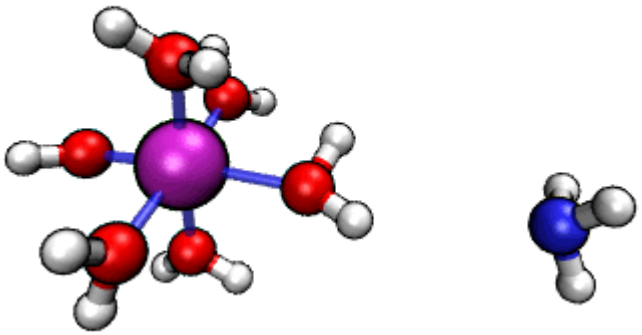
Ab initio Molecular Dynamics with ORCA

Martin Brehm
Martin-Luther-Universität Halle–Wittenberg
https://brehm-research.de/

Why another AIMD code?

- There exist several dozen AIMD codes *(CP2k, CPMD, ...)*
- We do not want to get into competition with those.



- **Different focus:**
  - No plane waves, just Gaussian atom centered basis functions
  - No periodic boundary conditions
  - Huge variety of electron structure methods *(HF, DFT, Semiempirics, MP2, CASSCF, DMRG, MRCI, CCSD)*
  - QM/MM directly in ORCA *(no external code/driver)*
  - Semiempirics MD *(xTB or PM3)* is interesting for large systems

The ORCA MD Module

- Standard ORCA input files are **keyword-based** *(not scripts)*:

```
! MP2 RHF def2-TZVP TightSCF

%mp2
  MaxCore 100
end

%paras
  rCO = 1.20
  ACOH = 120
  rCH = 1.08
end

* int 0 1
  C 0 0 0 0.00 0.0 0.00
  O 1 0 0 {rCO} 0.0 0.00
  H 1 2 0 {rCH} {ACOH} 0.00
  H 1 2 3 {rCH} {ACOH} 180.00
*
```

The ORCA MD Module

- Standard ORCA input files are **keyword-based** *(not scripts)*:
- Specifying „**MD**“ in the simple input switches on the MD module.
- All input for the MD module is found within „**%md ... end**“

```
! MD BLYP D3 def2-SVP

%md
  initvel 300_K
  timestep 0.5_fs
  thermostat berendsen 300_K timecon 10.0_fs
  dump position stride 1 filename "trajectory.xyz"
  run 2000
end

* xyz 0 1
  O    -2.03740    -1.21799    -0.08342
  H    -1.06493    -1.04408    -0.02285
  H    -2.37327    -1.07034     0.83692
  O    -1.65042     1.84243     0.07893
  H    -0.72656     1.49786    -0.01029
  H    -2.07086     1.65422    -0.79801
*
```

The ORCA MD Module

- Standard ORCA input files are **keyword-based** *(not scripts)*:
- Specifying „**MD**“ in the simple input switches on the MD module.
- All input for the MD module is found within „**%md ... end**“

```
! MD BLYP D3 def2-SVP

%md
  initvel 300_K
  timestep 0.5_fs
  thermostat berendsen 300_K timecon 10.0_fs
  dump position stride 1 filename "trajectory.xyz"
  run 2000
end

* xyz 0 1
  O    -2.03740    -1.21799    -0.08342
  H    -1.06493    -1.04408    -0.02285
  H    -2.37327    -1.07034     0.83692
  O    -1.65042     1.84243     0.07893
  H    -0.72656     1.49786    -0.01029
  H    -2.07086     1.65422    -0.79801
*
```

This part is a script – it runs from top to bottom!

The ORCA MD Module

- While the overall ORCA input file is **keyword-based**, the MD input (inside „**`%md ... end`**“) is a **script** which is executed during the ORCA run.

- The language of the MD input is **SANscript** *(which I am currently developing)*

`https://brehm-research.de/sanscript`

The ORCA MD Module

SANscript example program:

```
VAR tape : Character[]
    pos, p : Integer
    a : String

a := "+++++++++[->+++++++++<]>+.<++++[->++++<]>+++.<+++[->+++<]>+++++.++.---------.++++++
      ++.<+++++++[->-------<]>---------.<+++++[->-----<]>-.<++++[->++++<]>+++.-----.+++.+
      ++.---.++++.++++.-------.++++.-.--.++.++++.<++++++[->------<]>--------.---.<"

pos := 0
p := 0
tape.resize 1

WHILE p < a.length DO
  IF       a[p] = ']' THEN IF tape[pos] <> 0 THEN WHILE a[p] <> '[' DO p-- ENDDO ENDIF
    ELSEIF a[p] = '+' THEN tape[pos]++
    ELSEIF a[p] = '-' THEN tape[pos]--
    ELSEIF a[p] = '.' THEN Print tape[pos]
    ELSEIF a[p] = '<' THEN pos--
    ELSEIF a[p] = '>' THEN pos++ IF pos >= tape.size THEN tape.add 0 ENDIF
  ENDIF
  p++
ENDDO
```

The ORCA MD Module

SANscript example program:

```
VAR tape : Character[]
    pos, p : Integer
    a : String

a := "+++++++++[->+++++++++<]>+.<++++[->++++<]>+++.<+++[->+++<]>+++++.++.---------.++++++
      ++.<+++++++[->-------<]>---------.<+++++[->-----<]>-.<++++[->++++<]>+++.-----.+++.+
      ++.---.++++.++++.-------.++++.-.--.++.++++.<++++++[->------<]>--------.---.<"

pos := 0
p := 0
tape.resize 1

WHILE p < a.length DO
  IF       a[p] = ']' THEN IF tape[pos] <> 0 THEN WHILE a[p] <> '[' DO p-- ENDDO ENDIF
    ELSEIF a[p] = '+' THEN tape[pos]++
    ELSEIF a[p] = '-' THEN tape[pos]--
    ELSEIF a[p] = '.' THEN Print tape[pos]
    ELSEIF a[p] = '<' THEN pos--
    ELSEIF a[p] = '>' THEN pos++ IF pos >= tape.size THEN tape.add 0 ENDIF
  ENDIF
  p++
ENDDO
```

It emulates a Turing machine!

→ Proof that SANscript is a Turing-complete language! ☺

The ORCA MD Module

How to run the example? SANscript interpreter is not yet available...

The ORCA MD Module

How to run the example? SANscript interpreter is not yet available...

Just run it in ORCA ☺

```
! MD
%md
  $verbose execute 1$  # reduce printlevel
  VAR tape : Character[]
      pos, p : Integer
      a : String

  a := "+++++++++[->+++++++++<]>+.<++++[->++++<]>+++.<+++[->+++<]>+++++.++.---------.++++++
  ++.<+++++++[->-------<]>---------.<+++++[->-----<]>-.<++++[->++++<]>+++.-----.+++.+++.---
  .++++.++++.-------.++++.-.--.++.++++.<++++++[->------<]>--------.---.<"

  pos := 0
  p := 0
  tape.resize 1

  WHILE p < a.length DO
    IF       a[p] = ']' THEN IF tape[pos] <> 0 THEN WHILE a[p] <> '[' DO p-- ENDDO ENDIF
      ELSEIF a[p] = '+' THEN tape[pos]++
      ELSEIF a[p] = '-' THEN tape[pos]--
      ELSEIF a[p] = '.' THEN Print tape[pos]
      ELSEIF a[p] = '<' THEN pos--
      ELSEIF a[p] = '>' THEN pos++ IF pos >= tape.size THEN tape.add 0 ENDIF
    ENDIF
    p++
  ENDDO
end

* xyz 0 1
  Ar 0.0 0.0 0.0
*
```

The ORCA MD Module

There is now a Turing-complete scripting language inside ORCA

→ In theory, you can solve ***any*** computational problem with ORCA ☺

The SANscript language is still in development, but standard MD runs already work reliably.

Later, you will be able to have loops / branches / etc. in your MD input, allowing for complicated protocols.

```
! MD BLYP D3 def2-SVP

%md
  initvel 300_K
  timestep 0.5_fs
  thermostat berendsen 300_K timecon 10.0_fs
  dump position stride 1 filename "trajectory.xyz"
  run 2000
end

* xyz 0 1
  ...
*
```

Implemented Features *(ORCA 4.2.1)*

- AIMD with many different electron structure methods
  (HF, DFT, MP2, Semiempirics, xTB, CASSCF, ...)
- Trajectory output in **XYZ, PDB, DCD format**
- **Restart ability** to continue MD runs
- Simple **thermostats** *(Berendsen)*
- Keeping track of **energy drift** / conserved quantity
- Optional **repulsive walls** *(rectangular, elliptical, etc.)*
- **Constraints** *(distances / angles / dihedrals / centers of mass)*
- ORCA-internal **QM/MM** simulations
- **Region features** *(different thermostats for different parts of the system, etc.)*
- Cartesian **L-BFGS optimizer** for very large systems
  (10000s of atoms)

Planned Features

- **More sophisticated thermostats**
  (Nose-Hoover-Chains? CSVR?)
- **Restraints** *(currently only constraints implemented)*
- **Metadynamics** *(free energy sampling)*
  → e.g. accurate prediction of pK_A values
- **Calculation of properties** along trajectories
- **Vibrational spectra from AIMD** (IR, Raman, VCD, ROA)
  → many anharmonicities and full solvent influence!

Manual

There is a chapter on the MD module in the ORCA manual (*sec. 9.34, „Detailed Input –>* Ab initio Molecular Dynamics Simulations*“).*

There, you find a list of commands with description of all the options.

The manual of the MD module can also be found on my homepage:

`https://brehm-research.de/orcamd`

A library of input examples will be created in the future.

If you have questions, write a post in the **ORCA forum**!
(others might have same question; will be interested in answer)