega_{ji} (r_{i,\gamma} r_{i,\delta} - r_{j,\gamma} r_{j,\delta}). \quad (2.44) \quad \text{equ:diffusion}$$

858 Here  $\vec{r}_i$  is the coordinate of the site  $i$ ,  $D_{\gamma\delta}$  is the diffusion tensor,  $\gamma, \delta = x, y, z$ , and  $d = 3$  is the  
859 system dimension. Using the Einstein relation,

$$D_{\gamma\delta} = k_B T \mu_{\gamma\delta}, \quad (2.45)$$

860 one can, in principle, obtain the zero-field mobility tensor  $\mu_{\gamma\delta}$ . Eq. (2.44), however, does not take  
861 into account the use of periodic boundary conditions when simulating charge dynamics. In this  
862 case, the simulated occupation probabilities can be compared to the solution of the Smoluchowski  
863 equation with periodic boundary conditions (see the supporting information for details).

864 Alternatively, one can directly analyze time-evolution of the KMC trajectory and obtain the dif-  
865 fusion tensor from a linear fit to the mean square displacement,  $\Delta r_{i,\gamma} \Delta r_{i,\delta} = 2dD_{\gamma\delta} t$ .

866 The charge carrier mobility tensor,  $\hat{\mu}$ , for any value of the external field can be determined either  
867 from the average charge velocity defined in eq. (2.41)

$$\langle \dot{\vec{v}} \rangle = \sum_{i,j} p_j \omega_{ji} (\vec{r}_i - \vec{r}_j) = \hat{\mu} \vec{F}, \quad (2.46)$$

868 or directly from the KMC trajectory. In the latter case the velocity is calculated from the un-  
 869 wrapped (if periodic boundary conditions are used) charge displacement vector divided by the  
 870 total simulation time. Projecting this velocity on the direction of the field  $\vec{F}$  yields the charge car-  
 871 rier mobility in this particular direction. In order to improve statistics, mobilities can be averaged  
 872 over several KMC trajectories and MD snapshots.

#### 873 2.11.4 Spatial correlations of energetic disorder

sec:eanalyze

874 Long-range, e.g. electrostatic and polarization, interactions often result in spatially correlated  
 875 disorder [45], which affects the onset of the mobility-field (Poole-Frenkel) dependence [40, 46, 47].  
 876 To quantify the degree of correlation, one can calculate the spatial correlation function of  $E_i$  and  
 877  $E_j$  at a distance  $r_{ij}$

$$C(r_{ij}) = \frac{\langle (E_i - \langle E \rangle) (E_j - \langle E \rangle) \rangle}{\langle (E_i - \langle E \rangle)^2 \rangle}, \quad (2.47) \quad \text{equ:cf}$$

878 where  $\langle E \rangle$  is the average site energy.  $C(r_{ij})$  is zero if  $E_i$  and  $E_j$  are uncorrelated and 1 if they are  
 879 fully correlated. For a system of randomly oriented point dipoles, the correlation function decays  
 880 as  $1/r$  at large distances [48].

881 For systems with spatial correlations, variations in site energy differences,  $\Delta E_{ij}$ , of pairs of  
 882 molecules from the neighbor list are smaller than variations in site energies,  $E_i$ , of all individ-  
 883 ual molecules. Since only neighbor list pairs affect transport, the distribution of  $\Delta E_{ij}$  rather than  
 884 that of individual site energies,  $E_i$ , should be used to characterize energetic disorder.

885 Note that the `eanalyze` calculator takes into account *all* contributions to the site energies



#### Analyze distribution and correlations of site energies

```
| xtp_run -o options.xml -f state.sql -e eanalyze
```

<b>MORPHOLOGY</b>		
	<b>EXTRACT</b>	<b>REPRODUCE</b>
<b>RDF and coarse grained potential</b>	GROMACS g_rdf -f traj.xtc -s topol.tpr	VOTCA::CSG csg_inverse -options settings.xml
<b>morphology</b>		GROMACS mdrun

<b>RATES</b>		
	<b>EXTRACT</b>	<b>REPRODUCE</b>
<b>neighbor list (pairs)</b>	xtp_run -e panalyze -o options.xml -f state_reference.sql	xtp_run -e neighborlist -o options.xml -f state_cg.sql
<b>site energies</b>	xtp_run -e eanalyze -o options.xml -f state_reference.sql	xtp_run -e eimport -o options.xml -f state_cg.sql
<b>transfer integrals</b>	xtp_run -e ianalyze -o options.xml -f state_reference.sql	xtp_run -e iimport -o options.xml -f state_cg.sql
<b>rates</b>	not used directly	xtp_run -e rates -o options.xml -f state_cg.sql

Figure 2.5: Stochastic Model in VOTCA. Overview of the different steps for generating stochastic charge transport networks in VOTCA. The Molecular Dynamics software GROMACS allows to analyze the radial distribution function of a morphology, which is then used by VOTCA::CSG to generate a coarse-grained potential that reproduces this distribution function. This potential can then be used for coarse-grained simulations in GROMCAS. For calculating rates in the coarse-grained morphology, first the relevant parameters are extracted (panalyze, eanalyze, ianalyze) from the reference morphology and then reproduced in the coarse-grained morphology (neighborlist, eimport, iimport). With all these at hand, the rates calculator can be used in the coarse-grained morphology.

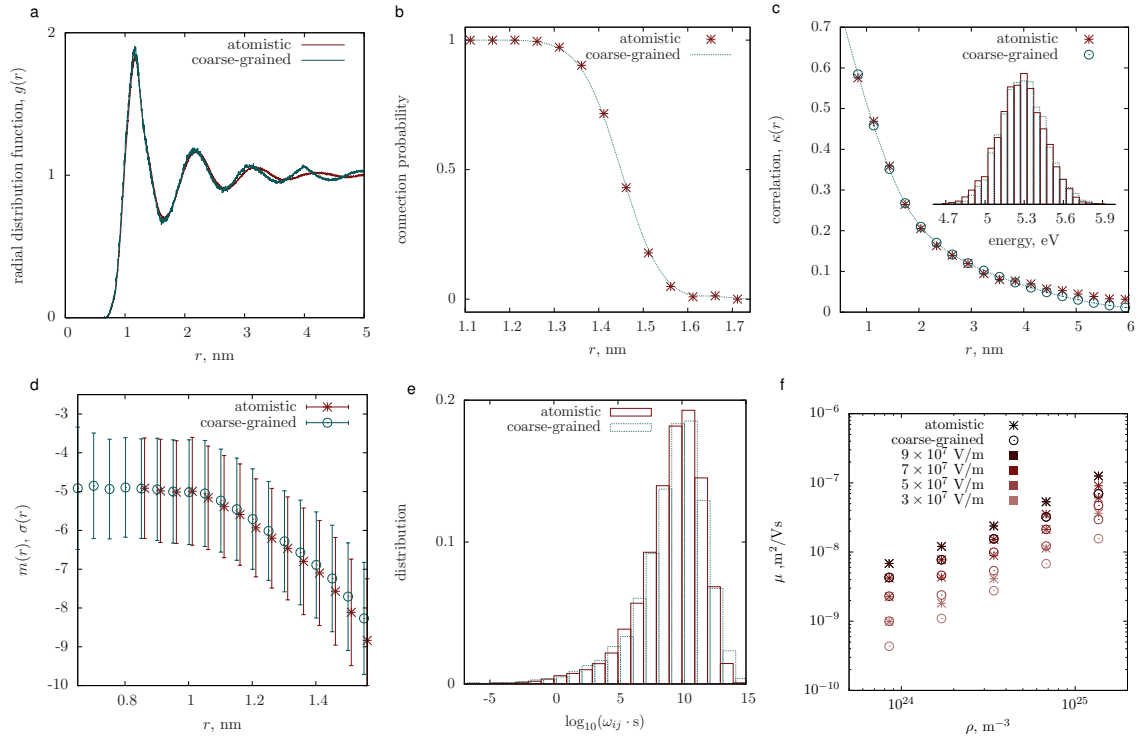


Figure 2.6: Comparison of the atomistic ( $17 \times 17 \times 17 \text{ nm}^3$ ) and coarse-grained ( $50 \times 50 \times 120 \text{ nm}^3$ ) models. (a) Radial distribution function,  $g(r)$ . (b) Probability of two sites to be connected (added to the neighbor list) as a function of their separation. (c) Spatial site energy autocorrelation function,  $\kappa(r)$ ; Inset: Site energy distribution. (d) Mean  $m$  and width  $\sigma$  of a distribution of the logarithm of electronic couplings,  $\log_{10}(J^2/eV^2)$ , for molecules at a fixed separation  $r$ . (e) Rate distributions. (f) Mobility as a function of hole density, plotted for four different electric fields. fig:stochastic

## 886 Chapter 3

# 887 Input and output files

### 888 3.1 Atomistic topology

889 sec:atomistic

889 If you are using GROMACS for generating atomistic configurations, it is possible to directly use  
890 the topology file provided by GROMACS (`topology.tpr`). In this case the GROMACS residue and  
891 atom names should be used to specify the coarse-grained topology and conjugated segments.  
892 A custom topology can also be defined using an XML file. Moreover, it is possible to partially  
893 overwrite the information provided in, for example, GROMACS topology file. We will illustrate  
894 how to create a custom topology file using DCV2T. The structure of DCV2T, together with atom  
895 type definitions, is shown in fig. 3.1. DCV2T has two thiophene (THI) and two dicyanovinyl  
896 (NIT) residues. The pdb file which contains residue types, residue numbering, atom names,  
897 atom types, and atom coordinates is shown in listing 3.1.

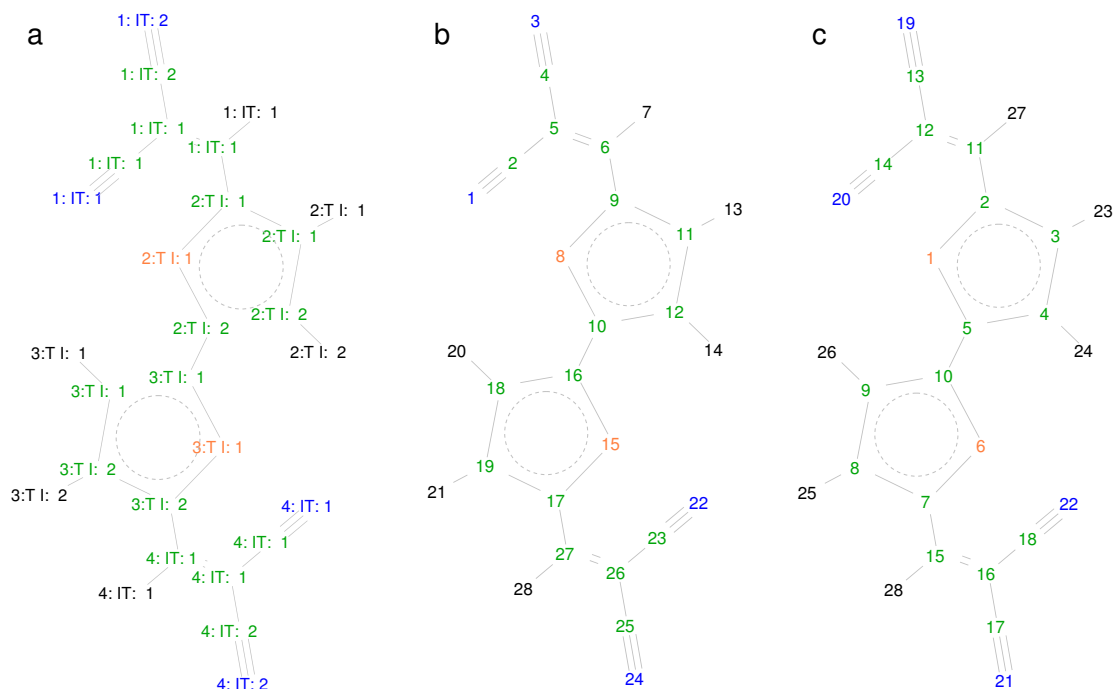


Figure 3.1: (a) DCV2T with atoms labelled according to `residue_number:residue_name:atom_name`. There are four residues and two residue types: thiophene (THI) and dicyanovinyl (NIT). The corresponding pdb file is shown in listing 3.1. Atom numbering is used to split conjugated segments on rigid fragments and to link atomistic (b) from GROMACS topology and quantum descriptions (c).

fig:dcv2t

list.pdb

Listing 3.1: pdb file of DCV2T.

898											
899	HETATM	1	N1	NIT	1	2.388	8.533	11.066	1.00	4.14	N
900	HETATM	2	CN1	NIT	1	1.984	9.553	10.718	1.00	2.54	C
901	HETATM	3	N2	NIT	1	-1.138	10.872	10.087	1.00	3.24	N
902	HETATM	4	CN2	NIT	1	0.003	10.871	10.213	1.00	2.37	C
903	HETATM	5	CC1	NIT	1	1.441	10.824	10.327	1.00	1.91	C
904	HETATM	6	C1	NIT	1	2.193	11.939	10.071	1.00	1.61	C
905	HETATM	7	HN1	NIT	1	1.715	12.710	9.872	1.00	1.97	H
906	HETATM	8	S1	THI	2	4.758	10.743	10.130	1.00	1.52	S
907	HETATM	9	CA1	THI	2	3.613	12.024	9.948	1.00	1.22	C
908	HETATM	10	CA2	THI	2	6.099	11.836	9.997	1.00	1.30	C
909	HETATM	11	CB1	THI	2	4.251	13.243	9.782	1.00	1.39	C
910	HETATM	12	CB2	THI	2	5.658	13.131	9.818	1.00	1.45	C
911	HETATM	13	HC1	THI	2	3.800	14.047	9.660	1.00	1.66	H
912	HETATM	14	HC2	THI	2	6.230	13.860	9.731	1.00	1.74	H
913	HETATM	15	S1	THI	3	8.803	12.414	9.882	1.00	1.38	S
914	HETATM	16	CA1	THI	3	7.456	11.347	10.094	1.00	1.37	C
915	HETATM	17	CA2	THI	3	9.940	11.122	10.152	1.00	1.42	C
916	HETATM	18	CB1	THI	3	7.873	10.048	10.355	1.00	1.73	C
917	HETATM	19	CB2	THI	3	9.267	9.926	10.399	1.00	1.82	C
918	HETATM	20	HC1	THI	3	7.288	9.335	10.487	1.00	2.05	H
919	HETATM	21	HC2	THI	3	9.704	9.123	10.576	1.00	2.21	H
920	HETATM	22	N1	NIT	4	11.235	14.572	9.094	1.00	3.08	N
921	HETATM	23	CN1	NIT	4	11.665	13.566	9.441	1.00	2.04	C
922	HETATM	24	N2	NIT	4	14.733	12.005	10.009	1.00	2.17	N
923	HETATM	25	CN2	NIT	4	13.590	12.149	9.933	1.00	1.77	C
924	HETATM	26	CC1	NIT	4	12.156	12.282	9.861	1.00	1.71	C
925	HETATM	27	C1	NIT	4	11.363	11.220	10.154	1.00	1.59	C
926	HETATM	28	HN1	NIT	4	11.813	10.440	10.389	1.00	1.89	H



tab:map

Table 3.1: Description of the XML mapping file (`map.xml`).

topology	Definitions of molecules, segments, and fragments.
molecules	Container for all molecules.
molecule	Mapping of a single molecule.
name	Name of the molecule in the coarse-grained model.
ident	Name (identification) of the molecule in the all-atom representation. This must match the molecule name in the atomistic representation.
segments	Partitioning of the molecule on conjugated segments.
segment	Description of a conjugated segment.
name	Name of a conjugated segment in a molecule.
fragments	Container for all fragments in a segment.
fragment	Description of a rigid fragment.
name	Name of the rigid fragment in a conjugated segment
mdatoms	List of all atoms belonging to the rigid fragment in the format residue number:residue name:atom name.
qmatoms	List of atoms of the rigid fragment in its ground state geometry, atom number:atom type.
weights	Weights are used to determine the fragment center. The order should be the same as in the mdatoms and qmatoms definitions. If the mass of a nucleus in atomic mass units is used, the center of the rigid fragment will be its center of mass.
localframe	Three atoms which define a local frame for each rigid fragment.

## 928 3.2 Mapping file

sec:xmlmap

929 The mapping file (referred here as `map.xml`) is used by the program `xtp_map` to convert an  
 930 atomistic trajectory to a trajectory with conjugated segments and rigid fragments. This trajectory  
 931 is stored in a state file and contains positions, names, types of atoms belonging to rigid fragments.  
 932 The description of the mapping options is given in table 3.1. An example of `map.xml` for a  
 933 DCV2T molecule is shown in listing 3.2.  
 934 The file `map.xml` contains the whole electrostatic information about the molecules as well as  
 935 the structural information. The `toolpdb2map` creates a `map.xml` from a `pdb` file and is a good  
 936 starting point for further refinement.

list:map

Listing 3.2: Examl of `map.xml` for DCV2T. Each rigid fragment (coarse-grained bead) is defined by a list of atoms. Atom numbers, names, and residue names should correspond to those used in GROMACS topology (see the corresponding listing 3.1 of the `pdb` file).

```

937 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
938 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
939 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
940 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
941 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
942 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
943 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
944 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
945 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
946 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
947 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
948 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
949 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
950 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
951 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
952 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
953 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
954 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
955 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
956 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
957 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
958 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
959 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
960 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
961 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
962 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
963 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->
964 <!-- this file is used to conver an atomistic trajectory to conjugated segments -->

```

```

965 <U_nC_nN_h>0.1</U_nC_nN_h> <!-- Reorg. discharge -->
966 <U_nC_nC_h>0.1</U_nC_nC_h> <!-- Reorg. charge -->
967
968 <!-- MD QM MP Mapping -->
969 <fragments>
970 <fragment>
971 <name>N1</name> <!-- name of the rigid fragment within the segment -->
972 <!-- list of atoms in the fragment resnum:resname:atomname -->
973 <mdatoms>1:NIT:N1 1:NIT:CN1 1:NIT:N2 1:NIT:CN2 1:NIT:CC1 1:NIT:C1 1:NIT:HN1</mdatoms>
974 <!-- corresponding ground state geometry atomnumber:atomtype read from .xyz file-->
975 <qmatoms> 20:N 19:C 14:N 13:C 12:C 11:C 23:H </qmatoms>
976 <!-- corresponding group state geometry multipoles read from .mps files -->
977 <mpoles> 20:N 19:C 14:N 13:C 12:C 11:C 23:H </mpoles>
978 <!-- weights to determine the fragment center (here CoM is used) -->
979 <weights> 14 12 14 12 12 12 1 </weights>
980 <!-- three atoms: define a cartesian local frame, two atoms: fragment is assumed to be
981 rotationally invariant around the axis, one atom: fragment is assumed isotropic -->
982 <localframe> 20 19 14 </localframe>
983 <!-- Optional parameters (if not set <localframe> is used): used when atom labels in the .mps
984 and .xyz file differ or more sites in the .mps file are used, so refers to <mpoles> -->
985 <localframe_mps> 20 19 14 </localframe_mps>
986 <!-- Optional parameters (if not set <localframe> is used): weights to determine the
987 fragment center (here CoM is used), used when atom labels in the .mps and .xyz file
988 differ or additional sites in the .mps file are used -->
989 <weights_mps> 14 12 14 12 12 12 1 </
990 weights_mps>
991 <!-- Optional flag: says if a site is virtual or not, (virtual=1, real=0)-->
992 <virtual_mps> 0 0 0 0 0 0 0 </
993 virtual_mps>
994 </fragment>
995
996 <fragment>
997 <name>TH1</name>
998 <mdatoms>2:THI:S1 2:THI:CA1 2:THI:CA2 2:THI:CB1 2:THI:CB2 2:THI:HC1 2:THI:HC2</mdatoms>
999 <qmatoms> 7:S 8:C 6:C 9:C 10:C 24:H 25:H </qmatoms>
1000 <mpoles> 7:S 8:C 6:C 9:C 10:C 24:H 25:H </mpoles>
1001 <weights> 32 12 12 12 12 1 1 </weights>
1002 <localframe> 7 8 6 </localframe>
1003 </fragment>
1004
1005 <fragment>
1006 <name>TH2</name>
1007 <mdatoms>3:THI:CA1 3:THI:CA2 3:THI:CB1 3:THI:CB2 3:THI:HC1 3:THI:HC2</mdatoms>
1008 <qmatoms> 3:S 4:C 2:C 5:C 1:C 26:H 27:H </qmatoms>
1009 <weights> 32 12 12 12 12 1 1 </weights>
1010 <localframe> 3 4 2 </localframe>
1011 </fragment>
1012
1013 <fragment>
1014 <name>N12</name>
1015 <mdatoms>4:NIT:N1 4:NIT:CN1 4:NIT:N2 4:NIT:CN2 4:NIT:CC1 4:NIT:C1 4:NIT:HN1</mdatoms>
1016 <qmatoms> 22:N 21:C 18:N 17:C 16:C 15:C 28:H </qmatoms>
1017 <mpoles> 22:N 21:C 18:N 17:C 16:C 15:C 28:H </mpoles>
1018 <weights> 14 12 14 12 12 12 1 </weights>
1019 <localframe> 22 21 18 </localframe>
1020 </fragment>
1021 </fragments>
1022 </segment>
1023 </segments>
1024 </molecule>
1025 </molecules>
1026 </topology>

```

### 3.3 Molecular orbitals

1028 If the semi-empirical method is used to calculate electronic coupling elements, molecular or-  
1029 bitals of all molecules must be supplied. They can be generated using Gaussian program. The  
1030 Gaussian input file for DCV2T is shown in listing 3.3. Provided with this input, Gaussian will  
1031 generate fort. 7 file which contains the molecular orbitals of a DCV2T. This file can be renamed  
1032 to DCV2T.orb. Note that the order of the atoms in the input file and the order of coefficients  
1033 should always match. Therefore, the coordinate part of the input file must be supplied together  
1034 with the orbitals. We will assume the coordinates, in the format atom\_type: x y z, is saved  
1035 to the DCV2T.xyz file.  
1036

**Be careful!**

Izindo requires the specification of orbitals for hole and electron transport in `map.xml`. They are the HOMO and LUMO respectively and can be retrieved from the `log` file from which the `DCV2T.orb` file is generated. The number of alpha electrons is the HOMO, the LUMO is HOMO+1

list:zindo\_orbitals

**Listing 3.3:** Gaussian input file `get_orbitals.com` used for generating molecular orbitals. The first line contains the name of the check file, the second the requested RAM. `int=zindos` requests the method ZINDO, `punch=mo` states that the molecular orbitals ought to be written to the `fort.7` file, `nosymm` forbids use of symmetry and is necessary to ensure correct position of orbitals with respect to the provided coordinates. The two integer numbers correspond to the charge and multiplicity of the system: 0 1 corresponds to a neutral system with a multiplicity of one. They are followed by the types and coordinates of all atoms in the molecule.

```

1037 %chk=DCV2T.chk
1038 %mem=100Mb
1039 #p int=zindos punch=mo nosymm
1040
1041 DCV2T molecular orbitals
1042
1043
1044 0 1
1045 S      -1.44650      2.12185      0.00135
1046 C      -2.43098      0.58936     -0.00048
1047 C      -1.59065     -0.51859     -0.00146
1048 C      -0.21222     -0.22233     -0.00095
1049 C       0.07761      1.13376      0.00040
1050 S       2.87651      0.79316      0.00148
1051 C       3.86099      2.32565      0.00235
1052 C       3.02066      3.43359      0.00231
1053 C       1.64223      3.13733      0.00162
1054 C       1.35240      1.78125      0.00114
1055 C      -3.85350      0.52245     -0.00081
1056 C      -4.79569      1.52479     -0.00008
1057 C      -6.18500      1.18622     -0.00117
1058 C      -4.47544      2.91565      0.00081
1059 C       5.28350      2.39256      0.00296
1060 C       6.22569      1.39020      0.00327
1061 C       7.61500      1.72876      0.00432
1062 C       5.90542     -0.00064      0.00333
1063 N      -7.32389      0.89743     -0.00195
1064 N      -4.21872      4.06274      0.00142
1065 N       8.75389      2.01754      0.00510
1066 N       5.64864     -1.14772      0.00361
1067 H      -1.98064     -1.52966     -0.00256
1068 H       0.55785     -0.98374     -0.00169
1069 H       3.41065      4.44466      0.00272
1070 H       0.87216      3.89874      0.00147
1071 H      -4.24640     -0.49192     -0.00188
1072 H       5.67641      3.40692      0.00337

```

## 3.4 Monomer calculations for DFT transfer integrals

list:edft\_gaussian\_xml

**Listing 3.4:** Example `package.xml` file for the Gaussian package required in the options of the `edft` calculator for the monomer calculations as preparation for the determination of transfer integrals using DIPRO.

```

1075 <package>
1076 <name>gaussian</name>
1077 <executable>g09</executable>
1078 <checkpoint></checkpoint>
1079

```

```

1080 <scratch></scratch>
1081
1082 <charge>0</charge>
1083 <spin>1</spin>
1084 <options># pop=minimal pbepbe/6-311g** scf=tight punch=mo nosymm test</options>
1085 <memory>1Gb</memory>
1086 <threads>2</threads>
1087
1088 <cleanup></cleanup>
1089 </package>

```

list:edft\_turbomole.xml

Listing 3.5: Example package.xml file for the Turbomole package required in the options of the `edft` calculator for the monomer calculations as preparation for the determination of transfer integrals using DIPRO.

```

1091 <package>
1092 <name>turbomole</name>
1093 <executable>ridft</executable>
1094 <scratch>/tmp</scratch>
1095
1096 <options>
1097 TITLE
1098 a coord
1099 *
1100 no
1101 b all def-TZVP
1102 *
1103 eht
1104 y
1105 0
1106 y
1107 dft
1108 on
1109 func
1110 pbe
1111 grid
1112 m3
1113 *
1114 ri
1115 on
1116 m 300
1117 *
1118 scf
1119 conv
1120 7
1121 iter
1122 200
1123
1124 marij
1125
1126 q
1127 </options>
1128
1129 <cleanup></cleanup>
1130 </package>

```

list:edft\_nwchem.xml

Listing 3.6: Example package.xml file for the NWChem package required in the options of the `edft` calculator for the monomer calculations as preparation for the determination of transfer integrals using

```

DIPRO.
1133
1134 <package>
1135   <name>nwchem</name>
1136   <executable>nwchem</executable>
1137   <checkpoint></checkpoint>
1138   <scratch>/tmp/nwchem</scratch>
1139   <charge>0</charge>
1140   <spin>1</spin>
1141   <threads>1</threads>
1142   <memory></memory>
1143   <options>
1144   start
1145   basis
1146   * library 6-311gss
1147   end
1148   memory 1500 mb
1149
1150   dft
1151     xc xpbe96 cpbe96
1152     direct
1153     iterations 100
1154     noprint "final vectors analysis"
1155   end
1156   task dft
1157 </options>
1158   <cleanup></cleanup>
1159 </package>

```

### 1161 3.5 Pair calculations for DFT transfer integrals

list:ldft\_gaussian\_xml

Listing 3.7: Example package.xml file for the Gaussian package required in the options of the `ldft calculator` for the pair calculations and the determination of transfer integrals using DIPRO.

```

1162
1163 <package>
1164   <name>gaussian</name>
1165   <executable>g09</executable>
1166   <checkpoint></checkpoint>
1167   <scratch></scratch>
1168
1169   <charge>0</charge>
1170   <spin>1</spin>
1171   <options># pop=minimal pbepbe/6-311g** nosymm IOp(3/33=1,3/36=-1) punch=mo guess=cards scf=
1172   <memory>1Gb</memory>
1173   <threads>1</threads>
1174
1175   <cleanup></cleanup>
1176 </package>

```

list:ldft\_turbomole\_xml

Listing 3.8: Example package.xml file for the Turbomole package required in the options of the `ldft calculator` for the pair calculations and the determination of transfer integrals using DIPRO.

```

1178
1179 <package>
1180   <name>turbomole</name>
1181   <executable>ridft</executable>
1182   <scratch>/tmp</scratch>
1183

```

```

1184 <options>
1185 $intsdebug cao
1186 a coord
1187 *
1188 no
1189 b all def-TZVP
1190 *
1191 eht
1192 y
1193 0
1194 y
1195 dft
1196 on
1197 func
1198 pbe
1199 grid
1200 m3
1201 *
1202 ri
1203 on
1204 m 300
1205 *
1206 scf
1207 conv
1208 7
1209 iter
1210 1
1211 diis
1212 3
1213 damp
1214 0.00
1215
1216
1217
1218 marij
1219
1220 q
1221 </options>
1222
1223 <cleanup></cleanup>
1224 </package>

```

list.idft\_nwchem.xml

Listing 3.9: Example package.xml file for the NWChem package required in the options of the `idft` calculator for the pair calculations and the determination of transfer integrals using DIPRO.

```

1226 <package>
1227 <name>nwchem</name>
1228 <executable>nwchem</executable>
1229 <checkpoint></checkpoint>
1230 <scratch>/tmp/nwchem</scratch>
1231 <charge>0</charge>
1232 <spin>1</spin>
1233 <memory></memory>
1234 <threads>1</threads>
1235 <options>
1236 start
1237 basis
1238 * library 6-311gss
1239 end
1240

```

```

1241 memory 1500 mb
1242
1243 dft
1244   print "ao overlap"
1245   xc xpbe96 cpbe96
1246   direct
1247   iterations 1
1248   convergence nodamping nodiis
1249   noprint "final vectors analysis"
1250   vectors input system.movecs
1251 end
1252 task dft
1253 </options>
1254   <cleanup></cleanup>
1255 </package>

```

## 1257 3.6 DFT transfer integrals

list:TI.xml

Listing 3.10: Example TI.xml file created as the output of a DIPRO calculation. Due to slightly different implementations, the orbitals indices refer to monomer indices in a Gaussian run but to indices in the merged dimer guess in a Turbomole run.

```

1258
1259 <pair name="pair_100_155">
1260   <parameters>
1261     <HOMO_A>162</HOMO_A>
1262     <NoccA>1</NoccA>
1263     <LUMO_A>164</LUMO_A>
1264     <NvirtA>1</NvirtA>
1265     <HOMO_B>161</HOMO_B>
1266     <NoccB>1</NoccB>
1267     <LUMO_B>163</LUMO_B>
1268     <NvirtB>1</NvirtB>
1269   </parameters>
1270   <transport name="hole">
1271     <channel name="single">
1272       <J>1.546400416750696E-003</J>
1273       <e_A>-6.30726450715697</e_A>
1274       <e_B>-6.36775613794166</e_B>
1275     </channel>
1276     <channel name="multi">
1277       <molecule name="A">
1278         <e_HOMOm0>-6.30726450715697</e_HOMOm0>
1279       </molecule>
1280       <molecule name="B">
1281         <e_HOMOm0>-6.36775613794166</e_HOMOm0>
1282       </molecule>
1283       <dimer name="integrals">
1284         <T_00>1.546400416750696E-003</T_00>
1285         <J_sq_degen>2.391354248926727E-006</J_sq_degen>
1286         <J_sq_boltz>2.391354248926727E-006</J_sq_boltz>
1287       </dimer>
1288     </channel>
1289   </transport>
1290   <transport name="electron">
1291     <channel name="single">
1292       <J>-2.797473760331286E-003</J>
1293       <e_A>-4.50318366770689</e_A>

```

```

1294     <e_B>-4.53143397059021</e_B>
1295   </channel>
1296   <channel name="multi">
1297     <molecule name="A">
1298       <e_LUMOp0>-4.50318366770689</e_LUMOp0>
1299     </molecule>
1300     <molecule name="B">
1301       <e_LUMOp0>-4.53143397059021</e_LUMOp0>
1302     </molecule>
1303     <dimer name="integrals">
1304       <T_00>-2.797473760331286E-003</T_00>
1305       <J_sq_degen>7.825859439742066E-006</J_sq_degen>
1306       <J_sq_boltz>7.825859439742066E-006</J_sq_boltz>
1307     </dimer>
1308   </channel>
1309 </transport>
1310 </pair>

```

### 3.7 State file

sec:statefile

All data structures are saved to the `state.sql` file in sqlite3 format, see <http://www.sqlite.org/>. They are available in form of tables in the `state.sql` file as can be seen by the command

```
sqlite3 state.sql ".tables "
```

An example of such a table are molecules. The full table can be displayed using the command (similar for the other tables)

```
sqlite3 state.sql " SELECT * FROM molecules "
```

The meaning of all the entries in the table can be displayed by a command like

```
sqlite3 state.sql ".SCHEMA molecules "
```

The first and second entry are integers for internal and regular id of the molecule and the third entry is the name. A single field from the table like the name of the molecule can be displayed by a command like

```
sqlite3 state.sql " SELECT name FROM molecules "
```

Besides molecules, the following tables are stored in the `state.sql`:

conjseg\_properties:  
 Conjugated segments are stored with id, name and x,y,z coordinates of the center of mass in nm.

conjsegs:  
 Reorganization energies for charging or discharging a conjugated segment are stored together with the coulomb energy and any other user defined energy contribution (in eV) and occupation probabilities.

pairs:  
 The pairs from the neighborlist are stored with the pair id, the id of the first and second segment, the rate from the first to the second , the rate from the second to the first (both in s<sup>-1</sup>) and the x,y,z coordinates in nm of the distance between the first and the second segment.

pairintegrals:  
 Transfer integrals for all pairs are stored in the following way: The pair id , the number for counting possible different electronic overlaps (e.g if only the frontier orbitals are taken into account this is always zero, while an effective value is stored in addition to the different overlaps of e.g. HOMO-1 and HOMO-1 if more frontier orbitals are taken into account) and the integral in eV.

pairproperties:  
 The outer sphere reorganization energy of all pairs is stored by an id, the pair id, a string `lambda_outer` and the energy in eV.

conjsegs:  
 Conjugated segments are saved in the following way: The id, the name, the type, the molecule id, the time frame, the x,y,z coordinates in nm and the occupation probability.



```

1347 conjseg_properties:
1348 Properties of the conjugated segments like reorganization energies for charging or discharging a
1349 charge unit or the coulomb contribution to the site energy are stored by: id, conjugated segment
1350 id, a string like lambda_intra_charging, lambda_intra_discharging or energy_coulomb
1351 and a corresponding value in eV.
1352 The tables rigidfrag_properties, rigidfrags and frames offer information about rigid
1353 fragments and time frames including periodic boundary conditions.
1354 The data in the state.sql file can also be modified by the user. Here is an example how to
1355 modify the transfer integral between the conjugated segments number one and two assuming
1356 that they are in the neighborlist. Their pair id can be found by the command
1357 pair_ID=`sqlite3 state.sql "SELECT _id FROM pairs WHERE conjseg1=1 AND conjseg2=2" `
1358 The old value of the transfer integral can be deleted using
1359 sqlite3 state.sql "DELETE FROM pair_integrals WHERE pair=$pair_ID"
1360 Finally the new transfer integral  $J$  can be written to the state.sql file by the command
1361 sqlite3 state.sql "INSERT INTO pair_integrals (pair,num,J) VALUES ($pair_ID,0,$J) "
1362 Here the num=0 indicates that only the effective transfer integrals is written to the file, while other
1363 values of num would correspond to overlap between other orbitals than the frontier orbitals.
1364 In a similar way the coulomb contribution to the site energy of the first conjugated segment can
1365 be overwritten by first getting its id
1366 c_ID=`sqlite3 state.sql "SELECT _id from conjseg_properties where conjseg=1 AND
1367 key =\"energy_coulomb\"" `
1368 Then deleting the old value
1369 sqlite3 state.sql "DELETE FROM from conjseg_properties WHERE _id=$c_ID"
1370 Then the new coulomb energy  $E$  can be written to this id
1371 sqlite3 state.sql "INSERT INTO conjseg_properties (_id,conjseg,key,value)
1372 VALUES ($c_ID,1,\"energy_coulomb\",$E) "
1373 Finally the resulting coulomb contribution to all conjugated segments can be displayed by
1374 sqlite3 state.sql "SELECT * from conjseg_properties WHERE key=\"energy_coulomb\""
1375

```



# 1376 Chapter 4

# 1377 Reference

sec:reference

## 1378 4.1 Programs

sec:programs

1379 Programs execute specific tasks (calculators).

### 1380 4.1.1 xtp\_testsuite

prog:xtp\_testsuite

1381 Performs tests en suite + optional arguments:

1382     -h, --help show this help message and exit  
1383     -e [ [ ...]], --execute [ [ ...]] Tests to perform, accepts regex (def=".\*")  
1384     -l, --listonly List all tests available, then quit.  
1385     -x , --xml Test-suite file (def="\$VOTCASHARE/xtp/xml/testsuite.xml")  
1386     -s , --source Test source input directory (def="source")  
1387     -td , --testdirectory Test run directory (def="suite")  
1388     -t , --target Directory where to store targets (def="targets")  
1389     -r , --reference Folder with reference data to compare to (def="reference")  
1390     -g, --generate Generate reference from targets (def=False)  
1391     -cmp, --compareonly Only compare existing targets to reference (def=False)  
1392     -v, --verbose The wordy version (def=False)  
1393     -sh, --showoutput Display VOTCA::XTP exec. output (def=False)  
1394     -c, --clean To clean or not to clean test dir. (def=False)  
1395     -m , --mailto Mail the result. (def=False)

### 1396 4.1.2 xtp\_update

prog:xtp\_update

1397 Updates the state file + optional arguments:

1398     -h, --help show this help message and exit  
1399     -f SQLFILE, --file SQLFILE State file to update.

### 1400 4.1.3 xtp\_update\_exciton

prog:xtp\_update\_exciton

1401 Updates the state file for singlets and triplets + optional arguments:

1402     -h, --help show this help message and exit  
1403     -f SQLFILE, --file SQLFILE State file to update.

### 1404 4.1.4 xtp\_basisset

prog:xtp\_basisset

1405 xtp\_update, version 1.4.1 Creates votca xml basissetfiles from NWCHEM basissetfiles optional  
1406 arguments:

```

1407     -h, --help show this help message and exit
1408     -f NWCHEM, --inputnw NWCHEM NWchem file containing the basisset.
1409     -o OUTPUTFILE, --outputvotca OUTPUTFILE Path of votca outputfile

```

#### 1410 4.1.5 xtp\_map

prog:xtp\_map

```

1411 Generates QM|MD topology
1412     -h [ --help ] display this help and exit
1413     -v [ --verbose ] be loud and noisy
1414     -t [ --topology ] arg topology
1415     -c [ --coordinates ] arg coordinates or trajectory
1416     -s [ --segments ] arg definition of segments and fragments
1417     -f [ --file ] arg state file

```

#### 1418 4.1.6 xtp\_run

prog:xtp\_run

```

1419 Runs excitation/charge transport calculators
1420     -h [ --help ] display this help and exit
1421     -v [ --verbose ] be loud and noisy
1422     -o [ --options ] arg calculator options
1423     -f [ --file ] arg sqlight state file, *.sql
1424     -i [ --first-frame ] arg (=1) start from this frame
1425     -n [ --nframes ] arg (=1) number of frames to process
1426     -t [ --nthreads ] arg (=1) number of threads to create
1427     -s [ --save ] arg (=1) whether or not to save changes to state file
1428     -e [ --execute ] arg List of calculators separated by ',' or ''
1429     -l [ --list ] Lists all available calculators
1430     -d [ --description ] arg Short description of a calculator

```

#### 1431 4.1.7 xtp\_tools

prog:xtp\_tools

```

1432 Runs excitation/charge transport tools
1433     -h [ --help ] display this help and exit
1434     -v [ --verbose ] be loud and noisy
1435     -t [ --nthreads ] arg (=1) number of threads to create
1436     -o [ --options ] arg calculator options Tools:
1437     -e [ --execute ] arg List of tools separated by ',' or ''
1438     -l [ --list ] Lists all available tools
1439     -d [ --description ] arg Short description of a tool

```

#### 1440 4.1.8 xtp\_parallel

prog:xtp\_parallel

```

1441 Runs job-based heavy-duty calculators
1442     -h [ --help ] display this help and exit
1443     -v [ --verbose ] be loud and noisy
1444     -o [ --options ] arg calculator options
1445     -f [ --file ] arg sqlite state file, *.sql
1446     -i [ --first-frame ] arg (=1) start from this frame
1447     -n [ --nframes ] arg (=1) number of frames to process
1448     -t [ --nthreads ] arg (=1) number of threads to create
1449     -s [ --save ] arg (=1) whether or not to save changes to state file
1450     -r [ --restart ] arg restart pattern: 'host(pc1:234) stat(FAILED)'
1451     -c [ --cache ] arg (=8) assigns jobs in blocks of this size
1452     -j [ --jobs ] arg (=run) task(s) to perform: input, run, import

```

```

1453     -m [ --maxjobs ] arg (=-1) maximum number of jobs to process (-1 = inf)
1454     -e [ --execute ] arg List of calculators separated by ',' or ''
1455     -l [ --list ] Lists all available calculators
1456     -d [ --description ] arg Short description of a calculator

```

#### 1457 4.1.9 xtp\_dump

prog:xtp\_dump

```

1458 Extracts information from the state file
1459     -h [ --help ] display this help and exit
1460     -v [ --verbose ] be loud and noisy
1461     -o [ --options ] arg calculator options
1462     -f [ --file ] arg sqlight state file, *.sql
1463     -i [ --first-frame ] arg (=1) start from this frame
1464     -n [ --nframes ] arg (=1) number of frames to process
1465     -t [ --nthreads ] arg (=1) number of threads to create
1466     -s [ --save ] arg (=1) whether or not to save changes to state file Extractors:
1467     -e [ --extract ] arg List of extractors separated by ',' or ''
1468     -l [ --list ] Lists all available extractors
1469     -d [ --description ] arg Short description of an extractor

```

#### 1470 4.1.10 xtp\_overlap

prog:xtp\_overlap

```

1471 moo_overlap
1472     -h [ --help ] display this help and exit
1473     -v [ --verbose ] be loud and noisy MOO Options:
1474     --conjseg arg xml file describing two conjugated segments
1475     --pos1 arg position and orientation of molecule 1
1476     --pos2 arg position and orientation of molecule 2
1477     --pdb arg (=geometry.pdb) pdb file of two molecules

```

#### 1478 4.1.11 xtp\_kmc\_run

prog:xtp\_kmc\_run

```

1479 kmc_run, version 1.4.1 (compiled Sep 2 2017, 10:55:55) Runs specified calculators
1480     -h [ --help ] display this help and exit
1481     -v [ --verbose ] be loud and noisy
1482     -o [ --options ] arg program and calculator options
1483     -f [ --file ] arg sqlite state file
1484     -t [ --textfile ] arg output text file (otherwise: screen output)
1485     -e [ --execute ] arg list of calculators separated by commas or spaces
1486     -l [ --list ] lists all available calculators
1487     -d [ --description ] arg detailed description of a calculator

```

## 1488 4.2 Calculators

sed:calculators

1489 Calculator is a piece of code which computes specific system properties, such as site energies,  
 1490 transfer integrals, etc. `xtp_run`, `xtp_kmc_run` are wrapper programs which executes such  
 1491 calculators. The generic syntax is

```
1492 xtp_run -e "calc1, calc2, ..." -o options.xml
```

1493 File `options.xml` lists all options needed to run a specific calculator. The format of this file is  
 1494 explained in listing 4.1. A complete list of calculators is given in the `calculators` reference section.

list:calc

Listing 4.1: A part of the `options.xml` file with options for the `calculator_name{1,2}` calculators.

```

1495
1496 <calculator_name1>
1497     <option1>value1</option1>
1498     <option2>value2</option2>
1499     ...
1500 </calculator_name1>
1501
1502 <calculator_name2>
1503     <option1>value1</option1>
1504     <option2>value2</option2>
1505     ...
1506 </calculator_name2>
1507 ...

```

A list of all calculators and their short descriptions can be obtain using

```
xtp_run --list
```

A detailed description of all options of a specific calculator(s) is available via

```
xtp_run --desc calc1,calc2,...
```

## 4.2.1 coupling

calc:coupling

Electronic couplings from log and orbital files (GAUSSAIN, TURBOMOLE, NWChem)

1513

1514

option	default	unit	description
dftpackage			First-principles package
output	coupling.out.xml		Output file
degeneracy	0	eV	Criterion for the degeneracy of two levels
moleculeA			
log	A.log		Log file of molecule A
orbitals	A.orb		Orbitals file
levels	3		Output HOMO, ..., HOMO-levels; LUMO, ..., LUMO+levels
trim	2		
moleculeB			
log	B.log		Log file of molecule B
orbitals	B.orb		Orbitals file
levels	3		Output HOMO, ..., HOMO-levels; LUMO, ..., LUMO+levels
trim	2		
dimerAB			
log	AB.log		Log file of dimer AB
orbitals	A.orb		Orbitals file

Return to the description of `coupling`.

## 4.2.2 excitoncoupling

calc:excitoncoupling

Exciton couplings from serialized orbital files

1516

1517

option	default	unit	description
classical			
output	excitoncoupling		Output file
bsecoupling_options			
orbitalsA	A.orb		Serialized orbitals file
orbitalsB	B.orb		Serialized orbitals file
orbitalsAB	AB.orb		Serialized orbitals file

1518 Return to the description of `excitoncoupling`.

### 1519 4.2.3 `gencube`

`calc:gencube`

1520 Tool to generate cube files from `.orb` file

option	default	unit	description
<code>output</code>	<code>state.cube</code>		Output file
<code>input</code>	<code>system.orb</code>		Input file
<code>padding</code>	6.5		How far the grid should start from the molecule
<code>xsteps</code>	25		Gridpoints in x-direction
<code>ysteps</code>	25		Gridpoints in y-direction
<code>zsteps</code>	25		Gridpoints in z-direction
<code>state</code>	1		State to generate cube file for
<code>spin</code>			Singlet or Triplet
<code>type</code>	<code>ground</code>		qp:quasiparticle,ground:groundstate,transition:transitionstate,excited/excitedstate density/density excited-ground state
<code>mode</code>	<code>new</code>		new: generate new cube file, subtract: subtract to cube files specified below
<code>infile1</code>			Cubefile to subtract infile2 from
<code>infile2</code>			Cubefile to subtract from infile1

1521 Return to the description of `gencube`.

### 1522 4.2.4 `log2mps`

`calc:log2mps`

1523 Generates an `mps`-file (with polar-site definitions) from a QM log-file

option	default	unit	description
<code>package</code>			QM package
<code>logfile</code>			Log-file generated by QM package, with population/esp-fit data

1524 Return to the description of `log2mps`.

### 1525 4.2.5 `molpol`

`calc:molpol`

1526 Molecular polarizability calculator (and optimizer)

option	default	unit	description
<code>mpsfiles</code>			
<code>input</code>			mps input file
<code>output</code>			mps output file
<code>polar</code>			xml file with infos on polarizability tensor
<code>induction</code>			
<code>expdamp</code>			Thole sharpness parameter
<code>wSOR</code>			mixing factor for convergence
<code>maxiter</code>			maximum number of iterations
<code>tolerance</code>			rel. tolerance for induced moments
<code>target</code>			
<code>optimize</code>			if 'true', refine atomic polarizabilities to match molecular polarizable volume specified in <code>target.molpol</code>
<code>molpol</code>			target polarizability tensor in format <code>xx xy xz yy yz zz</code> (this should be in the eigen-frame, hence <code>xy = xz = yz = 0</code> ), if <code>optimize=true</code> the associated polarizable volume will be matched iteratively and the resulting set of polar sites written to <code>mpsfiles.output</code>

tolerance			relative tolerance when optimizing the polarizable volume
-----------	--	--	---

1527 Return to the description of `molpol`.

#### 1528 4.2.6 orb2isogwa

calc:orb2isogwa

1529 Analysis tool for QM results stores in serialized file

option	default	unit	description
output	qmanalyze.out		Output file
property			
input	molecule.orb		Serialized file

1530 Return to the description of `orb2isogwa`.

#### 1531 4.2.7 partialcharges

calc:partialcharges

1532 Tool to derive partial charges from QM results stores in serialized file

option	default	unit	description
output	Moleculecharge		Output file either .mps or .pdb
input	molecule.orb		Serialized file
esp_options			options for the method

1533 Return to the description of `partialcharges`.

#### 1534 4.2.8 pdb2map

calc:pdb2map

1535 Converts MD + QM files to VOTCA mapping. Combinations: `pdb+xyz,gro+xyz,pdb`

option	default	unit	description
pdb	conf.pdb		Input pdb file
gro	conf.gro		Input gro file
xyz	conf.xyz		Input xyz file
xml	conf.xml		Resulting xml file

1536 Return to the description of `pdb2map`.

#### 1537 4.2.9 pdb2top

calc:pdb2top

1538 Generates fake Gromacs topology file `.top`

option	default	unit	description
num	1		Num of mols in the box
pdb	conf.pdb		Input pdb file
gro	conf.gro		Input gro file

1539 Return to the description of `pdb2top`.

#### 1540 4.2.10 ptopreader

calc:ptopreader

1541 Reads binary `.ptop`-files (serialized from `ewdbgpol`) and processes them into something readable

option	default	unit	description
ptop_file			Binary archive <code>.ptop</code> -file



1542 Return to the description of `ptopreader`.

### 1543 4.2.11 `qmanalyze`

1544 `calc:qmanalyze` Analysis tool for QM results stores in serialized file

option	default	unit	description
output	qmanalyze.out		Output file
BSE			additional info about BSE results
input	molecule.orb		Serialized file

1545 Return to the description of `qmanalyze`.

### 1546 4.2.12 `eanalyze`

1547 `calc:eanalyze` Histogram and correlation function of site energies and pair energy differences

option	default	unit	description
resolution_sites		eV	Bin size for site energy histogram
resolution_pairs		eV	Bin size for pair energy histogram
resolution_space		eV	Bin size for site energy correlation
states			?

1548 Return to the description of `eanalyze`.

### 1549 4.2.13 `eimport`

1550 `calc:eimport` Imports site energies from the output file of `emultipole` and writes them to the state file

option	default	unit	description
--------	---------	------	-------------

1551 Return to the description of `eimport`.

### 1552 4.2.14 `einternal`

1553 `calc:einternal` Reads in site and reorganosation energies and writes them to the state file

option	default	unit	description
energiesXML			XML input file with vacuum site, reorganization (charging, discharging) energies

1554 Return to the description of `einternal`.

### 1555 4.2.15 `emultipole`

1556 `calc:emultipole` Evaluates polarization contribution based on the Thole model

option	default	unit	description
multipoles			Polar Site Definitions in GDMA punch-file format
control			Control options for induction computation
induce	1		Enter '1' / '0' to toggle induction on / off
first			First segment for which to compute site energies
last			Last segment for which to compute site energies
output			File to write site energies to. Site energies are also stored in the state file
check			Check mapping of polar sites to fragment

tholeparam			Thole parameters required for charge-smearing
cutoff		nm	Cut-off beyond which all interactions are neglected
cutoff2		nm	Cut-off beyond which polarization is neglected
expdamp			Damping exponent used in exponential damping function
scaling			1-n interaction scaling, currently not in use
esp			Control options for potential calculation
calcESP			Enter '1' / '0' to toggle on / off. If '1', site energies will not be evaluated
cube			
grid			XYZ file specifying grid points for potential evaluation
output			File to write grid-point potential to
esf			Control options for field calculation
calcESF			Enter '1' / '0' to toggle on / off. If '1', site energies will not be evaluated
grid			XYZ file specifying grid points for field evaluation
output			File to write grid-point field to
alphamol			Control options for molecular-polarizability calculation
calcAlpha			Enter '1' / '0' to toggle on / off. If '1', site energies will not be evaluated
output			File to write polarizability tensor in global frame and in diagonal form to
convparam			Convergence parameters for self-consistent field calculation
wSOR_N			Mixing factor for successive overrelaxation of neutral system, usually between 0.3 and 0.5
wSOR_C			Mixing factor for successive overrelaxation of charged system, usually between 0.3 and 0.5
tolerance			Convergence criterion, fulfilled if relative change smaller than tolerance
maxiter			Maximum number of iterations in the convergence loop

1557 [Return to the description of `emultipole`.](#)

#### 1558 4.2.16 `eoutersphere`

`calc:eoutersphere`

1559 Evaluates outersphere reorganization energy

option	default	unit	description
multipoles			XML allocation polar sites
method			Type of the method: <code>constant</code> - all pairs have value <code>lambda</code> . <code>spheres</code> - molecules are treated as spheres with radii <code>radius</code> and Pekar factor <code>pekar</code> . <code>dielectric</code> - with Pekar factor <code>pekar</code> and partial charges from resulting dielectric fields
lambdaconst		eV	The value for all pairs in the <code>constant</code> method
pekar			Pekar factor used for methods <code>spheres</code> and <code>dielectric</code>
segment			
type			
radius			
segment			
type			
radius			
cutoff		nm	Cutoff radius in between pair and the exterior molecule. Can be used in <code>spheres</code> and <code>dielectric</code>

1560 [Return to the description of `eoutersphere`.](#)

1561 **4.2.17 ianalyze**

1562 `calc:ianalyze` Evaluates a histogram of a logarithm of squared couplings

option	default	unit	description
resolution_logJ2			Bin size of histogram $\log(J^2)$
resolution_space		nm	Bin size for $r$ in $\log(J^2(r))$
states			States for which to calculate the histogram. Example: 1 -1

1563 Return to the description of `ianalyze`.

1564 **4.2.18 iimport**

1565 `calc:iimport` Imports electronic couplings from xml of xtp-dipro using folders of pairdump

option	default	unit	description
idft_jobs_file			idft jobs file
probabilityfile_h	ianalyze.ispatial.h.out		For coarse grained simulations provide here the distance dependent means and sigmas of hole transfer integrals. This file can be created using the <code>ianalyze</code> calculator.
probabilityfile_e	ianalyze.ispatial.e.out		For coarse grained simulations provide here the distance dependent means and sigmas of electron transfer integrals. This file can be created using the <code>ianalyze</code> calculator.

1566 Return to the description of `iimport`.

1567 **4.2.19 izindo**

1568 `calc:izindo` Semiempirical electronic coupling elements for all neighbor list pairs

option	default	unit	description
orbitalsXML			File with paths to .orb files

1569 Return to the description of `izindo`.

1570 **4.2.20 jobwriter**

1571 `calc:jobwriter` Writes list of jobs for a parallel execution

option	default	unit	description
keys			job type
single_id			Segment ID as argument for <code>mps.single</code>
kmc_cutoff		nm	Pair-interaction cut-off as argument for <code>mps.kmc</code>

1572 Return to the description of `jobwriter`.

1573 **4.2.21 pairdump**

1574 `calc:pairdump` Coordinates of molecules and pairs from the neighbor list

option	default	unit	description
molecules			If <code>**true**</code> outputs single molecules, otherwise only pairs

1575 Return to the description of `pairdump`.

### 1576 4.2.22 `panalyze`

1576  
1577 `calc:panalyze`

Probability of neighbours being in the pair list as a function of their centre of mass distance

option	default	unit	description
resolution_space		nm	Spatial resolution for the probability function.

1578 Return to the description of `panalyze`.

### 1579 4.2.23 `profile`

1579  
1580 `calc:profile`

Density and site energy profiles

option	default	unit	description
axis			Axis along which to calculate density and energy profiles
direction	0 0 1		Axis direction
min		nm	Minimal projected position for manual binning
max		nm	Maximal projected position for manual binning
bin	0.1	nm	Spatial resolution of the profile
auto	1		'0' for manual binning using min and max, '1' for automated
particles			
type	segments		What centers of mass to use: 'segments' or 'atoms'
first	1		ID of the first segment
last	-1		ID of the last segment, -1 is the list end
output			
density	density.dat		Density profile file
energy	energy.dat		Energy profile file

1581 Return to the description of `profile`.

### 1582 4.2.24 `rates`

1582  
1583 `calc:rates`

Hopping rates using classical or semi-classical expression

option	default	unit	description
field			Field in x y z direction
temperature		K	Temperature for rates
method			Method chosen to compute rates. Can either be <code>**marcus**</code> or <code>**jortner**</code> . The first is the high temperature limit of Marcus theory, the second is the rate proposed by Jortner and Bixon
nmaxvib	20		If the method of choice is <code>**jortner**</code> , the maximal number of excited vibrations on the molecules has to be specified as an integer for the summation
omegavib	0.2	eV	If the method of choice is <code>**jortner**</code> , the vibration frequency of the quantum mode has to be given in units of eV. The default value is close to the CC bond-stretch at 0.2eV

1584 Return to the description of `rates`.

### 1585 4.2.25 `sandbox`

1585  
1586 `calc:sandbox`

Sandbox to test xtp classes

option	default	unit	description
--------	---------	------	-------------

ID			Not in use
----	--	--	------------

1587 Return to the description of `sandbox`.

#### 1588 **4.2.26 stateserver**

calc:stateserver

1589 Export SQLite file to human readable format

option	default	unit	description
out			Output file name
pdb			PDB coordinate file name
keys			Sections to write to readable format (topology, segments, pairs, coordinates)

1590 Return to the description of `stateserver`.

#### 1591 **4.2.27 tdump**

calc:tdump

1592 Coarse-grained and back-mapped (using rigid fragments) trajectories

option	default	unit	description
md	MD.pdb		Name of the coarse-grained trajectory
qm	QM.pdb		Name of the trajectory with back-substituted rigid fragments
frames	1		Number of frames to output

1593 Return to the description of `tdump`.

#### 1594 **4.2.28 vaverage**

calc:vaverage

1595 Computes site-centered velocity averages from site occupancies

option	default	unit	description
carriers			Carrier types for which to compute velocity averages
tabulate			Tabulate 'atoms' or 'segments'

1596 Return to the description of `vaverage`.

#### 1597 **4.2.29 zmultipole**

calc:zmultipole

1598 Evaluates polarization contribution based on the Thole model

option	default	unit	description
multipoles			Polar Site Definitions in GDMA punch-file format
control			Control options for induction computation
induce	1		Enter '1' / '0' to toggle induction on / off
first			First segment for which to compute site energies
last			Last segment for which to compute site energies
output			File to write site energies to. Site energies are also stored in the state file
check			Check mapping of polar sites to fragment
tholeparam			Thole parameters required for charge-smearing
cutoff		nm	Cut-off beyond which all interactions are neglected
cutoff2		nm	Cut-off beyond which polarization is neglected
expdamp			Damping exponent used in exponential damping function

scaling			1-n interaction scaling, currently not in use
esp			Control options for potential calculation
calcESP			Enter '1' / '0' to toggle on / off. If '1', site energies will not be evaluated
cube			
grid			XYZ file specifying grid points for potential evaluation
output			File to write grid-point potential to
esf			Control options for field calculation
calcESF			Enter '1' / '0' to toggle on / off. If '1', site energies will not be evaluated
grid			XYZ file specifying grid points for field evaluation
output			File to write grid-point field to
alphamol			Control options for molecular-polarizability calculation
calcAlpha			Enter '1' / '0' to toggle on / off. If '1', site energies will not be evaluated
output			File to write polarizability tensor in global frame and in diagonal form to
convparam			Convergence parameters for self-consistent field calculation
wSOR_N			Mixing factor for successive overrelaxation of neutral system, usually between 0.3 and 0.5
wSOR_C			Mixing factor for successive overrelaxation of charged system, usually between 0.3 and 0.5
tolerance			Convergence criterion, fulfilled if relative change smaller than tolerance
maxiter			Maximum number of iterations in the convergence loop

1599 [Return to the description of `zmultipole`.](#)

### 1600 **4.2.30 edft**

calc:edft

1601 A wrapper for first principles based single site calculations

option	default	unit	description
tasks	input,run,parse		What to run
store	orbitals		What to store

1602 [Return to the description of `edft`.](#)

### 1603 **4.2.31 idft**

calc:idft

1604 Projection method for electronic couplings. Requires edft output

option	default	unit	description
tasks	input,run,parse		What to do
store	orbitals,overlap		What to store
degeneracy	0	eV	Criterion for the degeneracy of two levels
levels	3		Output between HOMO, ..., HOMO-levels; LUMO, ..., LUMO+levels
trim	2		Use trim*occupied of virtual orbitals

1605 [Return to the description of `idft`.](#)

### 1606 **4.2.32 qmmm**

calc:qmmm

1607 QM/MM with the Thole MM model

option	default	unit	description
pdb_check			PDB file of polar sites
write_chk	dipoles.xyz		XYZ file with dipoles split onto point charges
format_chk	xyz		format, gaussian or xyz
split_dpl	1		'0' do not split dipoles onto point charges, '1' do split
dpl_spacing	1e-3	nm	Spacing to be used when splitting dipole onto point charges: $d = q * a$
dftpackage			DFT package to use for the QM region
gwbse			Specify if GW/BSE excited state calculation ist needed
gwbse_options			GW/BSE options file
state			Number of excited state, which is to be calculated
type			Character of the excited state to be calculated
filter			Filter with which to find the excited state after each calculation
oscillator_strength			Oscillator strength filter, only states with higher oscillator strength are considered
charge_transfer			Charge transfer filter , only states with charge transfer above threshold are considered
qmmmconvg			convergence criteria for the QM/MM
dR	0.001	nm	RMS of coordinates
dQ	0.001	e	RMS of charges
dE_QM	0.0001	eV	Energy change of the QM region
dE_MM	0.0001	eV	Energy change of the MM region
max_iter	10		Number of iterations
coulombmethod			Options for the MM embedding
method	cut-off		Method for evaluation of electrostatics
cutoff1			Cut-off for the polarizable MM1 shell
cutoff2			Cut-off for the static MM2 shell
tholemodel			Parameters for teh Thole model
induce			'1' - induce '0' - no induction
induce_intra_pair			'1' - include mutual interaction of induced dipoles in the QM region. '0' - do not
exp_damp	0.39		Sharpness parameter
scaling			Bond scaling factors
convergence			Convergence parameters for the MM1 (polarizable) region
wSOR_N			Mixing factor for the successive overrelaxation algorithm for a neutral QM region
wSOR_C			Mixing factor for the successive overrelaxation algorithm for a charged QM region
max_iter	512		Maximal number of iterations to converge induced dipoles
tolerance			Maximum RMS change allowed in induced dipoles

1608 Return to the description of qmmm.

### 1609 4.2.33 xqmultipole

calc:xqmultipole

1610 Electrostatic interaction and induction energy of charged molecular clusters

option	default	unit	description
mapping			Polar-site mapping definition
job_file			Job file
emp_file			Polar-background definition, allocation of mps-files to segments
pdb_check			Whether or not to output a pdb-file of the mapped polar sites
format_chk			Format for check-file: 'xyz' or 'gaussian'

split_dpl			Split dipoles onto point charges in check-file
dpl_spacing		nm	Spacing between point charges for check-file output
coulombmethod			
method			Currently only cut-off supported
cutoff1		nm	Full-interaction radius cut-off
cutoff2		nm	Radius of electrostatic buffer
tholemodel			
induce			Induce - or not
induce_intra_pair			Induce mutually within the charged cluster
exp_damp			Thole sharpness parameter
scaling			Bond scaling parameters, currently not used
convergence			
wSOR_N			SOR mixing factor for overall neutral clusters
wSOR_C			SOR mixing factor for overall charged clusters
max_iter			Maximum number of iterations
tolerance			Relative tolerance as convergence criterion

1611 Return to the description of `xqmultipole`.

#### 1612 4.2.34 `energy2xml`

`calc:energy2xml`

1613 Write out energies from SQL file

option	default	unit	description
--------	---------	------	-------------

1614 Return to the description of `energy2xml`.

#### 1615 4.2.35 `integrals2xml`

`calc:integrals2xml`

1616 Write out transfer integrals from SQL file

option	default	unit	description
--------	---------	------	-------------

1617 Return to the description of `integrals2xml`.

#### 1618 4.2.36 `occupations2xml`

`calc:occupations2xml`

1619 Write out site occupation probabilities from SQL file

option	default	unit	description
--------	---------	------	-------------

1620 Return to the description of `occupations2xml`.

#### 1621 4.2.37 `pairs2xml`

`calc:pairs2xml`

1622 Write out neighbourlist from SQL file

option	default	unit	description
--------	---------	------	-------------

1623 Return to the description of `pairs2xml`.

#### 1624 4.2.38 `rates2xml`

`calc:rates2xml`

1625 Write out charge transfer rates from SQL file

option	default	unit	description
--------	---------	------	-------------



1626 Return to the description of `rates2xml`.

#### 1627 **4.2.39 segments2xml**

`calc:segments2xml`

1628 Write out segment data from SQL file

option	default	unit	description
--------	---------	------	-------------

1629 Return to the description of `segments2xml`.

#### 1630 **4.2.40 trajectory2pdb**

`calc:trajectory2pdb`

1631 Generate PDB files for the mapped MD/QM topology

option	default	unit	description
--------	---------	------	-------------

1632 Return to the description of `trajectory2pdb`.

### 1633 **4.3 Common options**

`ref:options`

name	Description of the option
------	---------------------------



# Bibliography

- ruhle\_microscopic\_2011 [1] V. RÅijhle *et al.*, J. Chem. Theory Comput. **7**, 3335 (2011), bibtex: ruhle\_microscopic\_2011. [ii](#), [3](#), [19](#), [20](#)
- ruhle\_versatile\_2009 [2] V. RÅijhle *et al.*, J. Chem. Theory Comput. **5**, 3211 (2009), bibtex: ruhle\_versatile\_2009. [ii](#), [3](#), [6](#)
- gamma\_design\_1995 [3] E. Gamma, R. Helm, and R. E. Johnson, *Design Patterns. Elements of Reusable Object-Oriented Software.*, 1st ed., reprint. ed. (Addison-Wesley Longman, Amsterdam, ADDRESS, 1995), bibtex: gamma\_design\_1995. [3](#)
- ruhle\_multiscale\_2010 [4] V. RÅijhle, J. Kirkpatrick, and D. Andrienko, J. Chem. Phys. **132**, 134103 (2010), bibtex: ruhle\_multiscale\_2010. [7](#)
- vukmirovic\_charge\_2008 [5] N. VukmiroviÄŒ and L.-W. Wang, J. Chem. Phys. **128**, 121102 (2008), bibtex: vukmirovic\_charge\_2008. [7](#)
- vukmirovic\_charge\_2009 [6] N. VukmiroviÄŒ and L.-W. Wang, Nano Letters **9**, 3996 (2009), bibtex: vukmirovic\_charge\_2009.
- mcmahon\_ad\_2009 [7] D. P. McMahon and A. Troisi, Chem. Phys. Lett. **480**, 210 (2009), bibtex: mcmahon\_ad\_2009. [7](#)
- bredas\_charge-transfer\_2004 [8] J.-L. BrÅÍdas, D. Beljonne, V. Coropceanu, and J. Cornil, Chem. Rev. **104**, 4971 (2004), bibtex: bredas\_charge-transfer\_2004. [8](#), [19](#)
- may\_relationship\_2011 [9] F. May *et al.*, J. Mater. Chem. **21**, 9538 (2011), bibtex: may\_relationship\_2011. [9](#), [20](#)
- breneman\_determining\_1990 [10] C. M. Breneman and K. B. Wiberg, J. Comput. Chem. **11**, 361 (1990), bibtex: breneman\_determining\_1990. [11](#)
- stone\_distributed\_1985 [11] A. Stone and M. Alderton, Molecular Physics **56**, 1047 (1985), bibtex: stone\_distributed\_1985. [11](#), [12](#)
- chirlian\_atomic\_1987 [12] L. E. Chirlian and M. M. Francl, J. Comput. Chem. **8**, 894 (1987), bibtex: chirlian\_atomic\_1987. [11](#)
- singh\_approach\_1984 [13] U. C. Singh and P. A. Kollman, J. Comput. Chem. **5**, 129 (1984), bibtex: singh\_approach\_1984. [11](#)
- stone\_distributed\_2005 [14] A. J. Stone, J. Chem. Theory Comput. **1**, 1128 (2005), bibtex: stone\_distributed\_2005. [12](#)
- stone\_theory\_1997 [15] A. J. Stone, *The Theory of intermolecular forces* (Clarendon Press, Oxford, 1997), bibtex: stone\_theory\_1997. [13](#)
- thole\_molecular\_1981 [16] B. Thole, Chem. Phys. **59**, 341 (1981), bibtex: thole\_molecular\_1981. [13](#)
- van\_duijnen\_molecular\_1998 [17] P. T. van Duijnen and M. Swart, J. Phys. Chem. A **102**, 2399 (1998), bibtex: van\_duijnen\_molecular\_1998-1. [13](#), [14](#)
- applequist\_atom\_1972 [18] J. Applequist, J. R. Carl, and K.-K. Fung, J. Am. Chem. Soc. **94**, 2952 (1972), bibtex: applequist\_atom\_1972. [13](#)
- ren\_polarizable\_2003 [19] P. Ren and J. W. Ponder, J. Phys. Chem. B **107**, 5933 (2003), bibtex: ren\_polarizable\_2003. [13](#), [14](#)
- baessler\_charge\_1993 [20] H. Baessler, Phys. Status Solidi B **175**, 15 (1993), bibtex: baessler\_charge\_1993. [15](#), [19](#)
- troisi\_charge-transport\_2006 [21] A. Troisi and G. Orlandi, Phys. Rev. Lett. **96**, (2006), bibtex: troisi\_charge-transport\_2006.
- troisi\_charge\_2009 [22] A. Troisi, D. L. Cheung, and D. Andrienko, Phys. Rev. Lett. **102**, 116602 (2009), bibtex: troisi\_charge\_2009.
- mcmahon\_organic\_2010 [23] D. P. McMahon and A. Troisi, ChemPhysChem **11**, 2067 (2010), bibtex: mcmahon\_organic\_2010.
- vehoff\_charge\_2010 [24] T. Vehoff *et al.*, J. Phys. Chem. C **114**, 10592 (2010), bibtex: vehoff\_charge\_2010. [15](#)
- baumeier\_density-functional\_2010 [25] B. Baumeier, J. Kirkpatrick, and D. Andrienko, Phys. Chem. Chem. Phys. **12**, 11103 (2010), bibtex: baumeier\_density-functional\_2010. [15](#), [16](#), [19](#)
- kirkpatrick\_approximate\_2008 [26] J. Kirkpatrick, Int. J. Quantum Chem. **108**, 51 (2008), bibtex: kirkpatrick\_approximate\_2008. [19](#)
- walker\_electrical\_2002 [27] A. B. Walker, A. Kambili, and S. J. Martin, J. Phys-Condens. Mat. **14**, 9825 (2002), bibtex: walker\_electrical\_2002. [19](#)

- [borsenberger\\_charge\\_1991](#) [28] P. M. Borsenberger, L. Pautmeier, and H. BÄd’ssler, *J. Chem. Phys.* **94**, 5447 (1991), bibtex: borsenberger\_charge\_1991.
- [pasveer\\_unified\\_2005](#) [29] W. F. Pasveer *et al.*, *Phys. Rev. Lett.* **94**, 206601 (2005), bibtex: pasveer\_unified\_2005. [19](#)
- [bredas\\_molecular\\_2009](#) [30] J.-L. BrÄYdas, J. E. Norton, J. Cornil, and V. Coropceanu, *Accounts Chem. Res.* **42**, 1691 (2009), bibtex: bredas\_molecular\_2009. [19](#)
- [coropceanu\\_charge\\_2007](#) [31] V. Coropceanu *et al.*, *Chem. Rev.* **107**, 926 (2007), bibtex: coropceanu\_charge\_2007.
- [nelson\\_modeling\\_2009](#) [32] J. Nelson, J. J. Kwiatkowski, J. Kirkpatrick, and J. M. Frost, *Accounts Chem. Res.* **42**, 1768 (2009), bibtex: nelson\_modeling\_2009. [19](#)
- [marcus\\_electron\\_1993](#) [33] R. A. Marcus, *Rev. Mod. Phys.* **65**, 599 (1993), bibtex: marcus\_electron\_1993. [19](#)
- [hutchison\\_hopping\\_2005](#) [34] G. R. Hutchison, M. A. Ratner, and T. J. Marks, *J. Am. Chem. Soc.* **127**, 2339 (2005), bibtex: hutchison\_hopping\_2005. [19](#)
- [chang\\_new\\_2005](#) [35] J.-L. Chang, *J. Mol. Spectrosc.* **232**, 102 (2005), bibtex: chang\_new\_2005. [20](#)
- [hoffman\\_reorganization\\_1996](#) [36] B. M. Hoffman and M. A. Ratner, *Inorg. Chim. Acta.* **243**, 233 (1996), bibtex: hoffman\_reorganization\_1996. [20](#)
- [bortz\\_new\\_1975](#) [37] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz, *J. Comput. Phys.* **17**, 10 (1975), bibtex: bortz\_new\_1975. [21](#)
- [scher\\_anomalous\\_1975](#) [38] H. Scher and E. Montroll, *Phys. Rev. B* **12**, 2455 (1975), bibtex: scher\_anomalous\_1975. [21](#)
- [borsenberger\\_role\\_1993](#) [39] P. M. Borsenberger, E. H. Magin, M. D. VanAuweraer, and F. C. D. Schryver, *Phys. Status Solidi A* **140**, 9 (1993), bibtex: borsenberger\_role\_1993. [21](#)
- [derrida\\_velocity\\_1983](#) [40] B. Derrida, *J. Stat. Phys.* **31**, 433 (1983), bibtex: derrida\_velocity\_1983. [21](#), [30](#)
- [cordes\\_one-dimensional\\_2001](#) [41] H. Cordes *et al.*, *Phys. Rev. B* **63**, 094201 (2001), bibtex: cordes\_one-dimensional\_2001.
- [seki\\_electric\\_2001](#) [42] K. Seki and M. Tachiya, *Phys. Rev. B* **65**, 014305 (2001), bibtex: seki\_electric\_2001. [21](#)
- [lukyanov\\_extracting\\_2010](#) [43] A. Lukyanov and D. Andrienko, *Phys. Rev. B* **82**, 193202 (2010), bibtex: lukyanov\_extracting\_2010. [22](#)
- [van\\_der\\_holst\\_modeling\\_2009](#) [44] J. J. M. van der Holst *et al.*, *Phys. Rev. B* **79**, 085203 (2009), bibtex: van\_der\_holst\_modeling\_2009. [29](#)
- [dunlap\\_charge-dipole\\_1996](#) [45] D. Dunlap, P. Parris, and V. Kenkre, *Phys. Rev. Lett.* **77**, 542 (1996), bibtex: dunlap\_charge-dipole\_1996. [30](#)
- [novikov\\_essential\\_1998](#) [46] S. V. Novikov *et al.*, *Phys. Rev. Lett.* **81**, 4472 (1998), bibtex: novikov\_essential\_1998. [30](#)
- [nagata\\_atomistic\\_2008](#) [47] Y. Nagata and C. Lennartz, *J. Chem. Phys.* **129**, 034709 (2008), bibtex: nagata\_atomistic\_2008. [30](#)
- [novikov\\_cluster\\_1995](#) [48] S. V. Novikov and A. V. Vannikov, *J. Phys. Chem.* **99**, 14573 (1995), bibtex: novikov\_cluster\_1995. [30](#)