

Lesson 7: Solvation models

Implicit vs. Explicit solvation

- Implicit solvation
 - Dielectric continuum
 - No water molecules *per se*
 - Wavefunction of solute affected by dielectric constant of solvent
 - At 20 °C: Water - $\epsilon = 78.4$; benzene: $\epsilon = 2.3 \dots$
- Explicit solvation
 - Solvent molecules included (i.e. with electronic & nuclear structure)
 - Used mainly in MM approaches
 - Microsolvation: only few solvent molecules placed around solute
 - Charge transfer with solvent can occur

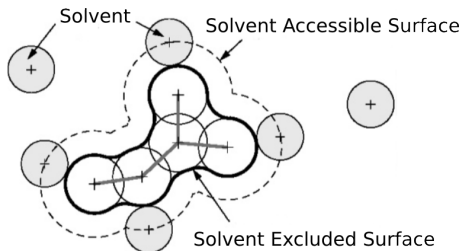
Implicit Models

Basic assumptions

- Solute characterized by QM wavefunction
- Born-Oppenheimer approximation
- Only interactions of electrostatic origin
- Isotropic solvent at equilibrium
- Static model

Cavity

- Solute is placed in a void of surrounding solvent called “cavity”
- Size of the cavity:
 - Computed using vdW radii of atoms (from UFF, for example)
 - Taken from the electronic isodensity level (typically 0.001 a.u.)
- The walls of cavity determine the interaction interface (Solvent Excluded Surface, **SES**)
- Size of the solvent molecule determines the Solvent Accessible Surface (**SAS**)



- Geomview software (in the modules) .
- Works only with Gaussian 03
- SCRF=(read) in the route section of the job
- “geomview” in the SCRF specification
- Visualize the “tesserae.off” file

Electrostatic Interactions

- Self-consistent solution of solute-solvent mutual polarizations
- Solute induces polarization at the interface of cavity
- This polarization acts back on the solute changing its wavefunction
- Various solvation models use different schemes for evaluation of solvation effects
- Problems arise when electrostatics do not dominate solvent-solute interactions

Polarizable Continuum Model (PCM)

- Treats the solvent as polarizable dielectric continuum
- Implemented in Gaussian, GAMESS

Solvation Model “Density” (SMD)

- Full solute density is used instead of partial charges
- Implicitly treats dispersion
- Lower unsigned errors against experimental data than other models

COnductor-like Screening MOdel (COSMO)

- Solute in virtual conductor environment
- Charge q on molecular surface is lower by a factor $f(\epsilon)$:

$$q = f(\epsilon)q^* \quad (1)$$

- where $f(\epsilon) = (\epsilon - 1)/(\epsilon + x)$; x being usually set to 0.5 or 0
- Implemented in Turbomole, ADF

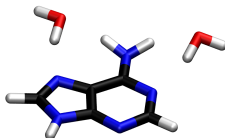
- Anisotropic liquids
- Nonequilibrium solution (Vertical excitations, TDDFT)
- Concentrated solutions

Explicit Models

Two models:

- Microsolvation

- Few solvent molecules (1 to 3) put at chemically reasonable place
- Water close to exchangeable protons (OH, NH₂...)



- Macrosolvation

- First (sometimes second) solvent layer around the whole molecule
- Usually snapshots from MD

- +++ Modelling of real interactions with solvent (this can be crucial for exchangeable protons in protic solvents)
- - Microsolvation lacks sampling
- - Computationally more demanding
- - For macrosolvation only single point calculations - the geometry is as good as forcefield

Practical task

- **cd /scratch/USERNAME***
- Model the $\text{Cl}^- + \text{CH}_3\text{Br} \rightarrow \text{CH}_3\text{Cl} + \text{Br}^-$
- Find the energy barrier for the reaction
- Select any solvent from Gaussian library (be not concerned about solubility of species or chemical relevance)
- Assume S_{n1} and S_{n2} reaction pathways
- Use “SCRF=(solvent=XY)” in the route section of the calculation

* If you are using INFINITY, you can ignore this step

- Use B3LYP 6-31+g(d,p) method
- Usage of diffuse functions when dealing with anions is crucial!
- Use ultrafine integration grid
- Use Frequency calculations to be sure where on PES you are
- For the scan use the distance between C and Cl as RC
- Negative value of step defines two atoms approaching

- Extraction of values from gaussian runs:
 - *extract-gopt-ene* logfile
 - *extract-gopt-xyz* logfile
 - *extract-gdrv-ene* logfile
 - *extract-gdrv-xyz* logfile
 - *extract-xyz-str* xyzfile framenummer
 - *extract-xyz-numstr* xyzfile
- Values ready for plotting in your favorite software

- Prepare job using *define* module (see presentation 6 for help)
- Setup COSMO using *cosmoprep* module
- Set epsilon to 78.4 and rsolv to 1.93
- Leave all other values at their default
- Define radii of atoms using “r all o” for optimized values
- Optimize all geometries