comatsci-barrier
computational **mat**erials **sci**ence utility package
reaction barrier search utilities

Jan M. Knaup, Bremen Center for Compuational Materials Science
Jan.Knaup@bccms.uni-bremen.de

November 6, 2008

Contents

1 Introduction

1.1 Requirements

comatsci-barrier is an extension to the comatsci package. Therefore, comatsci must be installed on a system as a basic prerequisite. comatsci itself requires the following software/libraries to be present on a system:

- Python version $\geq$ 2.4 (not comatible to python 3)
  - POSIX compatible build environment and header files to compile python c-extensions
- numpy
- pyqt version $\geq$ 3
- ElementTree (included in Python version $\geq$ 2.5)

Note that windows is not an officially supported platform for comatsci.

Additionally, comatsci-barrier requires external computational chemistry packages to perform the energies and gradients calculations underlying it's operations. One or more of the following packages must be installed for comatsci-barrier to function:

- DFTB$^+$
- Gaussian (tested with ver. 03, other versions may or may not work)
- SIESTA

comatsci-barrier can easily be extended to support other calculation backend programs, by anyone with basic skills in object oriented programming and python. If you are interested in adapting comatsci-barrier, feel free to contact the author for support.

comatsci-barrier provides experimental functionality to distribute individual calculation jobs across several computers using MPI. To use this, a working MPI installation must be present, as well as the pypar package. (Note that at this time only limited support can be given for this facility, due to lack of development and testing resources.)

1.2 Installation

comatsci-barrier should be installed via the supplied setup script. Execute `python setup.py install` as root, to install comatsci system-wide. Any other installation tree can be chosen with the `--prefix` option to install, e.g. `--prefix=~` to install into `$HOME/bin` and `$HOME/lib`. The setup script also offers options to generate installers for different operating systems, depending on the platform and installed python version, please refer to the integrated documentation available by calling `python setup.py --help`.

1.3 License

comatsci-barrier is provided without any warranty as open software under the terms of the Noncommercial Open Software License v3.0. Please refer to the attached LICENSE file or appendix A.

Users are asked to acknowlegde the use of comatsci-barrier or the individual utility in publications of results obtained using this software.

1.4 Further reading

Please refer to the comatsci documentation for a descritption of single-geometry related utilities and supported file formats.

2 Utilities

As of version 1.0 only NEB reaction path search is supported. Other barrier search methods are being implemented but have not reached a publishable state.

2.1 Pastafarian

The *Path Search Tool bAsed on Flexible Atomistic Reaction Image ANalysis* searches for a chemical transition path using the *Nudged Elastic Band* method with various extensions.

Command Line Options

The general command line for pypath is as follows:

`pastafarian [options] <start geometry> [<intermediate geometries>] <end geometry>`

or

`pastafarian [options] <path file>`

where `<path file>` is a reaction path or NEB checkpoint in .fmg format.

The possible command line options are:

`-h, --help` show this help message and exit

`-m N, --maxit=N` Maximum number of NEB iterations for this run, default=100

`-r RMS, --rmstol=RMS` stop iterating if path RMS force in a.u. falls below RMS, default=1e-05

`-f FMAX, --forcetol=FMAX` stop iterating if path maximum of atomic forces in a.u. falls below FMAX, default=0.0001

`-c CALC, --calculator=CALC` use CALC to calculate energies and forces, default=dftb

`-x LIST, --fixedatoms=LIST` Keep atoms in comma delimeted LIST fixed, default=None

`-s D_T, --stepwidth=D_T` Use D_T as stepwidth for velocity Verlet relaxation, default=0.1

`--charge=Q` Set the system total charge to Q. Given in units of electrons, negative charge means electron excess, default=0

`-v` Increase verbosity level, default=1

`-q, --quiet` Limit output to fatal errors and critical warnings, no status output at all

`--silence` Limit output to fatal errors only. Not even warnings.

`-d SCHEDULER, --scheduler=SCHEDULER` Choose scheduler to manage execution of single calculations. Choices are 's': serial, 'p': MPI parallel. default=s

`-t MAXTHREADS, --maxworkthreads=MAXTHREADS` Set maximum number of concurrent parallel threads in thread scheduling mode. 0:= no limit. default=0 (currently unused)

`--pathdebug` Write individual .xyz and .fmg path files for each NEB step and write some energy and forces data to pathDebug.nrg for data debugging purposes

Default values for all command line options are overridden by the config file, `pastafarian --help` shows the default values actually used.

Experimental

MPI scheduling requires an installed pypar 1.9.2 or higher package and execution of the pastafarian script in an mpi environment, i.e. via mpirun or similar. *Caveat: currently the detection of MPI master or slave status is performed by evaluation of the LAMRANK or MPIRANK environment variables. This may break, depending on your installation!*

Config File

The PaSTaFARIAn config file has the name **pypath.ini** and is in the form of a windows .ini type file. (Extensions in Windowx XP .reg files are not supported.) The config file contains one section labeled **[PYPATH]**, which contains all configuration variables for the path search itself and possibly several sections with configuration variables for the different calculators. The calculator configuration sections are described int he Calculators section of this document.

All settings from the **[PYPATH]** section of the config file are overridden by the respective command line options, if such are given.

The Configuration variables are:

`maxit=100` maximum number of NEB iterations

`rmstol=2e-4` rms force convergence criterion

`forcetol=2e-3` max force convergence criterion

`calculator=dftb` calculator to use, currently dftb, noodle (=dftb+), gaussian and siesta are supported

`stepwidth=0.25` velocity verlet stepwidth

`fixedatoms=[]` put a comma separated list of fixed atoms here

`charge=0` set the system total charge to Q. Given in units of electrons, negative charge means electron excess.

Additionally, the section **[NEB]** stores NEB algorithm-specific settings

`fmode=s` force calculation mode:

- **s** standard NEB forces
- **c** climbing image NEB forces

`relmode=v` relaxation mode:

- **v** projected velocity verlet
- **s** (constant displacement) steepest descent

`tangmode=s`

- **s** use standard NEB tangents
- **w** use weighted tangents after J. Chem. Phys. **113**, 9978 (2000). Can lead to NEB force discontinuities between iterations if used on paths far from the PES valley. *Attention:* Overshoot detection is disabled when using weighted tangents.
- **p** use spline tangents from a parametric cubic spline representation of the path. **Caution** highly experimental.

Output files

PaSTaFARIAn writes most of its output to stdout. Additionally, the following files are written:

energies.dat: This file contains the reaction path energy profile for each NEB cycle that was performed in a format suitable for gnuplot. Column one contains the image sequence number, col. 2 contains the total energy in H. NEB cycles are separated by blank lines.

path.xyz: This file contains the last reaction path geometries calculated as a multi-frame .xyz file. This is mainly intended to be able to conveniently visualize the path using standard molecular visualization software.

checkpoint: This directory contains the last reaction path calculated in the form of .gen files named path-X.gen, where X is in the range of $0 \ldots 1$, reflecting the image sequence. This can be used to restart an interupted calculation by calling `pypath.py [options] checkpoint/*`. Old files are not deleted, so be sure to check the contents of this directory before using wildcards!

checkpoint.fmg: The last reaction path geometries in a single, multi-geometry fmg file. Suitable for use as a restart file. This is intended to replace the **checkpoint** directory in the future.

checkpoint.xyz: The last reaction path in a single, multi-frame .xyz file. Intended for convenient visualization.

Additionally, calculators may create further files in additional directories, e.g. for data reuse purposes.

3 Calculators

comatsci-barrier is designed to support any atomic simulation package as a calculator, as long as the package provides total energies and atomic forces in Cartesian coordinates. Adding another calculator is done by subclassing an abstract calculator class, implementing a few methods which write geometry and control input files, run the calculation binary and read the results. Currently, the following calculators are implemented and reasonably stable:

3.1 DFTB$^+$/Noodle

The noodle calculator uses DFTB$^+$ to calculate energies and forces. It has been tested with Version 0.2p0 and uses the new hsd input parser. Most of the dftb parameters are set via a parameter file included into the main hsd input via an <<+ or <<! directive. Parameters, which are to be set via the **[NOODLE]** section of the pypath.ini file are:

binary=/usr/local/bin/noodle full path to the noodle binary. Must be accessible on all nodes in parallel calculations

workdir=TEMP Name prefix for the directories in which to run the dftb+ calculations. The special Value "TEMP" uses mktmpdir to create a temporary directory in the path specified by the $TMPDIR environment variable.

skdir=/home/knaup/SlKo/ifam/alsicnoh-mavo full path to the SK files. The SK file naming convention follows the dftb calculator. This may change with future versions of noodle!

chrdir=charges name of the directory in which noodle restart files are stored

rchr=t read charges from last NEB iteration pypath checks whether the charge file is present, so it's safe to say t here

paraminclude=params.ndl A .hsd or .xml noodle input file which is included at the beginning of the input.hsd. System Charge, SK Files, LMAX and some options are overridden by automatically generated entries, so setting them here is useless.

oldSKnames=true use old Slater-Koster file naming convention, rather than the new one. *Defaults to true for Version 1.0, this may change in the future!*

All the usual caveats for dftb calculations apply here.

The directory specified in chargesdir must exist, even if charges are not reused! When in doubt remove the charges directory, damaged or incomplete charges files will lead to wrong results and may or may not cause error messages.

Slater-Koster file names

In the old convention, the dftb calculator expects to find SK files named by concatenating the lowercase element symbols for each interaction in the SK directory. E.g. for s C,O,H system, the file names would be `cc`,`co`, `ch`, `oc`, `oo`, `oh`, `hc`, `ho`, `hh`. Both files for must always be present, even for fully symmetric interactions. In such cases, use symlinks.

The new naming convention for SK files is to join the capitalized Element names by a dash - and add the extension `.skf`. The files from the example above would thus be called `C-C.skf`, `C-O.skf` `C-H.skf`..., a file for the Si – Na interaction would be called `Si-Na.skf`.

Point charges

The Noodle calculator supports the point charges external electric field option of noodle. There are two possible ways to employ it:

1. The point charges specification can be added to the noodle parameter include file. This requires no further intervention or even awareness of the calculator. The approach can be cumbersome for the user, as it requires the manual generation of the point charges to the parameter file. It also counteracts the possibility to use a single parameter file for a whole series of calculations to ensure consistency.

2. The internal Geometry representation supports grouping atoms into layers and the selection of subgeometries based on atomic layer assignment. Layers are identified by a unique integer layer index. Currently the Layer object only contains a Name for the layer. Layer names need not be unique. If the geometry object passed to the noodle calculator contains a layer named “PCHR”, the calculator automatically adds an external field/pointcharges specification, adding the atoms from the “PCHR” layer as point charges. *If several “PCHR” layers exist, only on of them is used, the choice is not strictly determined!*

   DEPRECATED

   In this case, only the atoms from the default layer “0” are added to the noodle geometry input. This means that all atoms not contained in the default layer or the “PCHR” layer will be ignored.
   Currently, geometry layers and atomic charge specification are only supported via the .fmg file format, cf. comatsci documentation. The geostats.py tool of comatsci provides a more user friendly interface to edit these properties than a plain text editor does.
   This way to specify external charges is deprecated, because in most situations tens of thousands of point-charges are necessary, which have to be kept in memory for each image along the path. Additionally, the computational effort to separate the layers in geometry of such size is considerable and may lead to significant overhead.

Both methods can be mixed, as the automatic point charges specification is added to any possible point charges spec from the parameter file.

Be aware that this is by no means an automatic QM/MM scheme. The user is responsible for providing link atoms and specifying the necessary geometry constraints.

3.2 SIESTA

The siesta calculator uses SIESTA to do the energy and forces calculations. It has only been tested with v1.3 and v1.3-f1p, but older versions might work as well. Most of the SIESTA parameters are set in an .fdf file, which must be supplied by the user and is includes into the siesta input using an “%include” directive. The Parameters to be set in the **[SIESTA]** section of the pypath.ini file are:

binary=/usr/local/bin/siesta Full path to the SIESTA binary.

ppdir=/home/user/siesta-pseudo Directory containing the pseudo potential files to use for the SIESTA calculations.

workdir=TEMP Name prefix for the directories in which to run the SIESTA calculations. The special Value “TEMP” uses mktmpdir to create a temporary directroy in the path specified by the $TMPDIR environment variable.

dmdir=DMS Directory in which to store the .DM files from the calculation for the different images. *Warning: If the basis set has changed, the old .DM files cannot be reused. SIESTA will inevitably fail.*

rdms=t Reuse the .DM file from the previous calculation on the same image. This should usually bet set to “t” as it speeds up the calculations enormously. It is safe to to say “t” here, if no old .DM files are present.

paraminclude=params.fdf An .fdf file which should contain all calculation parameters not automatically set by pypath. Pypath automatically sets the following parameters, which *must not* be set in this file:

- Atomic Coordinates
- Cell Vectors
- Chemical Species
- NumCGSteps
- any MD parameters
- systemlabel
- %include directives

Besides the usual caveats for SIESTA calculations, the following tip might be useful: SIESTA has difficulties finding a proper density matrix if the atomic configuration is not close to an equilibrium state. This might lead to convergence failure in the vicinity an interpolated intermediate state. In Such cases it is useful, to copy a .DM file from a neighboring image to the image with convergence problems. A converged density map for a similar isoelectronic configuration is usually a much better starting point than SIESTA’s initial guess.

3.3 Gaussian (03)

The Gaussian calculator is only tested for Gaussian03. Please consider your Gaussian Manual for extensive documentation on Gaussian use and features.

Parameters for the Gaussian calculator are entered in the [Gaussian] section of pypath.ini. Parameters are:

binary=g03 path to the gaussian binary

workdir=TEMP Name prefix for the directories in which to run the SIESTA calculations. The special Value “TEMP” uses mktmpdir to create a temporary directroy in the path specified by the $TMPDIR environment variable.

chkdir=g03chk directroy in which gaussian checkpoint files will be stored

rchk=false read checkpoint file (**warning:** this is untested and may go horribly wrong)

hamiltonian=BPL/6-31G*/AUTO Hamiltonian part of the route section

routeopts= further options to incorporate into the route section. Do not specify SP, OPT or FREQ statements here!

spinmul=1 spin multiplicity

3.4 Müller-Brown potential

Called as `muellerbrown`. The two-dimensional test potential for reaction path analysis algorithms suggested by **?**]. The potential is dependent on the absolute atomic coordinate in Bohr. Intended for testing and demonstration purposes. Be aware that the potential operates on shorter distances than interatomic potentials, i.e. the gradients are very steep and smaller step sizes are necessary.

Takes no options and has therefore no config file section.

A Non-Profit Open Software License ("Non-Profit OSL") 3.0

Licensed under the Non-Profit Open Software License version 3.0

1) Grant of Copyright License. Licensor grants You a worldwide, royalty-free, non-exclusive, sublicensable license, for the duration of the copyright, to do the following:

a) to reproduce the Original Work in copies, either alone or as part of a collective work;

b) to translate, adapt, alter, transform, modify, or arrange the Original Work, thereby creating derivative works ("Derivative Works") based upon the Original Work;

c) to distribute or communicate copies of the Original Work and Derivative Works to the public, with the proviso that copies of Original Work or Derivative Works that You distribute or communicate shall be licensed under this Non-Profit Open Software License or as provided in section 17(d);

d) to perform the Original Work publicly; and

e) to display the Original Work publicly.

2) Grant of Patent License. Licensor grants You a worldwide, royalty-free, non-exclusive, sublicensable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, for the duration of the patents, to make, use, sell, offer for sale, have made, and import the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work.

4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior permission of the Licensor. Except as expressly stated herein, nothing in this License grants any license to Licensor's trademarks, copyrights, patents, trade secrets or any other intellectual property. No patent license is granted to make, use, sell, offer for sale, have made, or import embodiments of any patent claims other than the licensed claims defined in Section 2. No license is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under terms different from this License any Original Work that Licensor otherwise would have a right to license.

5) External Deployment. The term "External Deployment" means the use, distribution, or communication of the Original Work or Derivative Works in any way such that the Original Work or Derivative Works may be used by anyone other than You, whether those works are distributed or communicated to those persons or made available as an application intended for use over a network. As an express condition for the grants of license hereunder, You must treat any External Deployment by You of the Original Work or a Derivative Work as a distribution under section 1(c).

6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent, or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an "Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. The Original Work is provided under this License on an "AS IS" BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of non-infringement, merchantability or fitness for a particular purpose. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to the Original Work is granted by this License except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to anyone for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to the extent applicable law prohibits such limitation.

9) Acceptance and Termination. If, at any time, You expressly assented to this License, that assent indicates your clear and irrevocable acceptance of this License and all of its terms and conditions. If You distribute or communicate copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. This License conditions your rights to undertake the activities listed in Section 1, including your right to create Derivative Works based upon the Original Work, and doing so without honoring these terms and conditions is prohibited by copyright law and international treaty. Nothing in this License is intended to affect copyright exceptions and limitations (including "fair use" or "fair dealing"). This License shall terminate immediately and You may no longer exercise any of the rights granted to You by this License upon your failure to honor the conditions in Section 1(c).

10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, against Licensor or any licensee alleging that the Original Work infringes a patent. This termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software or hardware.

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of copyright or patent law in the appropriate jurisdiction. This section shall survive the termination of this License.

12) Attorneys' Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) Miscellaneous. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by You.

16) Modification of This License. This License is Copyright © 2005 Lawrence Rosen. Permission is granted to copy, distribute, or communicate this License without modification. Nothing in this License permits You to modify this License as applied to the Original Work or to Derivative Works. However, You may modify the text of this License and copy, distribute or communicate your modified version (the "Modified License") and apply it to other original works of authorship subject to the following conditions: (i) You may not indicate in any way that your Modified License is the "Open Software License" or "OSL" and you may not use those names in the name of your

Modified License; (ii) You must replace the notice specified in the first paragraph above with the notice "Licensed under <insert your license name here>" or with a notice of your own that is not confusingly similar to the notice in this License; and (iii) You may not claim that your original works are open source software unless your Modified License has been approved by Open Source Initiative (OSI) and You comply with its license review and certification process.

17) Non-Profit Amendment. The name of this amended version of the Open Software License ("OSL 3.0") is "Non-Profit Open Software License 3.0". The original OSL 3.0 license has been amended as follows:

(a) Licensor represents and declares that it is a not-for-profit organization that derives no revenue whatsoever from the distribution of the Original Work or Derivative Works thereof, or from support or services relating thereto.

(b) The first sentence of Section 7 ["Warranty of Provenance"] of OSL 3.0 has been stricken. For Original Works licensed under this Non-Profit OSL 3.0, LICENSOR OFFERS NO WARRANTIES WHATSOEVER.

(c) In the first sentence of Section 8 ["Limitation of Liability"] of this Non-Profit OSL 3.0, the list of damages for which LIABILITY IS LIMITED now includes "direct" damages.

(d) The proviso in Section 1(c) of this License now refers to this "Non-Profit Open Software License" rather than the "Open Software License". You may distribute or communicate the Original Work or Derivative Works thereof under this Non-Profit OSL 3.0 license only if You make the representation and declaration in paragraph (a) of this Section 17. Otherwise, You shall distribute or communicate the Original Work or Derivative Works thereof only under the OSL 3.0 license and You shall publish clear licensing notices so stating. Also by way of clarification, this License does not authorize You to distribute or communicate works under this Non-Profit OSL 3.0 if You received them under the original OSL 3.0 license.

(e) Original Works licensed under this license shall reference "Non-Profit OSL 3.0" in licensing notices to distinguish them from works licensed under the original OSL 3.0 license.