Finding Saddle Points on Potential Energy Surfaces:

Exploration with ART

Antoine Jay[1], Nicolas Salles[3], Miha Gunde[1], Matic Poberznik[3], Layla Martin-Samos[3], Nicolas Richard[4], Stefano De Gironcoli[3], Anne Hémeryck[1], Normand Mousseau[5]

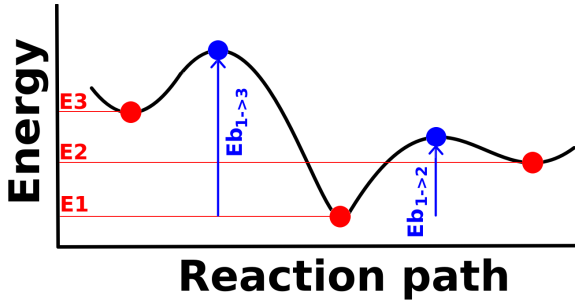
[1]LAAS-CNRS, *Université de Toulouse, CNRS, Toulouse, France*
[2]CIMI-DEOS, *ISAE Supaéro, Toulouse, France*
[3]CNR-IOM, *Democritos and Sissa, Trieste, Italy*
[4]CEA, DAM, DIF, *Arpajon, France*
[5]Université de Montréal, *Montréal, Canada*

Overview

Table of Contents

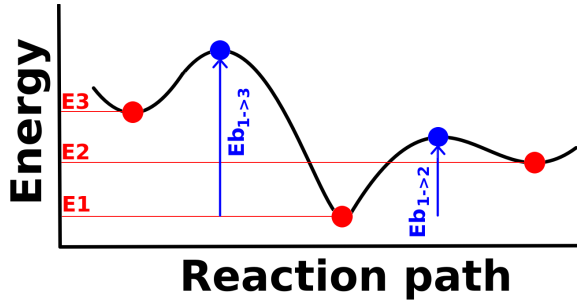
Goal

From an atomic structure, discover new structures and energy barriers



Goal

From an atomic structure, discover new structures and energy barriers



Minima → thermodynamics

$$Prob(i) = \frac{e^{\frac{-E_i}{k_B T}}}{Z}$$

$$Z = \sum_{i}^{N_{configurations}} e^{\frac{-E_i}{k_B T}}$$

Saddle points → kinetics

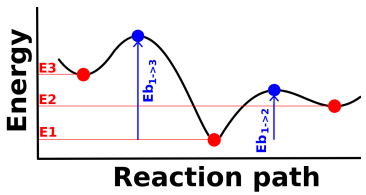
$$k_{1\to 2} = \omega_{vib} e^{\frac{-(E_b)}{k_B T}}$$

$$\omega_{vib} = \frac{\prod_i^{3N_{at}} \omega_i(State1)}{\prod_j^{3N_{at}-1} \omega_j(saddle)} \sim 10^{13} Hz$$

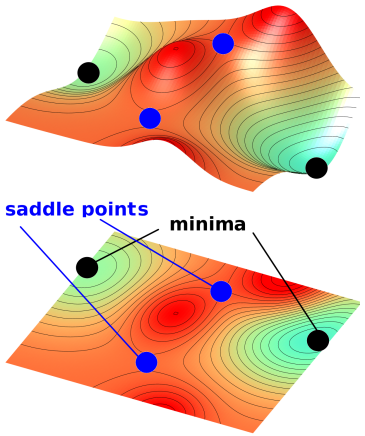
$$\Delta t = -\frac{ln\lambda}{\sum_n k_n}$$

Definitions: Minima and Saddle points

1D



2D

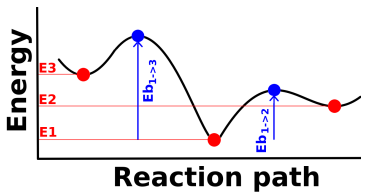


$3 \times N_{at} D$

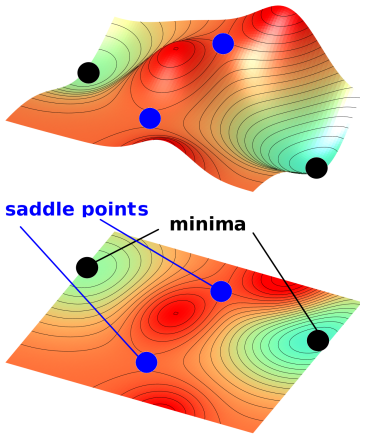
?????

Definitions: Minima and Saddle points

1D



2D



$3 \times N_{at}D$

?????

Minimum:
min in $3N_{at}$ D

Saddle point:
max in 1D,
min in $(3N_{at}\text{-}1)$D

Steps:

- Find the vector corresponding to this D
- Minimize in the orthogonal hyperplane.

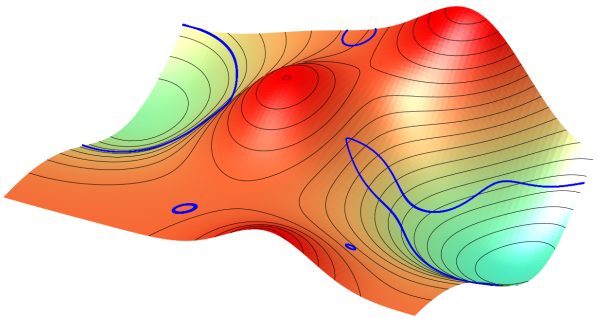
Convergence: $\forall i,\ F_i = \frac{dE}{dx_i} \sim 0$

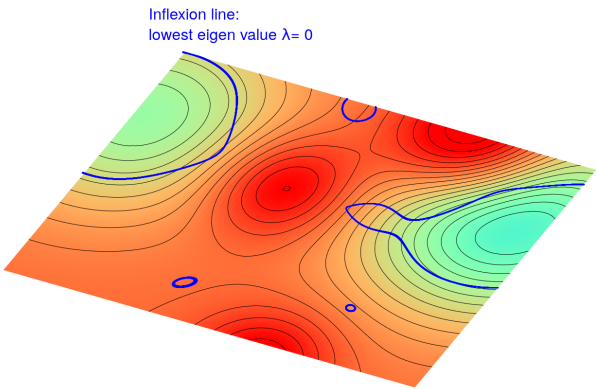
Definitions: Hessian Matrix and inflexion

$H_{ij} = \frac{\partial^2 E}{\partial x_i \partial x_j}$ eigenvalues λ_i and eigenvectors $\mathbf{V_i}$

Harmonic bassin: $\forall i, \lambda_i > 0$

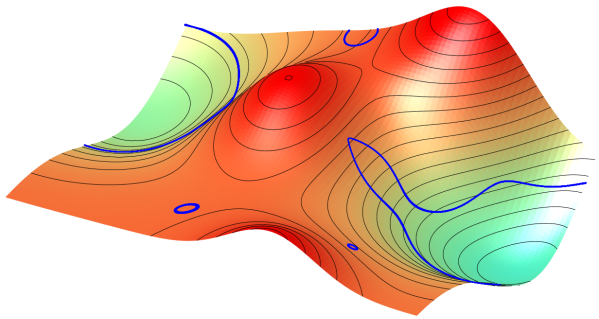
Minima: $\frac{\partial E}{\partial x} = 0$

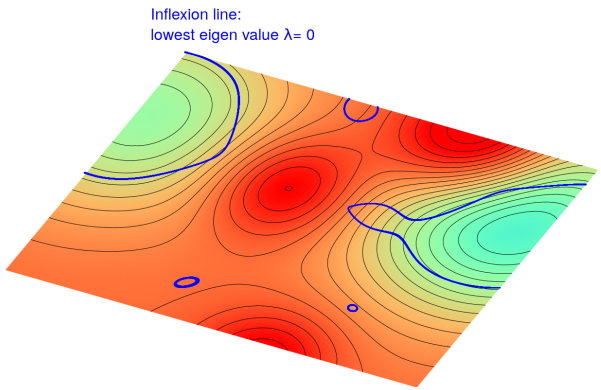




Above Inflexion: $\exists i, \lambda_i < 0$

Saddle points: $\frac{\partial E}{\partial x} = 0$



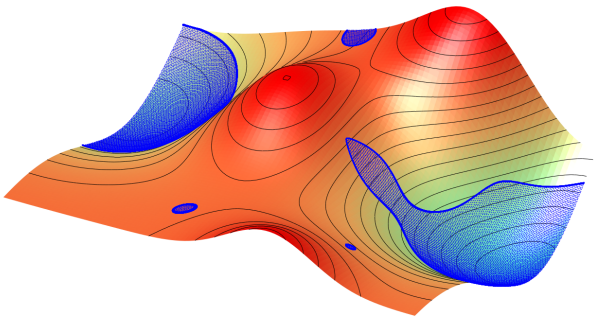


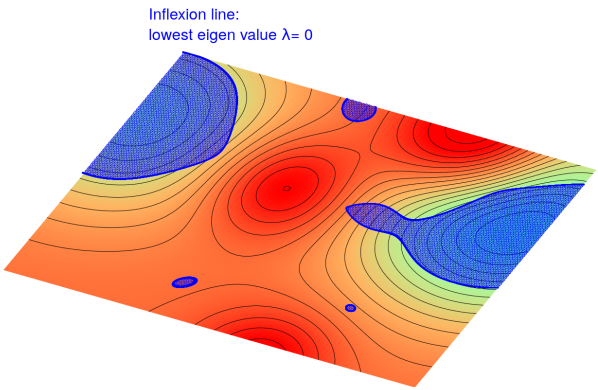
Definitions: Hessian Matrix and inflexion

$H_{ij} = \frac{\partial^2 E}{\partial x_i \partial x_j}$ eigenvalues λ_i and eigenvectors $\mathbf{V_i}$

Harmonic bassin: $\forall i, \lambda_i > 0$

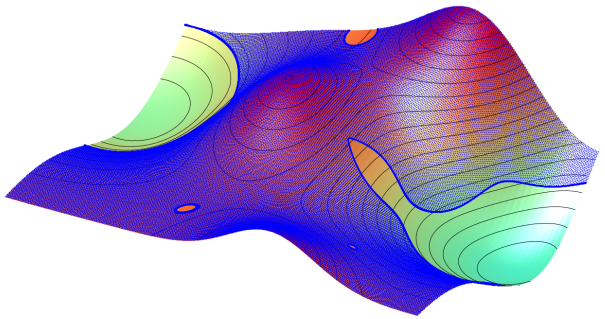
Minima: $\frac{\partial E}{\partial x} = 0$

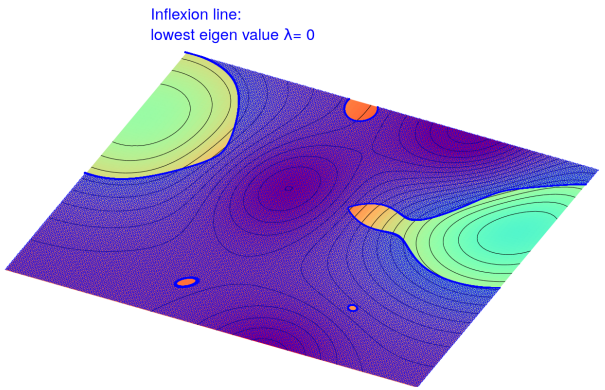




Above Inflexion: $\exists i, \lambda_i < 0$

Saddle points: $\frac{\partial E}{\partial x} = 0$



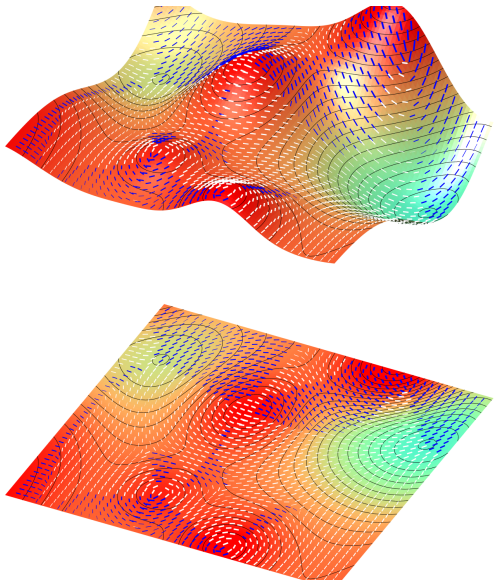


Only the lowest eigenvalue λ_{min} and $\mathbf{V_{min}}$ are needed

$\rightarrow$ LANCZOS

Definitions: Valeys, Forces, Eigen Vectors

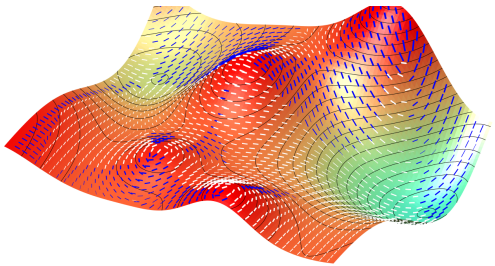
Eigen Vectors:
$\mathbf{V_{min}}$ from Hessian

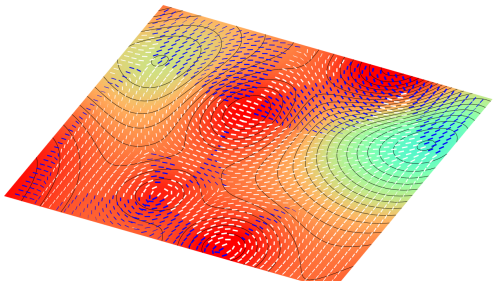


$\rightarrow \parallel$ MEP

Definitions: Valeys, Forces, Eigen Vectors

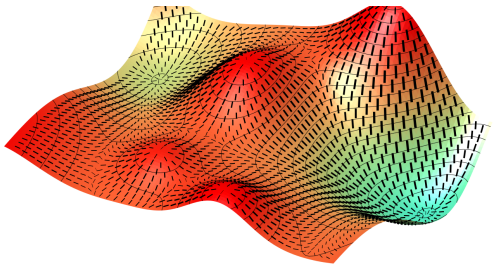
Eigen Vectors:
$\mathbf{V_{min}}$ from Hessian

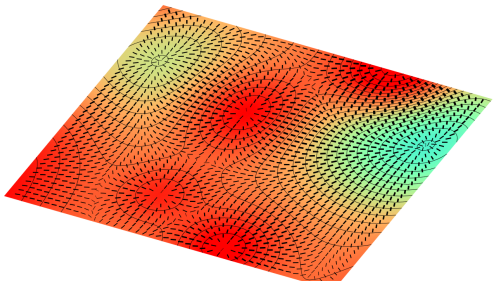




$\rightarrow \parallel$ MEP

Forces:
$-\frac{\partial E}{\partial x}$

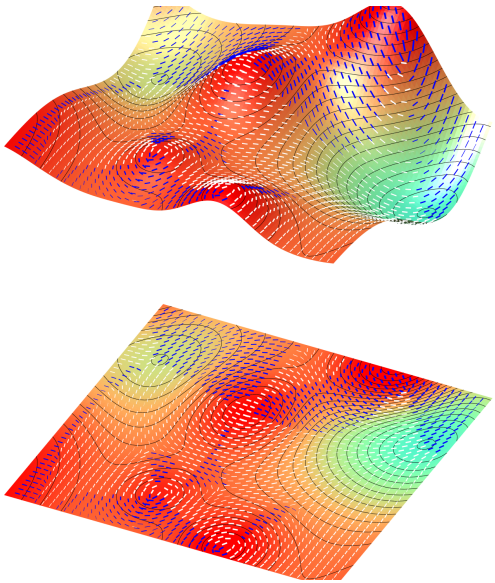




$\perp$ isolines

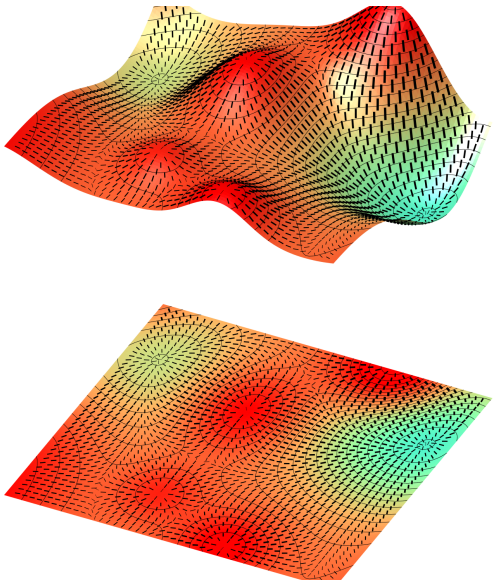
Definitions: Valeys, Forces, Eigen Vectors

Eigen Vectors:
$\mathbf{V_{min}}$ from Hessian



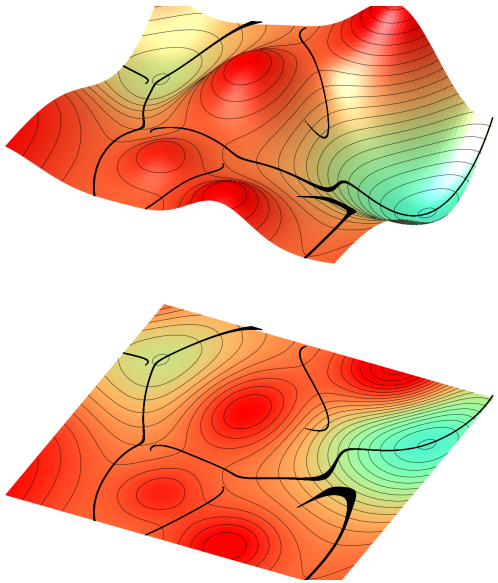
$\rightarrow \parallel$ MEP

Forces:
$-\frac{\partial E}{\partial x}$



$\perp$ isolines

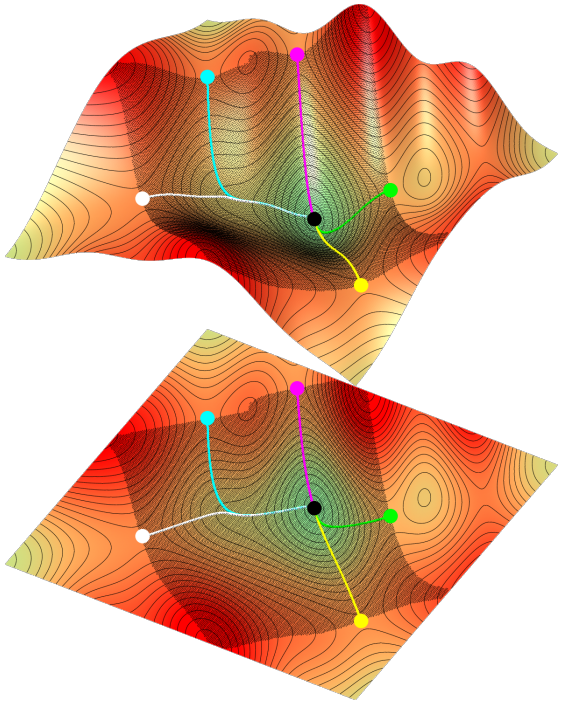
Valeys:
$\mathbf{V_{min}} \parallel \mathbf{F}$ or $\mathbf{F}_{\perp} = 0$



is MEP

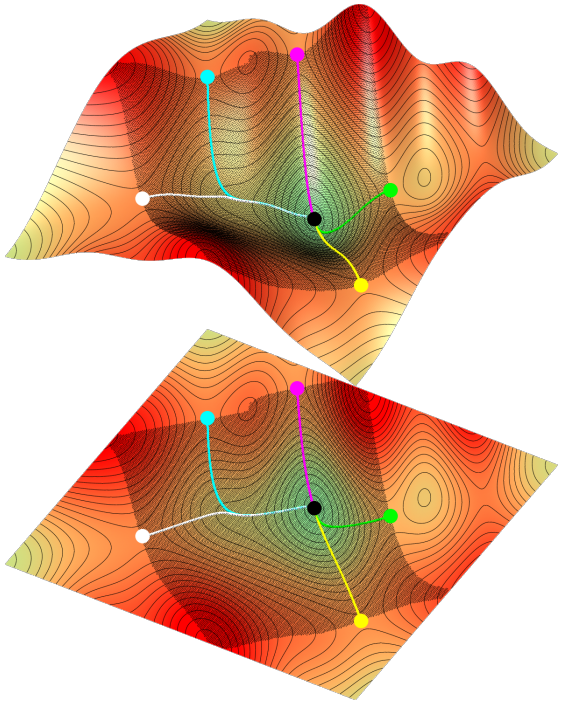
Definitions: Asymetric problem

Basin and IRC

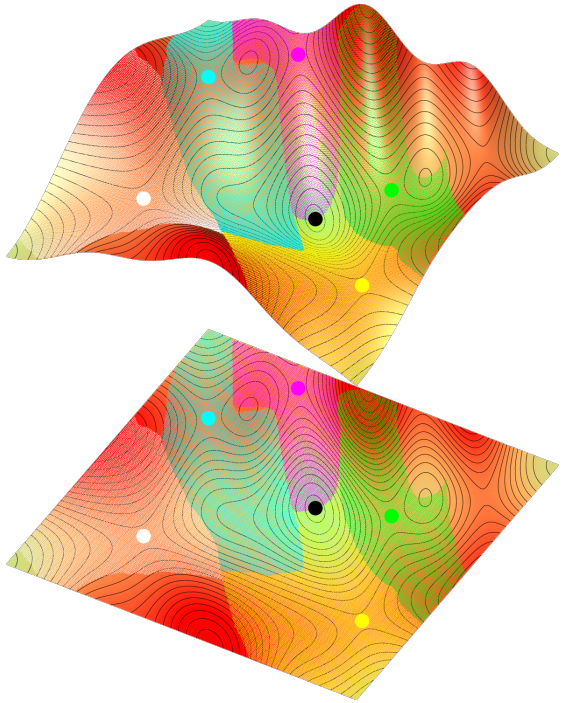


Definitions: Asymetric problem

Basin and IRC

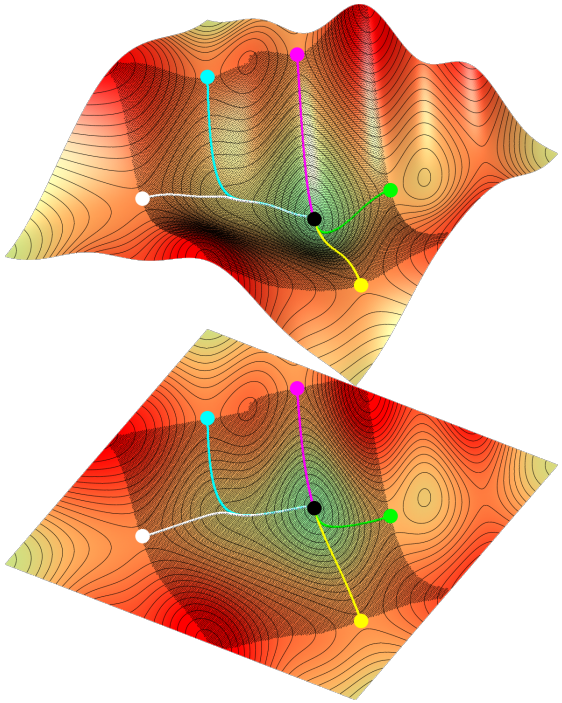


attracted area

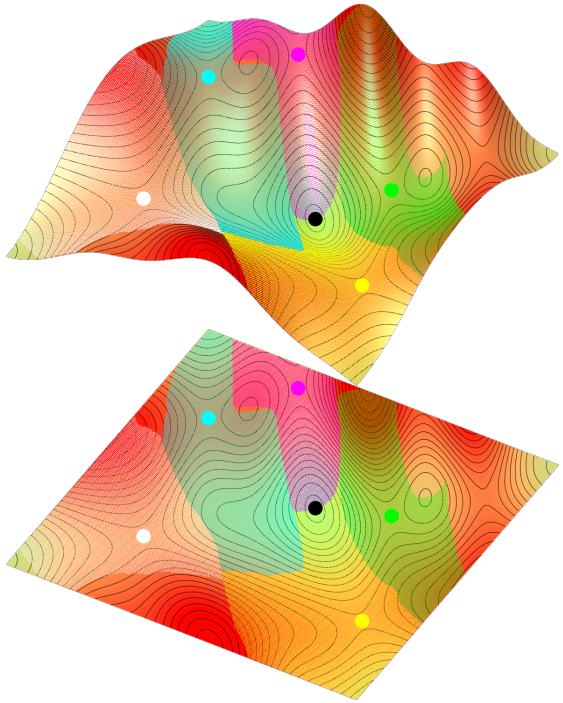


Definitions: Asymetric problem

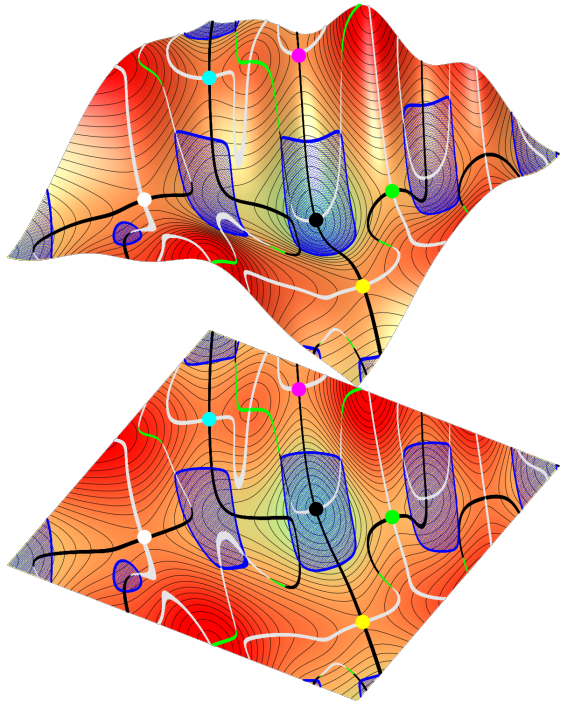
Basin and IRC



attracted area

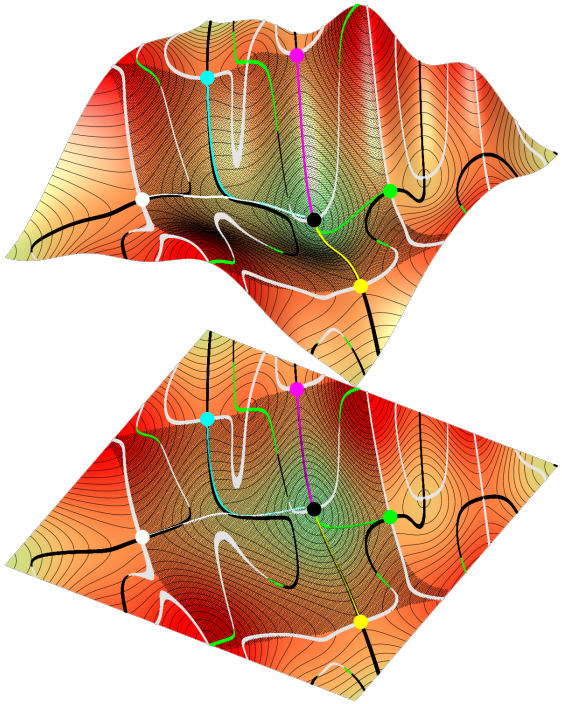


PES features

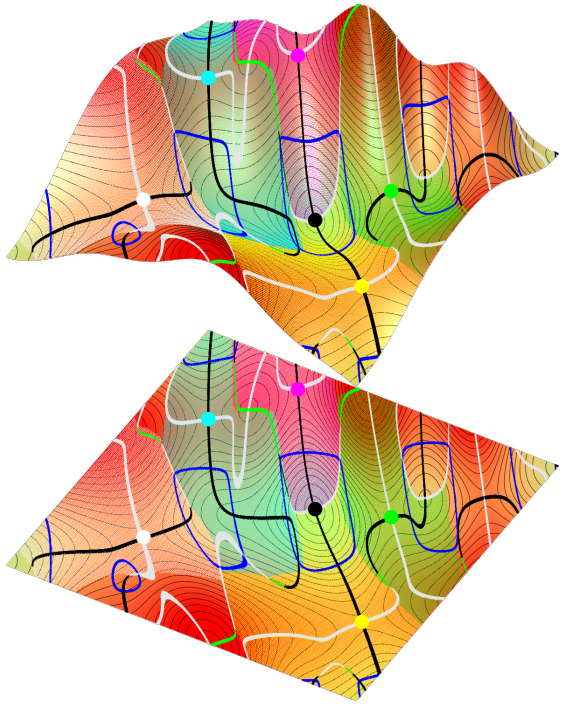


Definitions: Asymetric problem

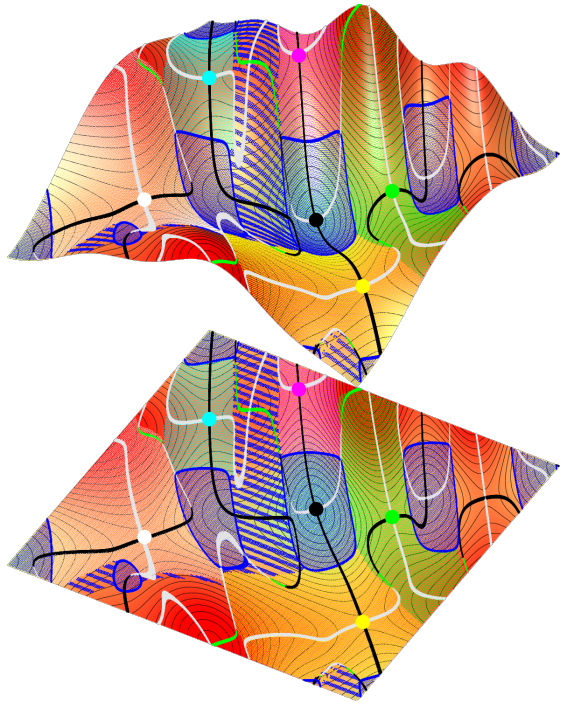
Basin and IRC



attracted area

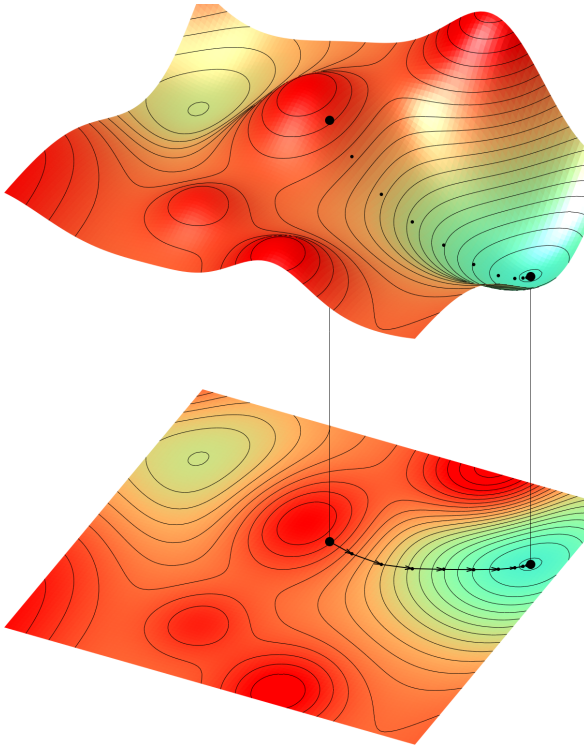


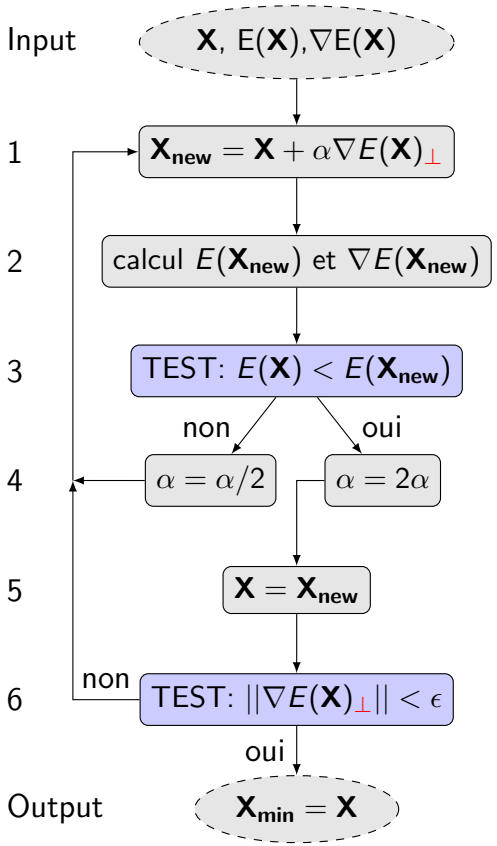
PES features



Definitions: Simple Vs Orthogonal relaxation

Simple





Orthogonal

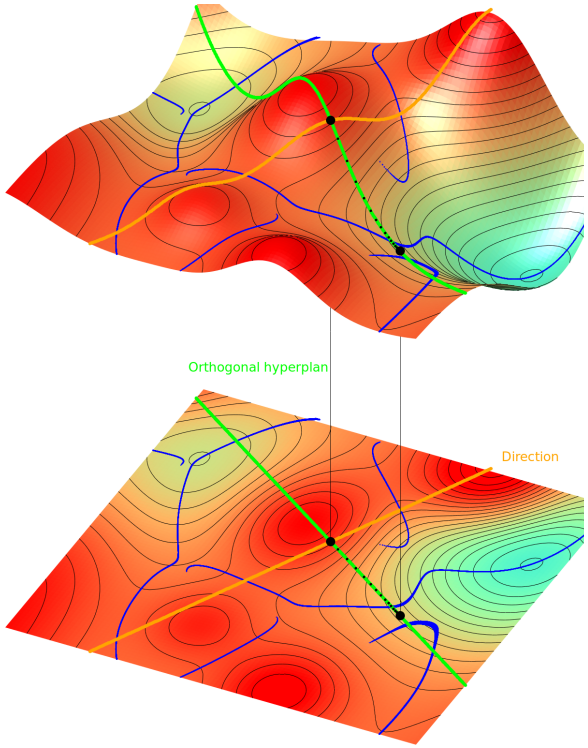
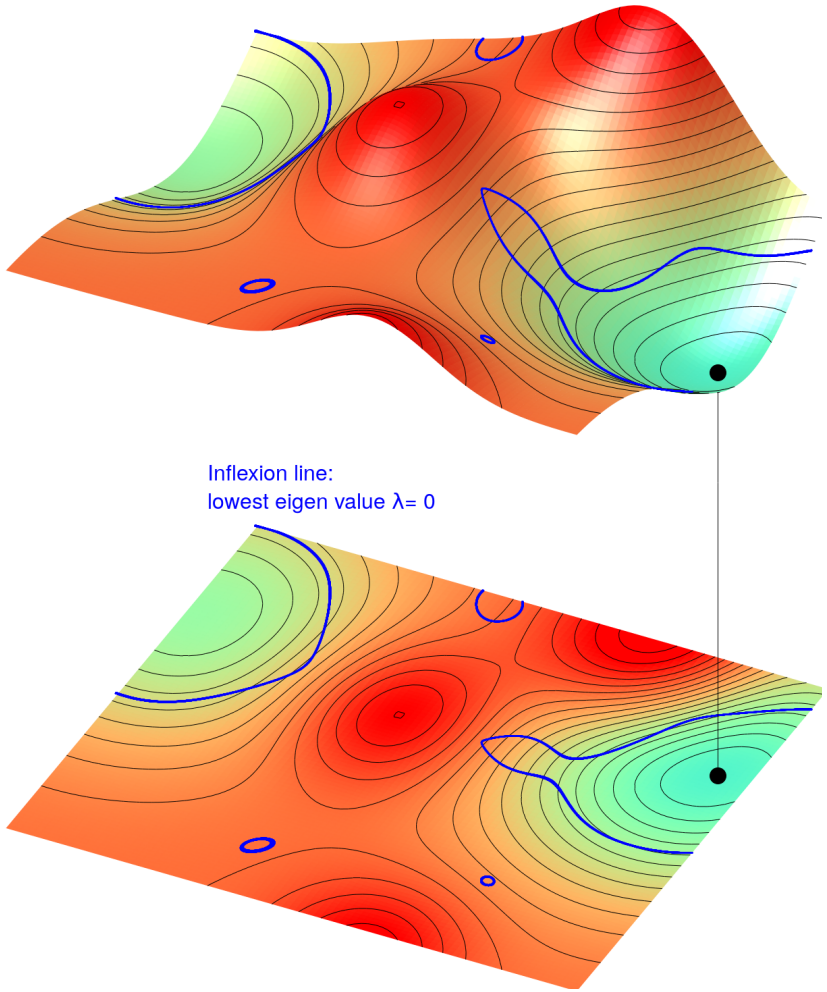


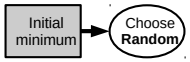
Table of Contents

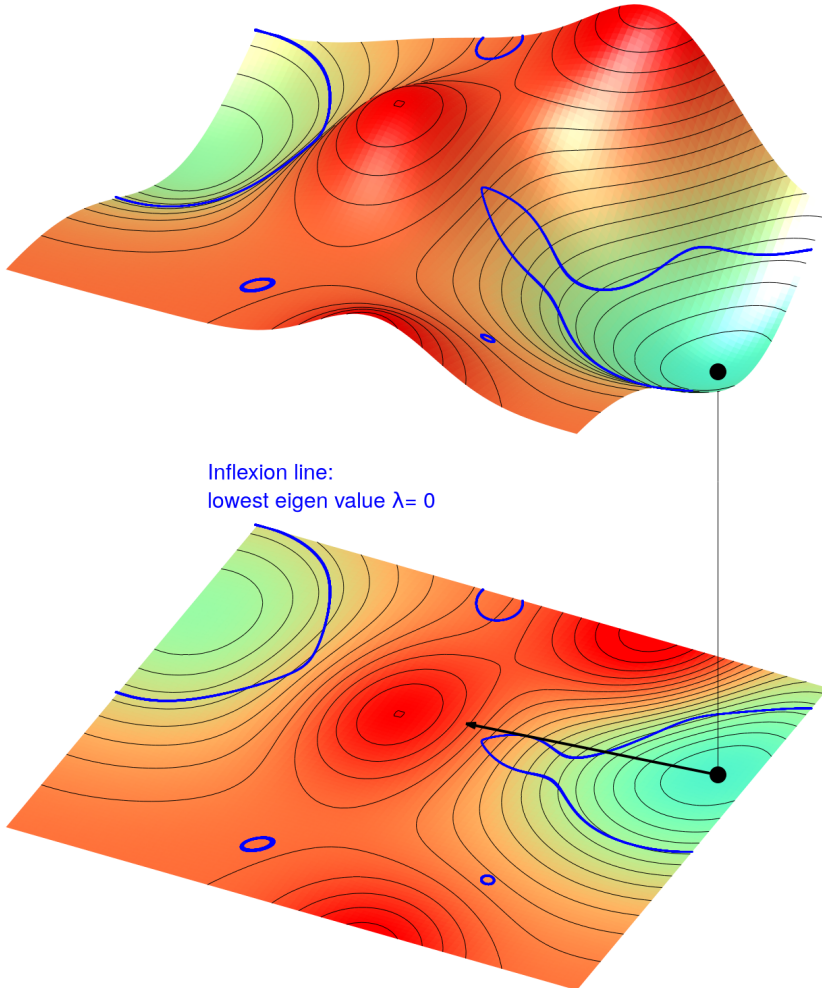
ARTn: Algorithm (1996)

Initial minimum

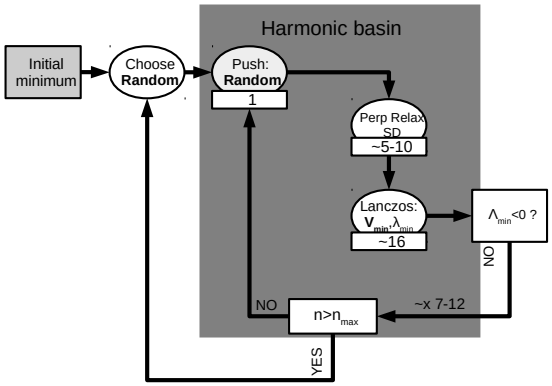


ARTn: Algorithm (1996)

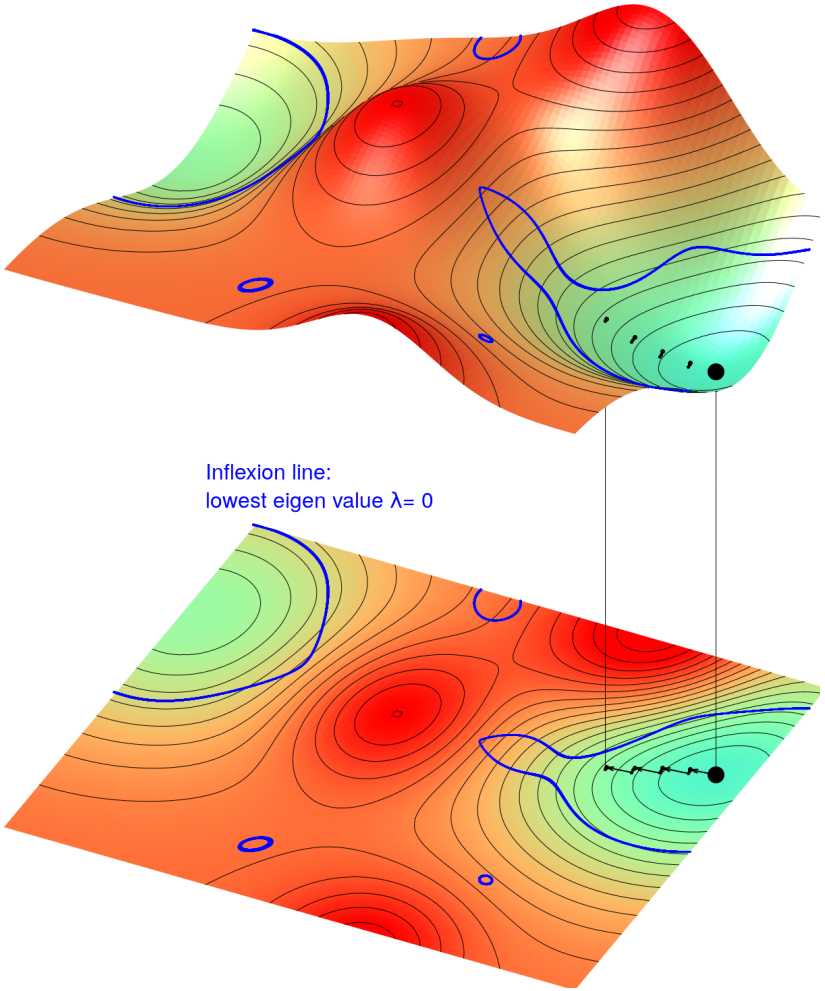




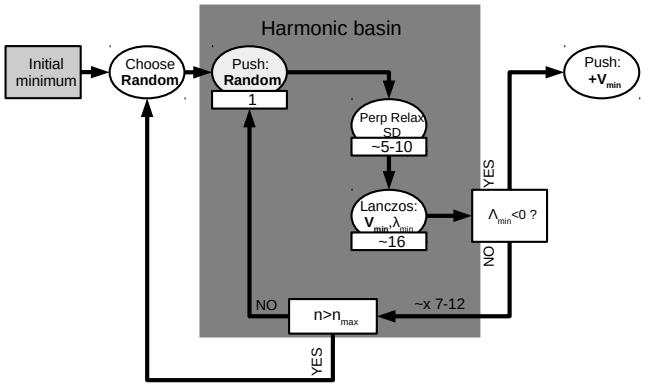
ARTn: Algorithm (1996)



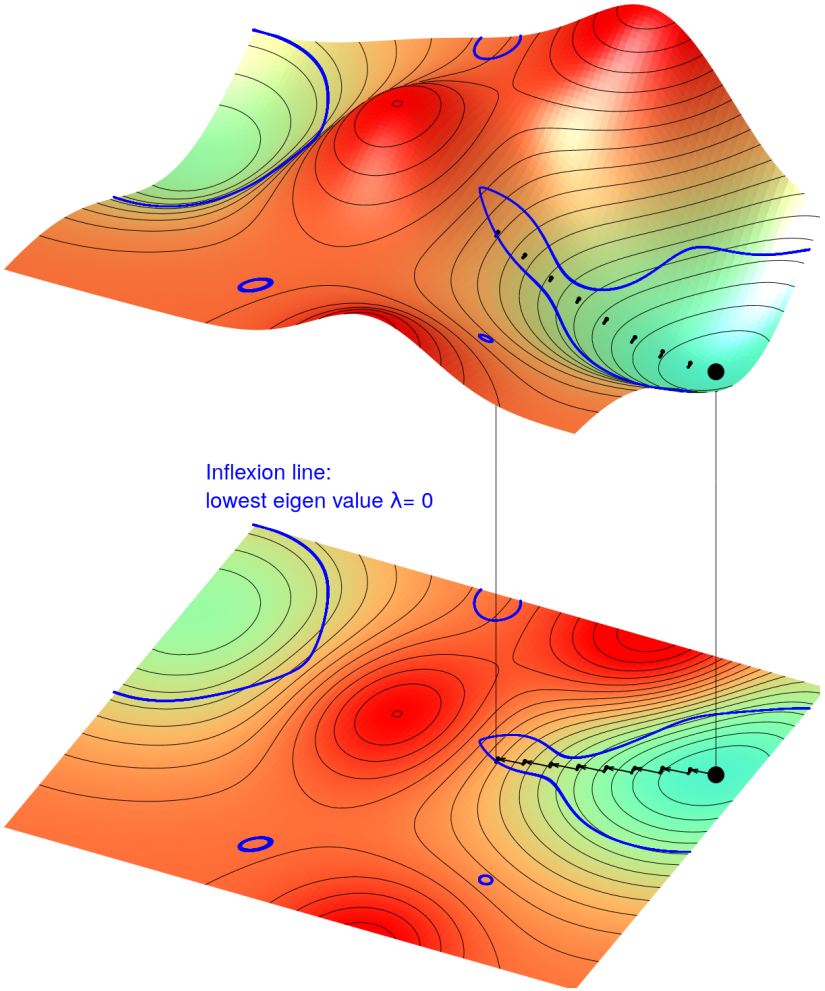
$$||Push|| = cst$$



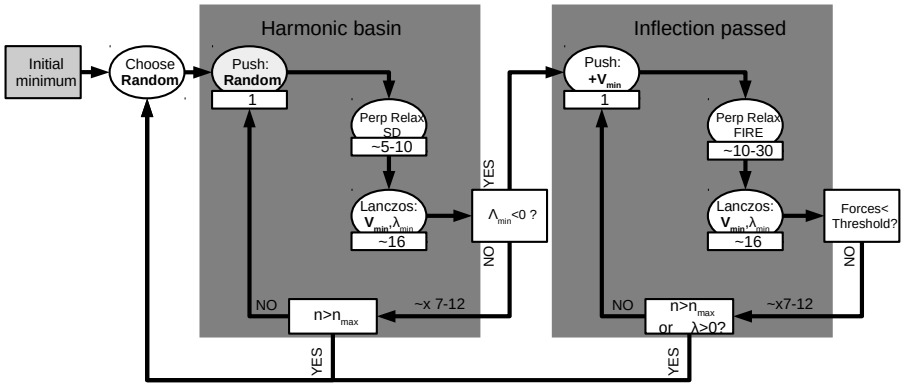
ARTn: Algorithm (1996)

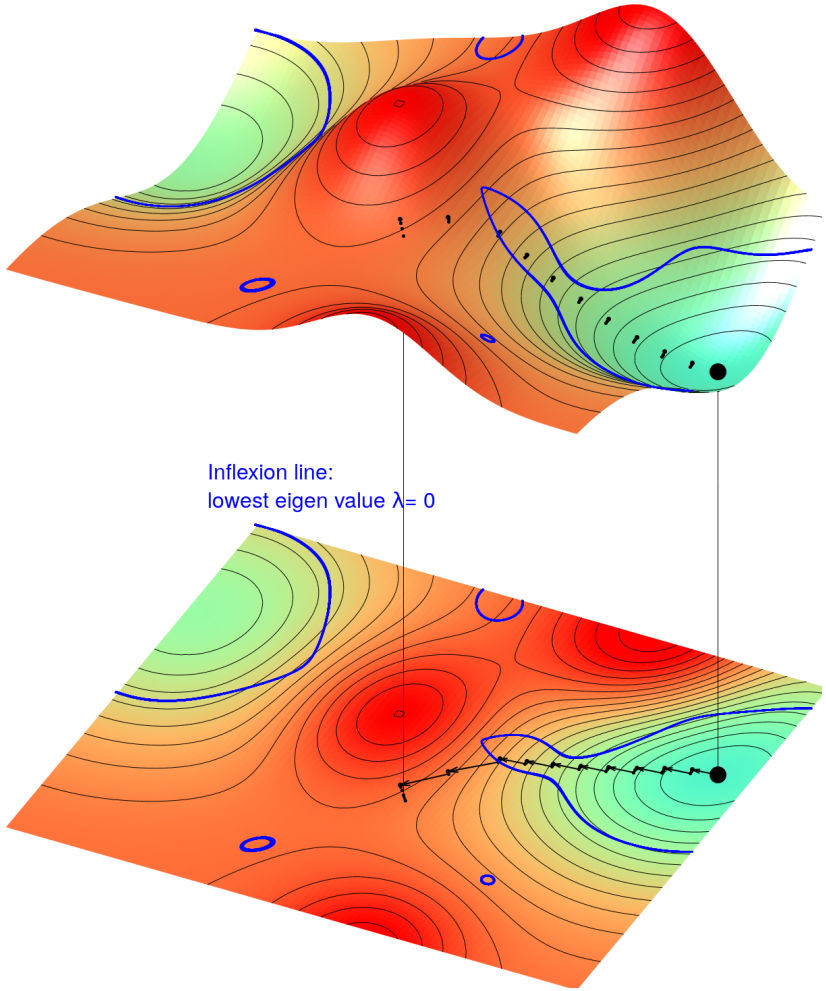


$$||Push|| = cst$$



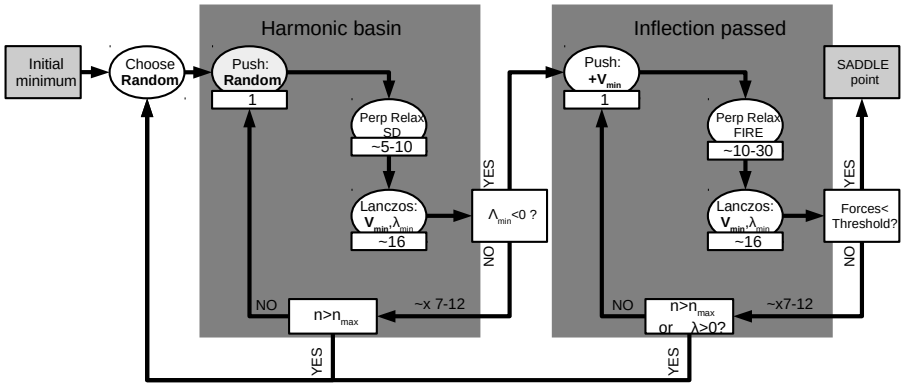
ARTn: Algorithm (1996)

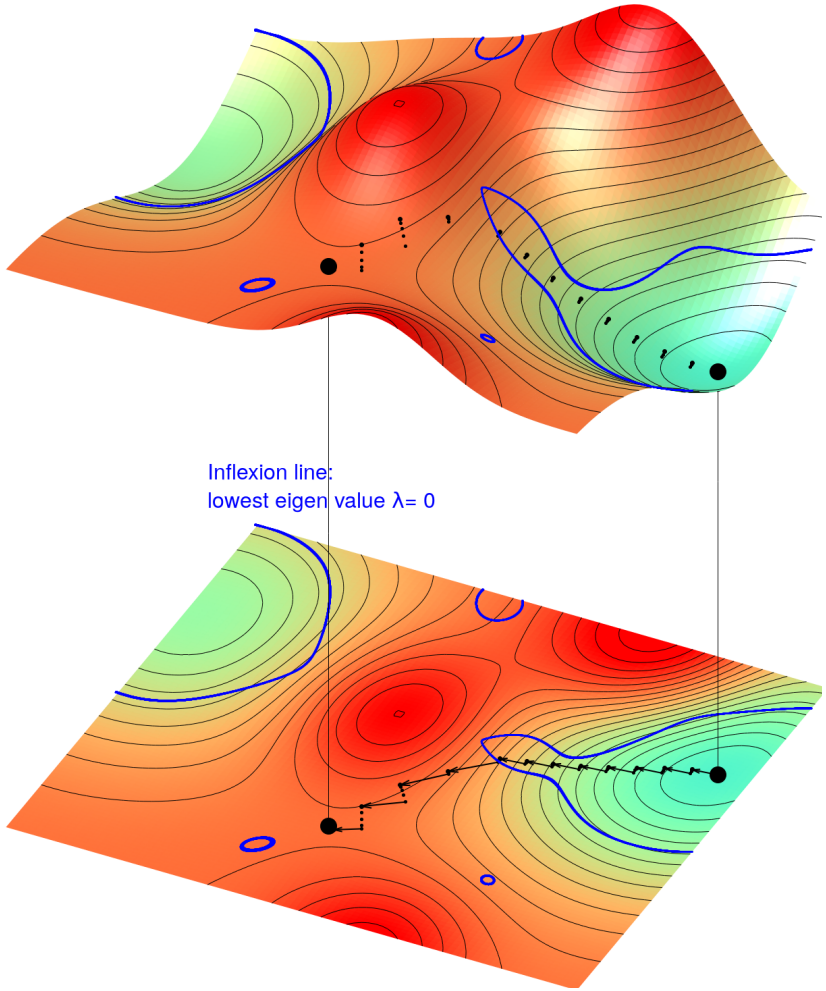




$$||Push|| = min\left(size_{max}, \frac{|f_{par}|}{max(|\lambda_0|,0.5)}\right)$$

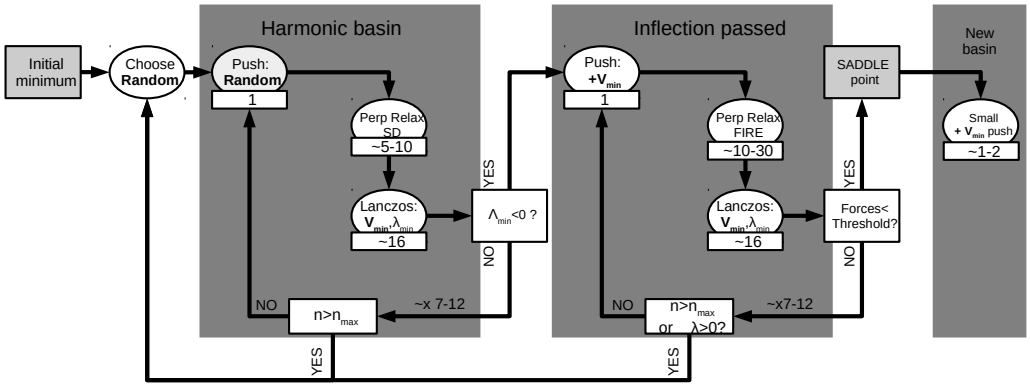
ARTn: Algorithm (1996)

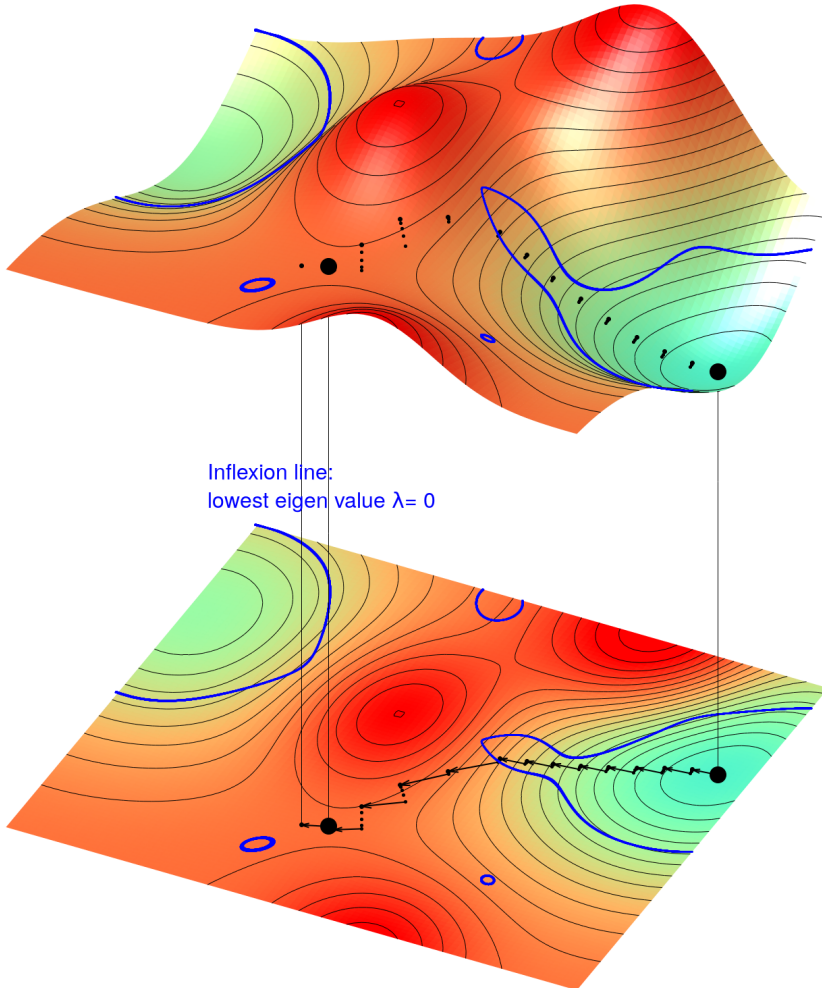




$$||Push|| = min\left(size_{max}, \frac{|f_{par}|}{max(|\lambda_0|,0.5)}\right)$$

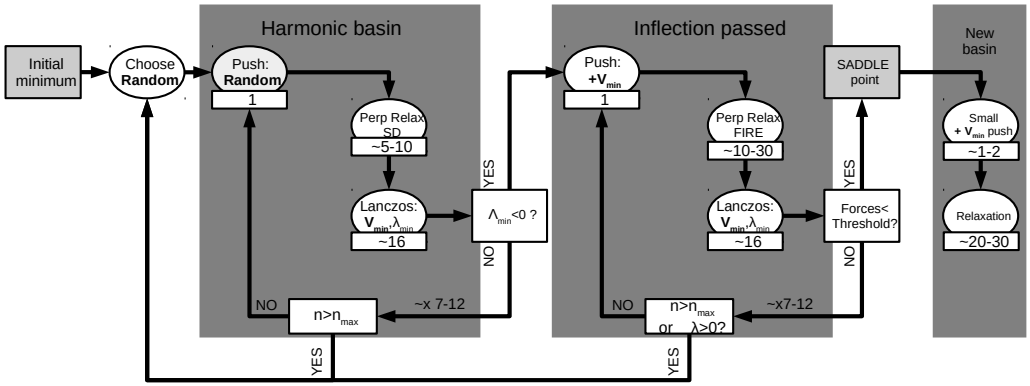
ARTn: Algorithm (1996)

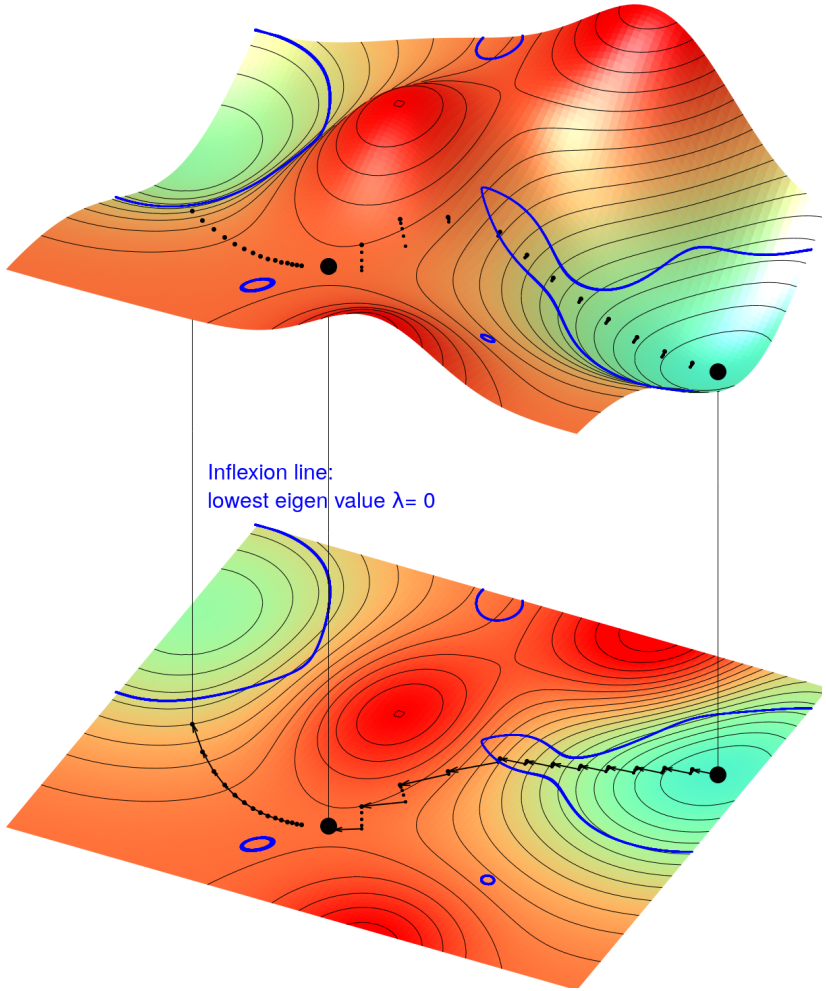




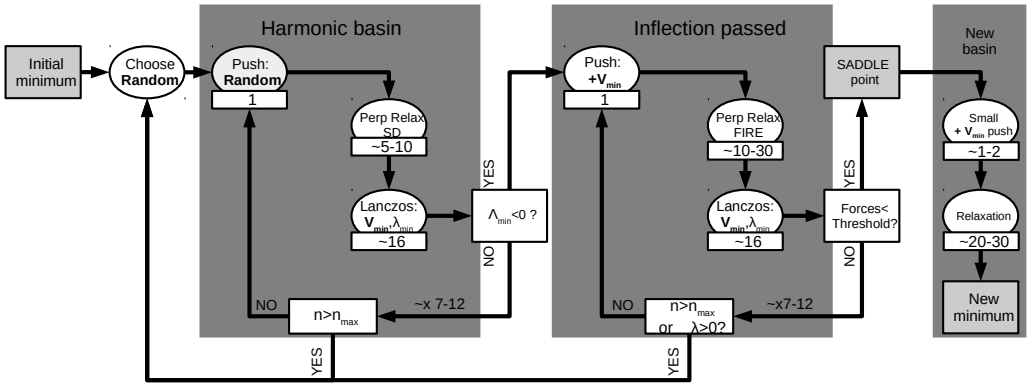
$$||Push|| = 0.1 * (init - saddle)$$

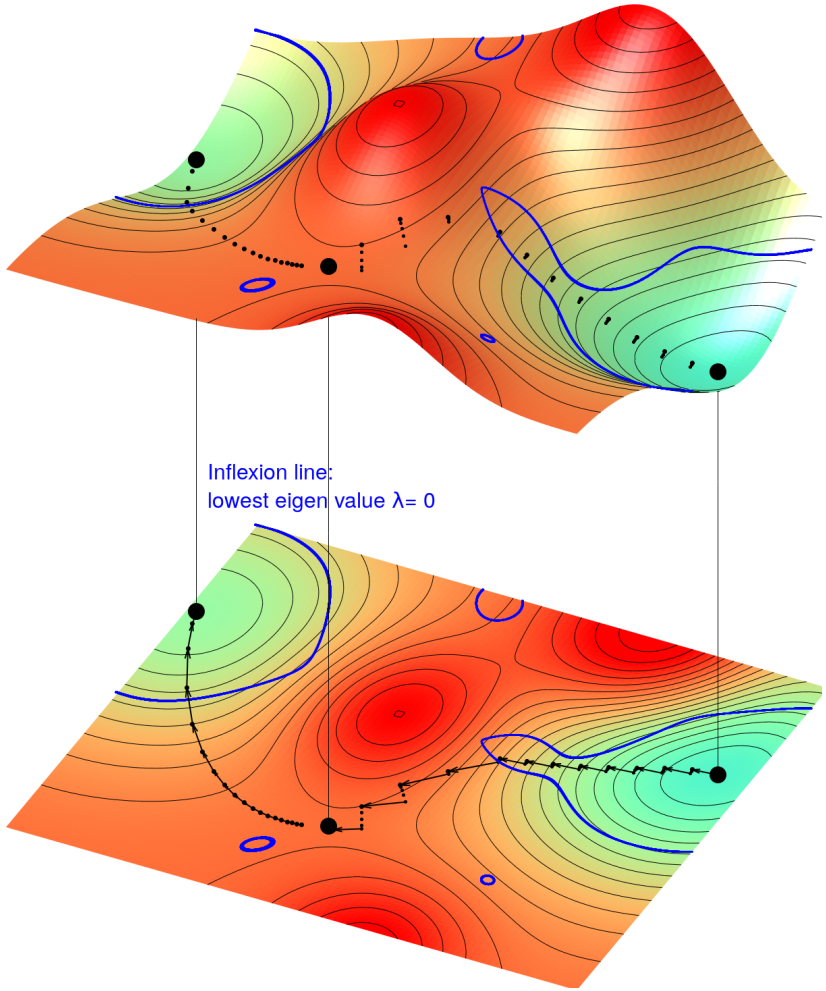
ARTn: Algorithm (1996)



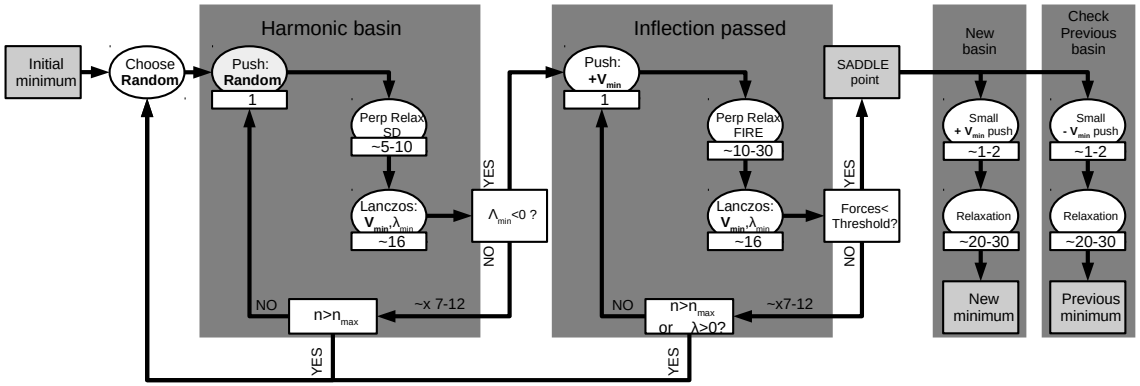


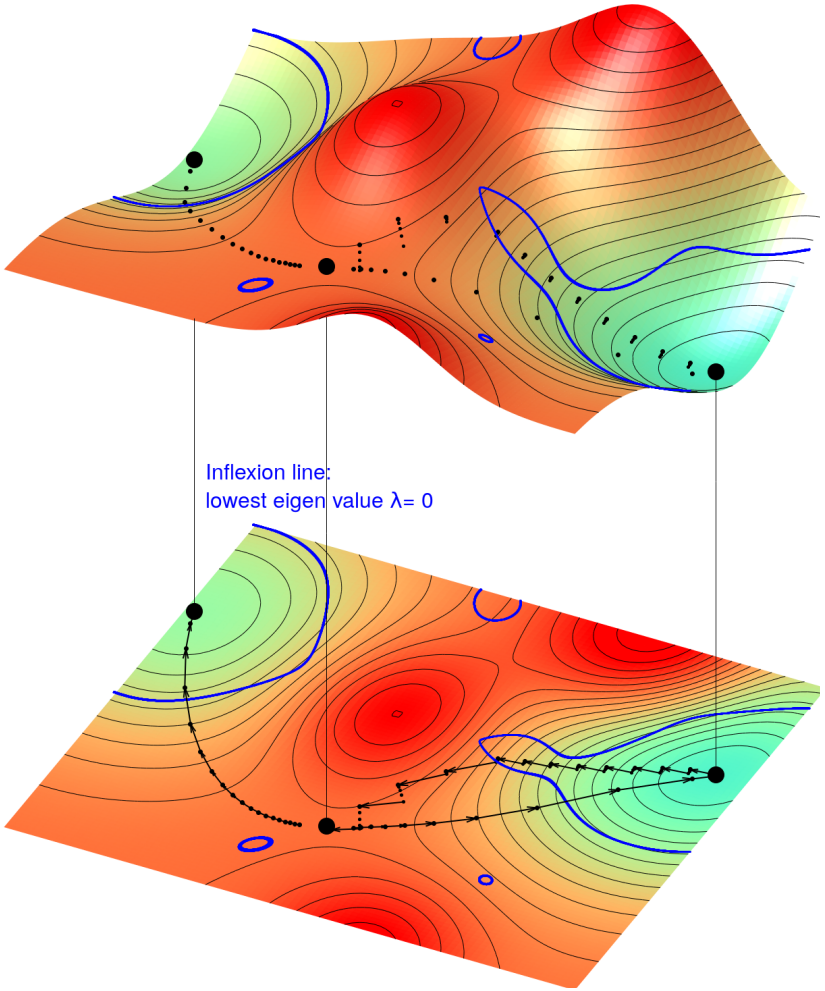
ARTn: Algorithm (1996)





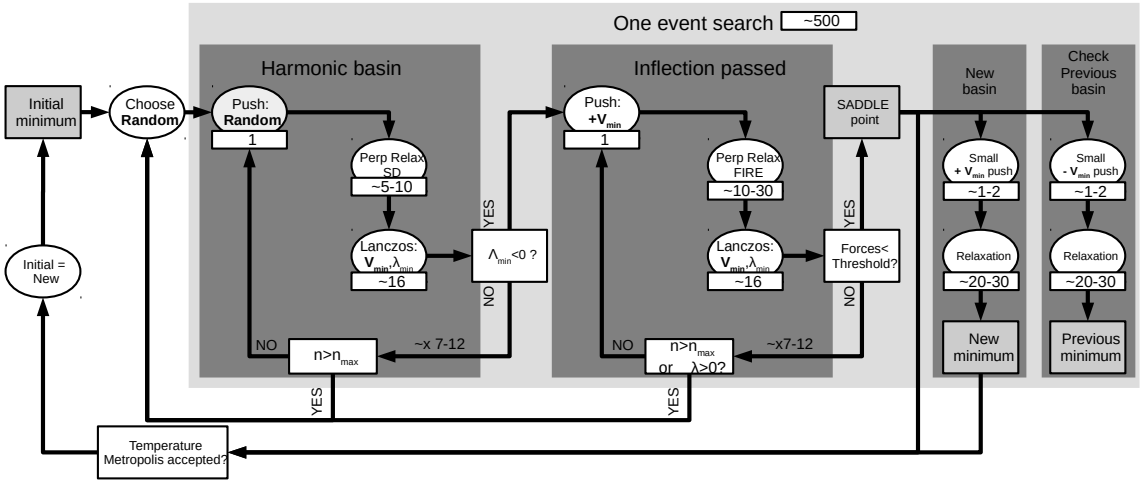
ARTn: Algorithm (1996)

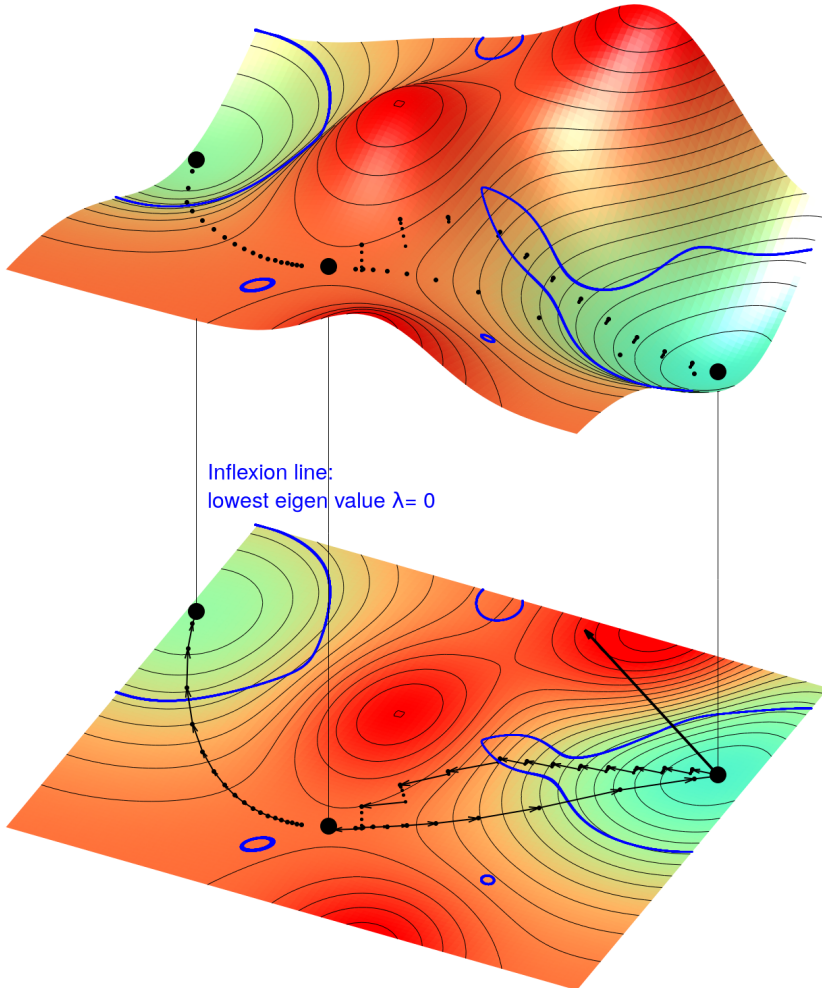




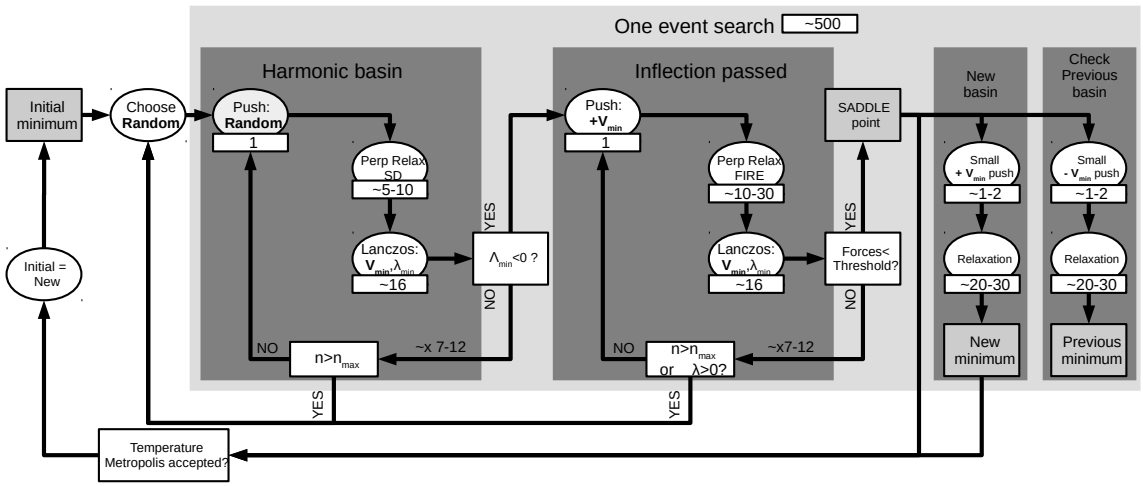
$$||Push|| = 0.1 * (init - saddle)$$

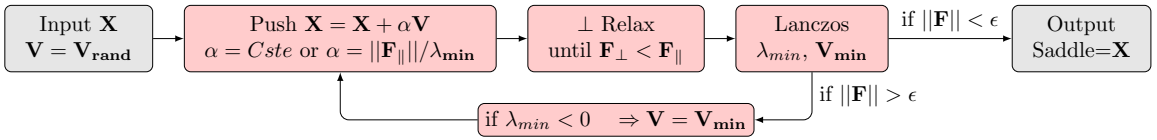
ARTn: Algorithm (1996)

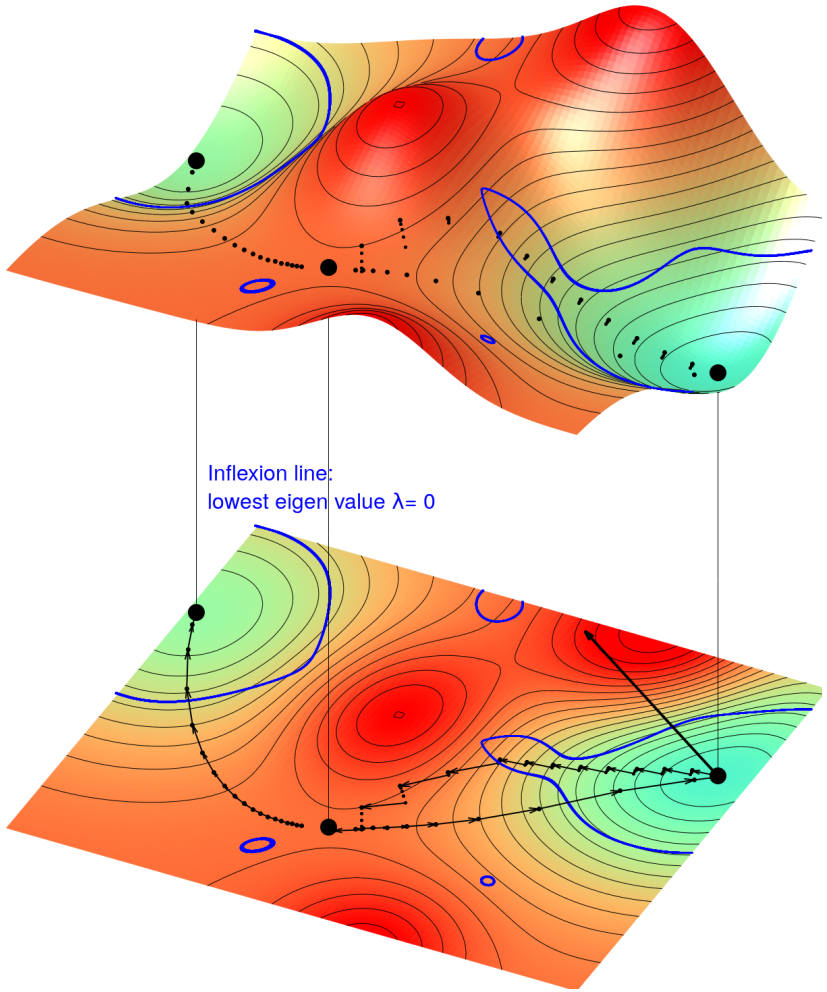




ARTn: Algorithm (1996)



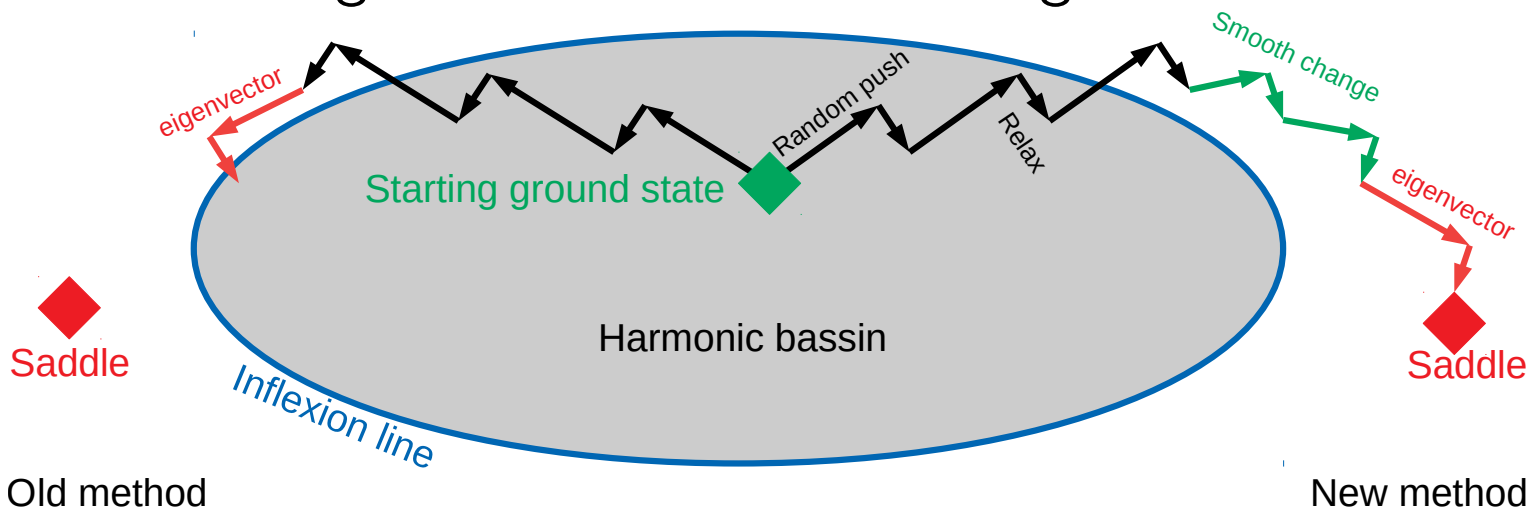




ARTn: Algorithm

ARTn: Facing Convex Regions

Avoid loosing EV back into the starting basin:



ARTn: Facing Convex Regions

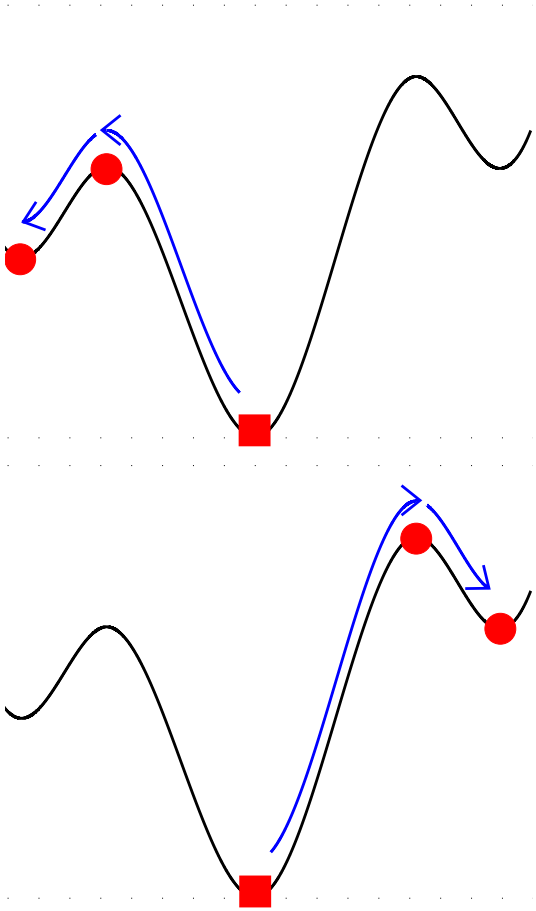
About 80% of the explorations in a-Si!!!

ARTn: Facing Convex Regions

Double random vector also solve cycling behavior!

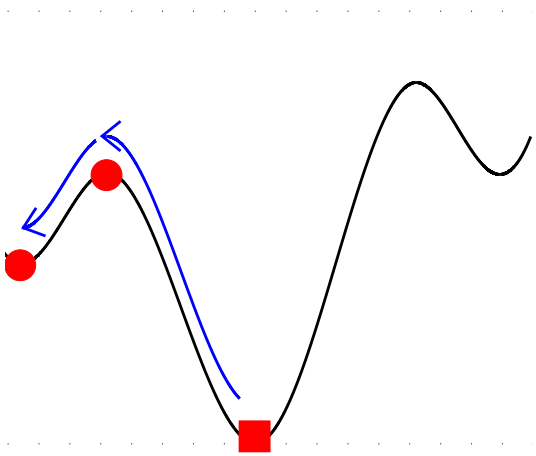
ARTn: possibilities

Explore the PES

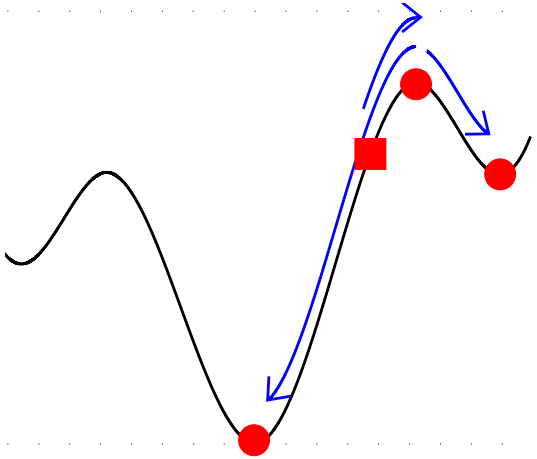


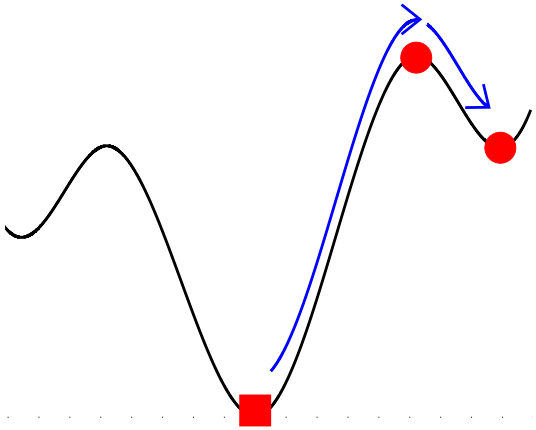
ARTn: possibilities

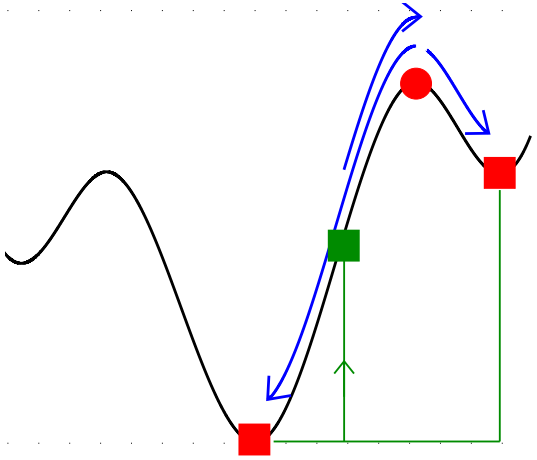
Explore the PES



Converge to saddle

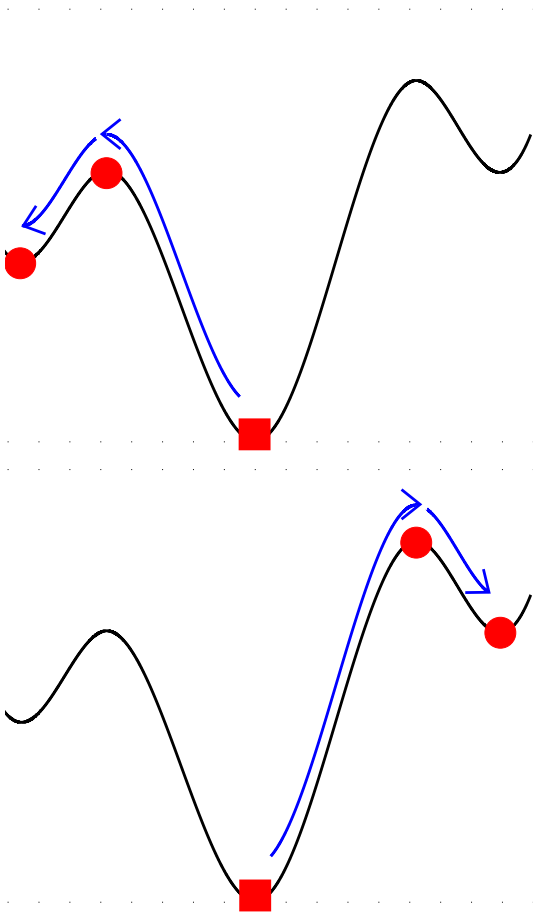




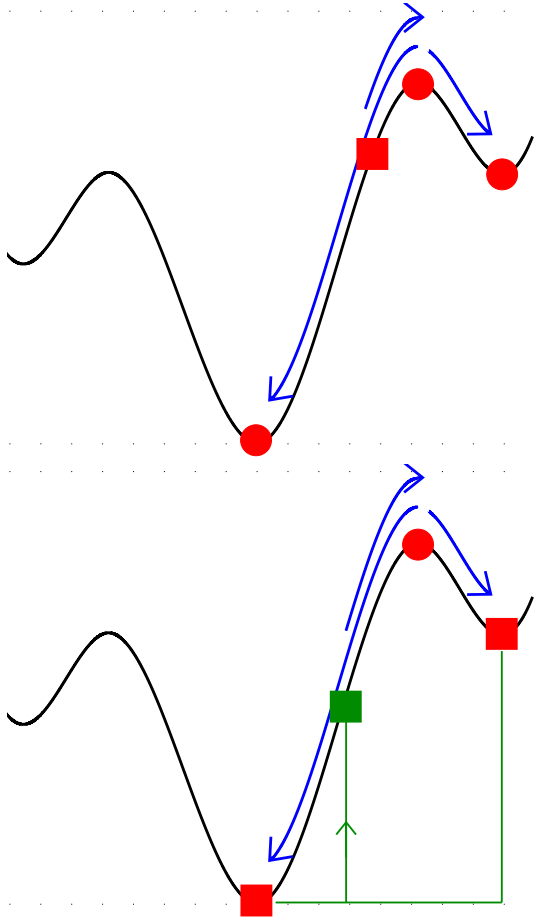


ARTn: possibilities

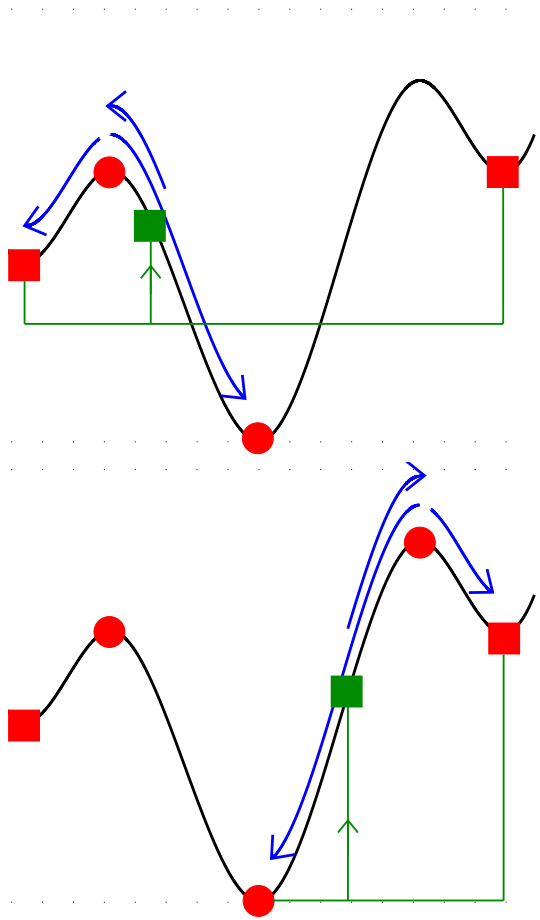
Explore the PES



Converge to saddle

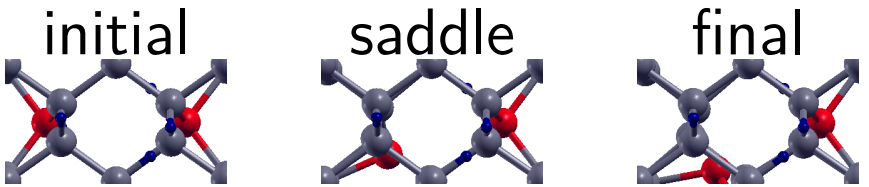


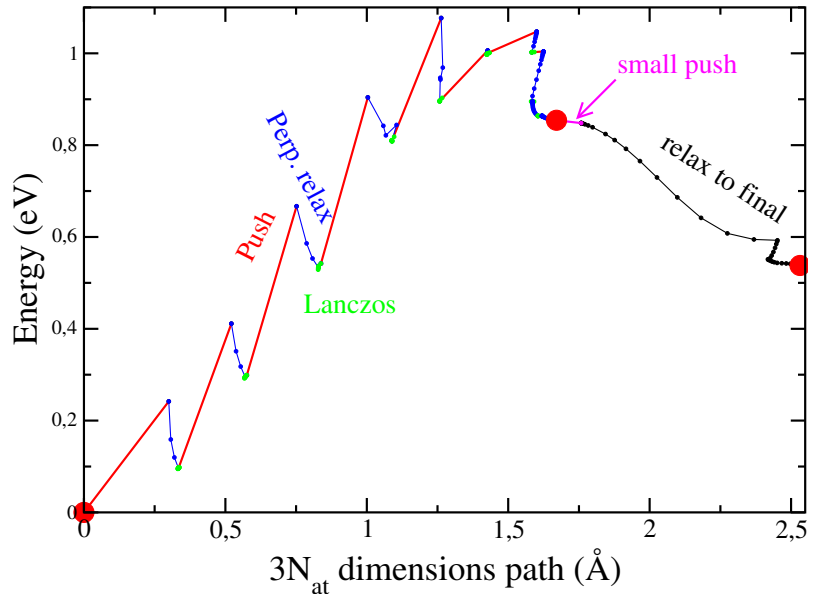
Construct a path



Real example

Metastable defect in silicon : 4V





arrows = forces

Lanczos: Why it works?

Lanczos → Hessian lowest eigenvalue without full Hessian

$$[H]^{-d}\mathbf{L} = \lambda_{min}^{-d} c_{min}\mathbf{V_{min}} + \sum_{i=2}^{3N_{at}} c_i \lambda_i^{-d}\mathbf{V_i}$$

$$= \lambda_{min}^{-d}\left(c_{min}\mathbf{V_{min}} + \underbrace{\sum_{i=2}^{3N_{at}} c_i \left(\frac{\lambda_i}{\lambda_{min}}\right)^{-d}\mathbf{V_i}}_{\xrightarrow[d\to+\infty]{} 0}\right)$$

$$\Rightarrow \lim_{d\to+\infty}[H]^{-d}\mathbf{L} \to\sim \mathbf{V_{min}}$$

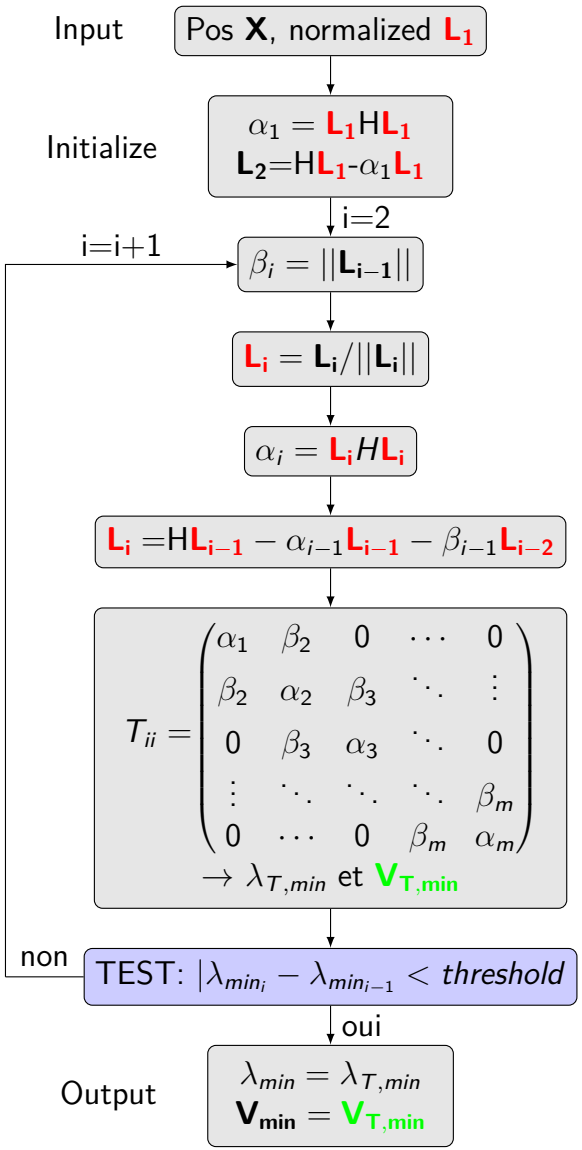
To get $[H]\mathbf{L_i}$:

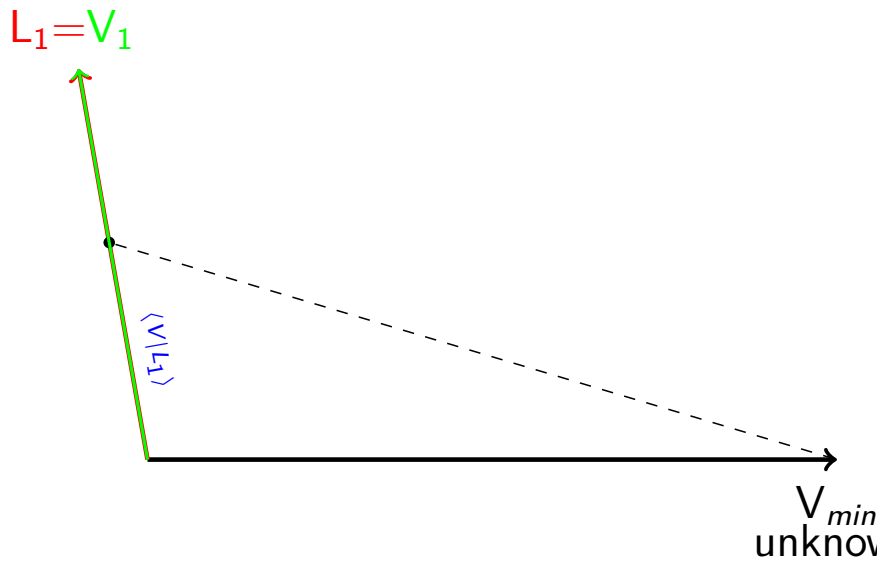
$$\mathbf{X_{L_i}} = \mathbf{X} + dr\mathbf{L_{i-1}}$$

$$\mathbf{F}(\mathbf{X_{L_i}}) = -\nabla E(\mathbf{X_{L_i}})$$

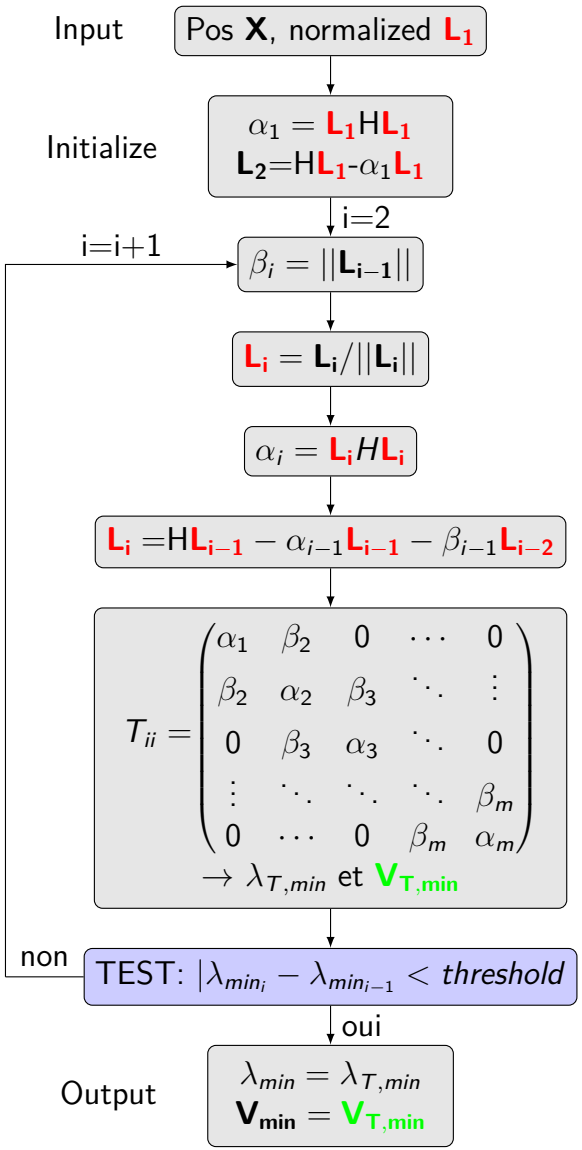
$$[H]\mathbf{L_i} = \mathbf{\Delta F_{L_i}} = \frac{\mathbf{F}(\mathbf{X_{L_i}}) - \mathbf{F}(\mathbf{X})}{dr}$$

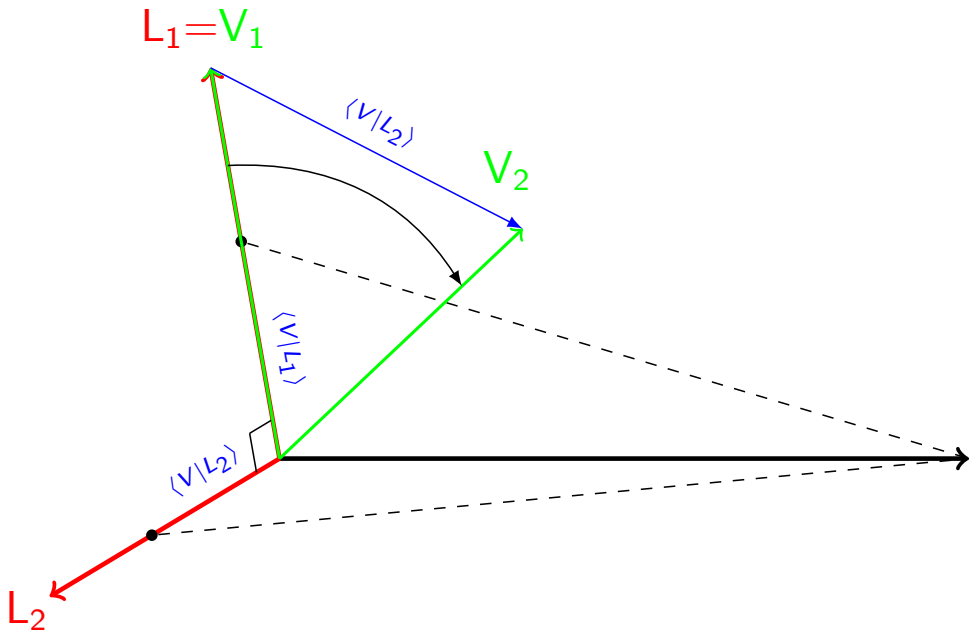
Lanczos: Algorithm



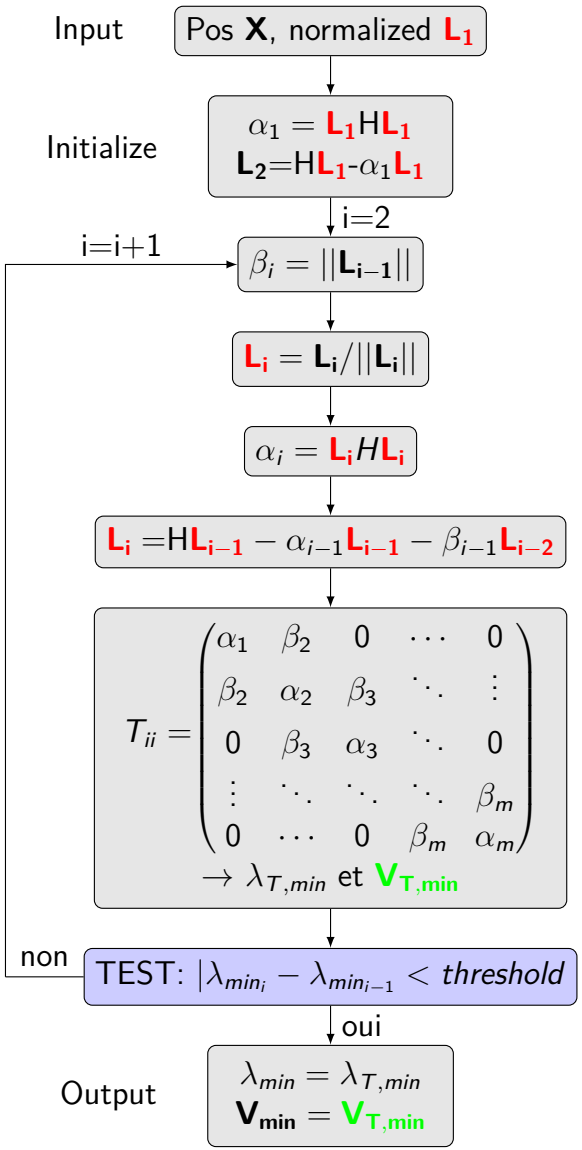


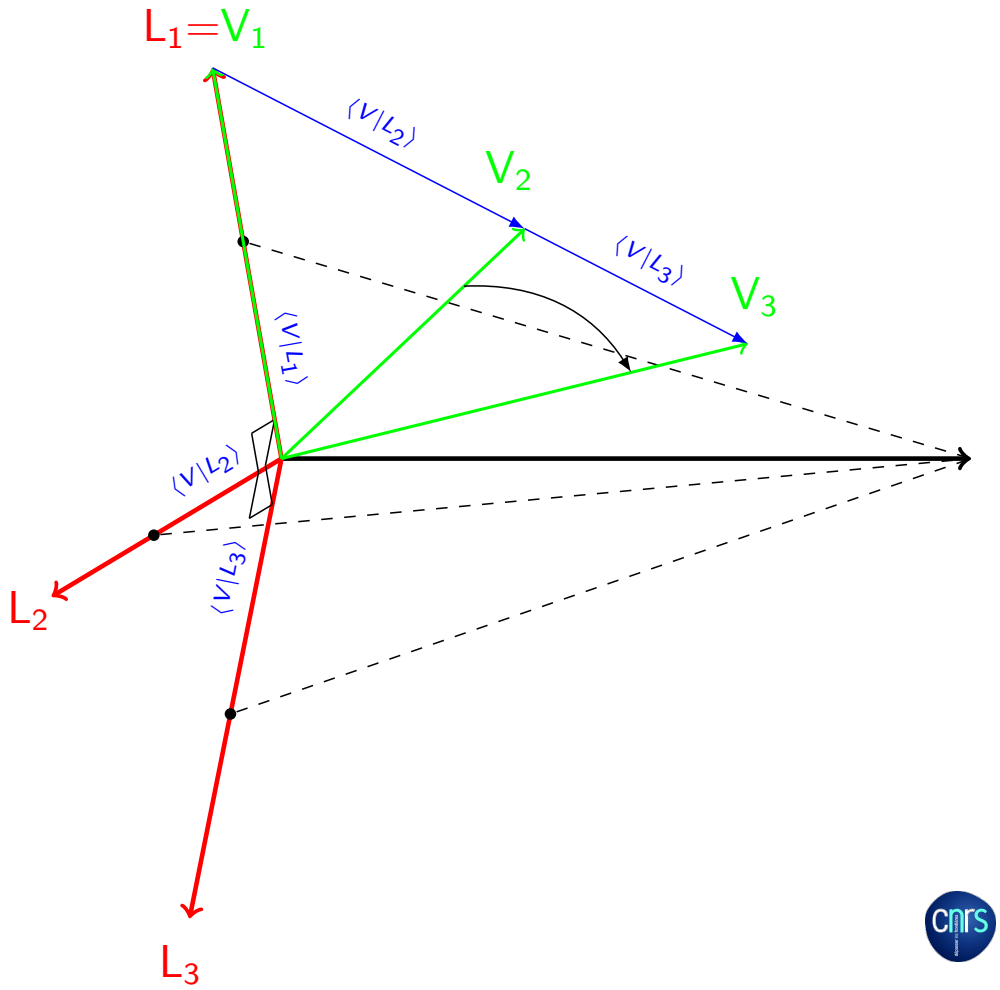
Lanczos: Algorithm



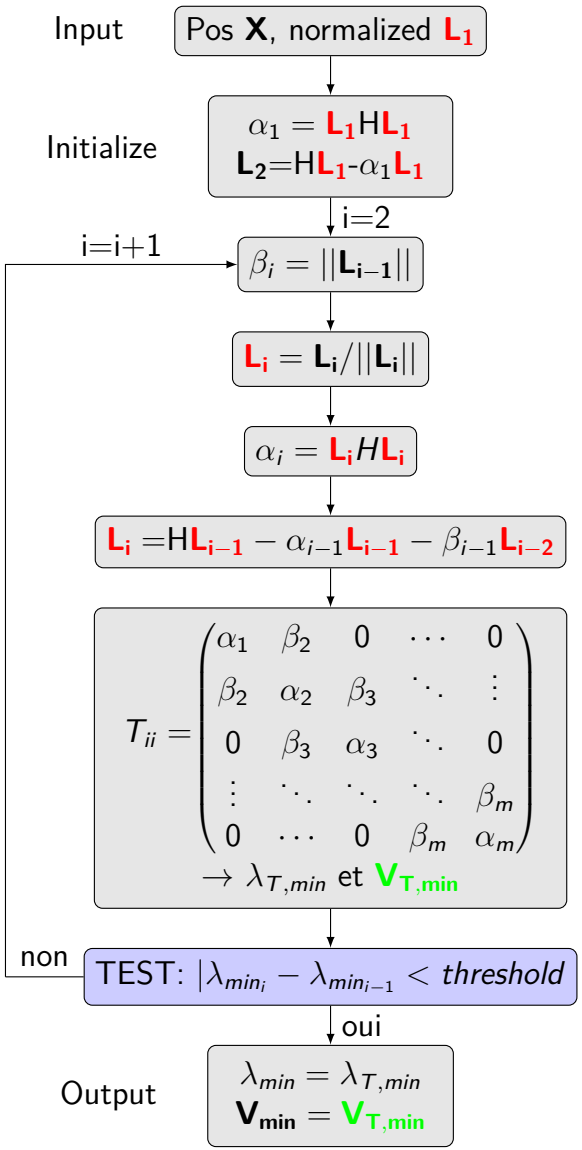


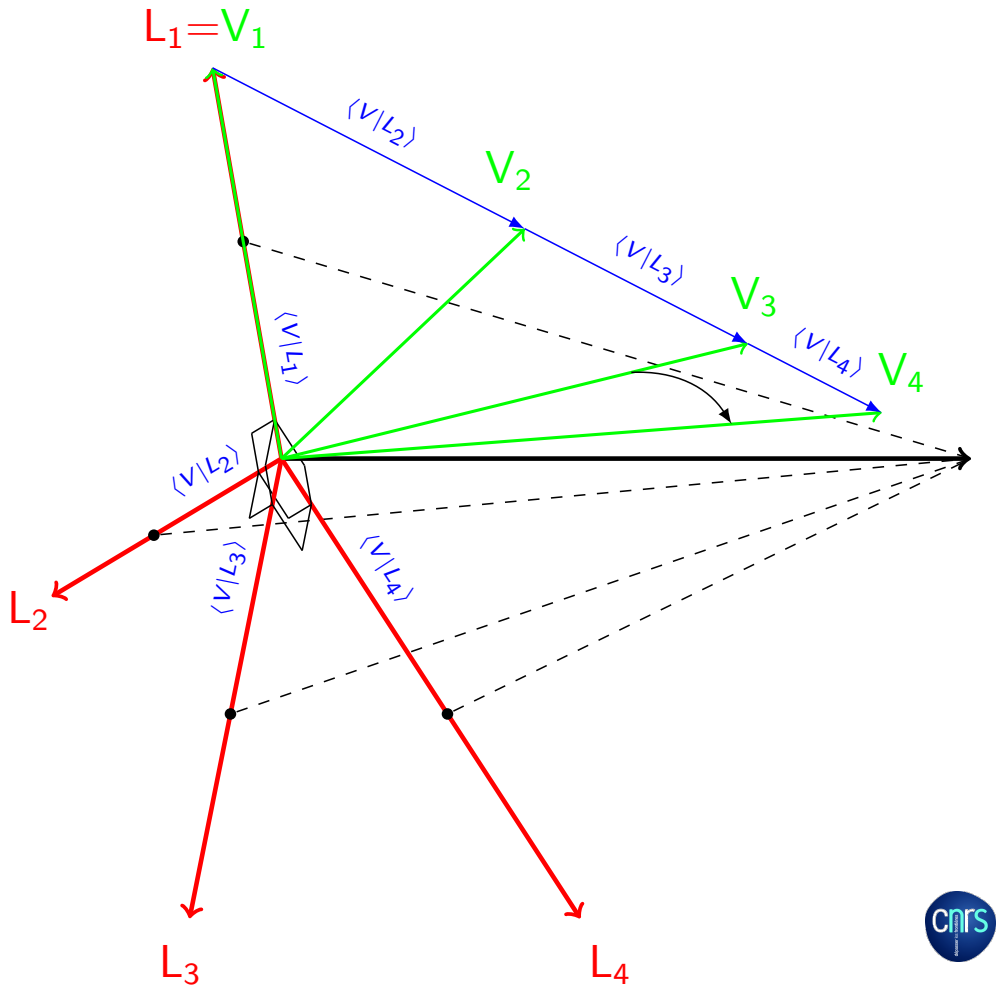
Lanczos: Algorithm





Lanczos: Algorithm





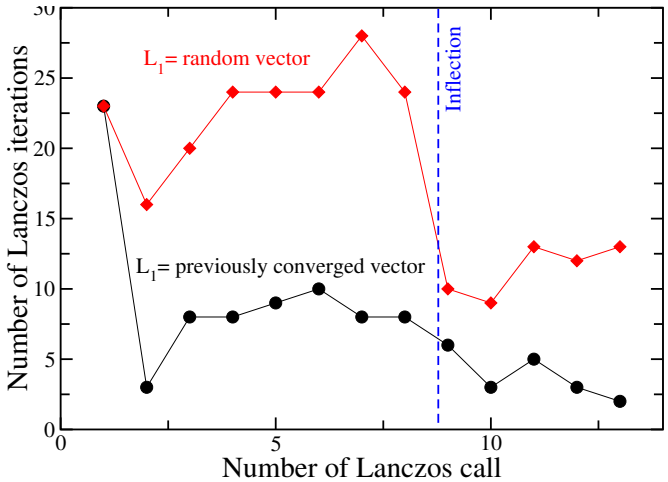
Lanczos: Algorithm

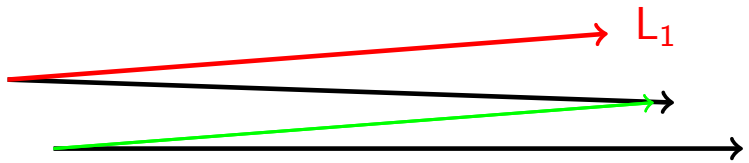
1rst Lanczos call:

$\mathbf{L_1}$ = random

i-th Lanczos call:

$\mathbf{L_1}$ = converged at call$_{i-1}$





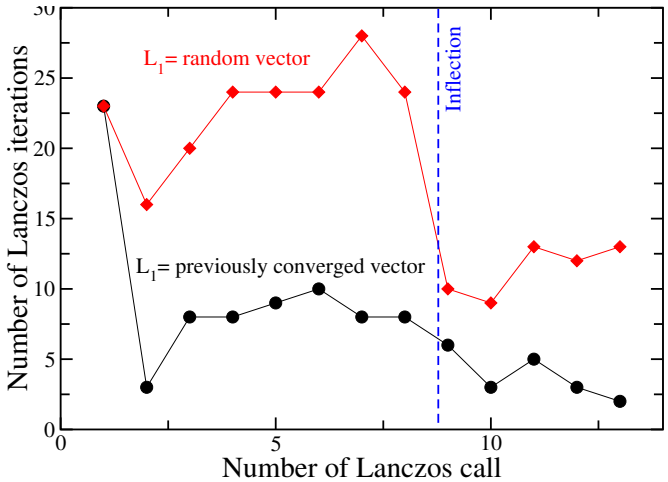
Lanczos: Algorithm

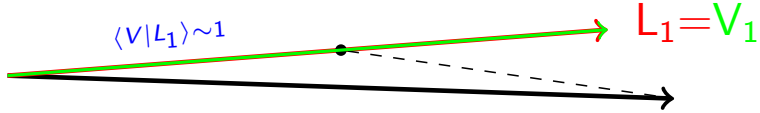
1rst Lanczos call:

$\mathbf{L_1}$ = random

i-th Lanczos call:

$\mathbf{L_1}$ = converged at call$_{i-1}$





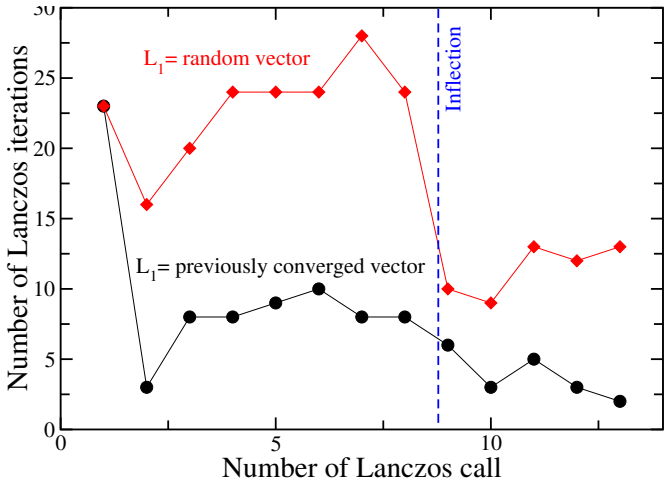
Lanczos: Algorithm

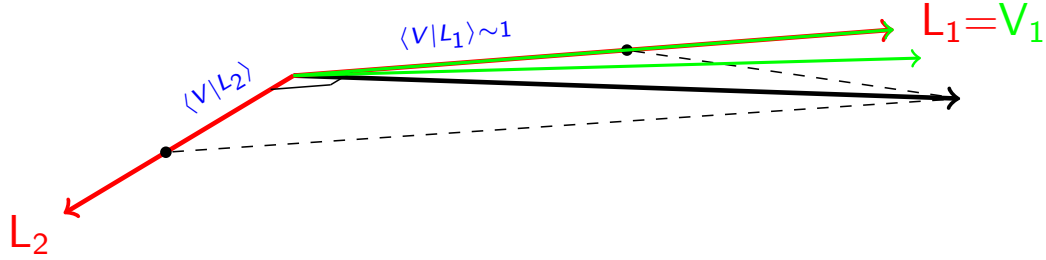
1rst Lanczos call:

$\mathbf{L_1}$ = random

i-th Lanczos call:

$\mathbf{L_1}$ = converged at call$_{i-1}$





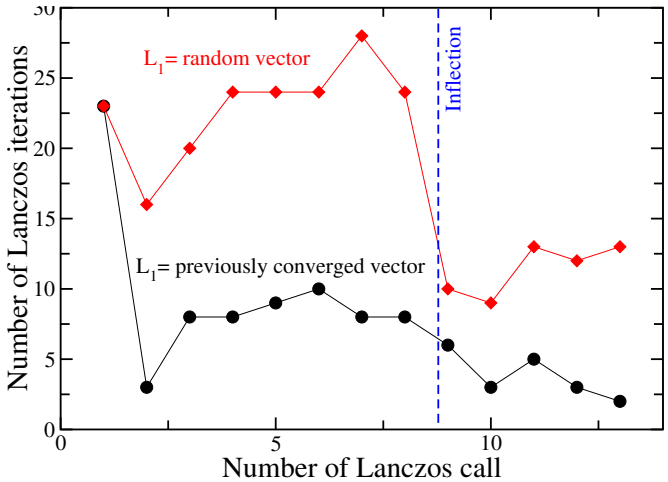
Lanczos: Algorithm

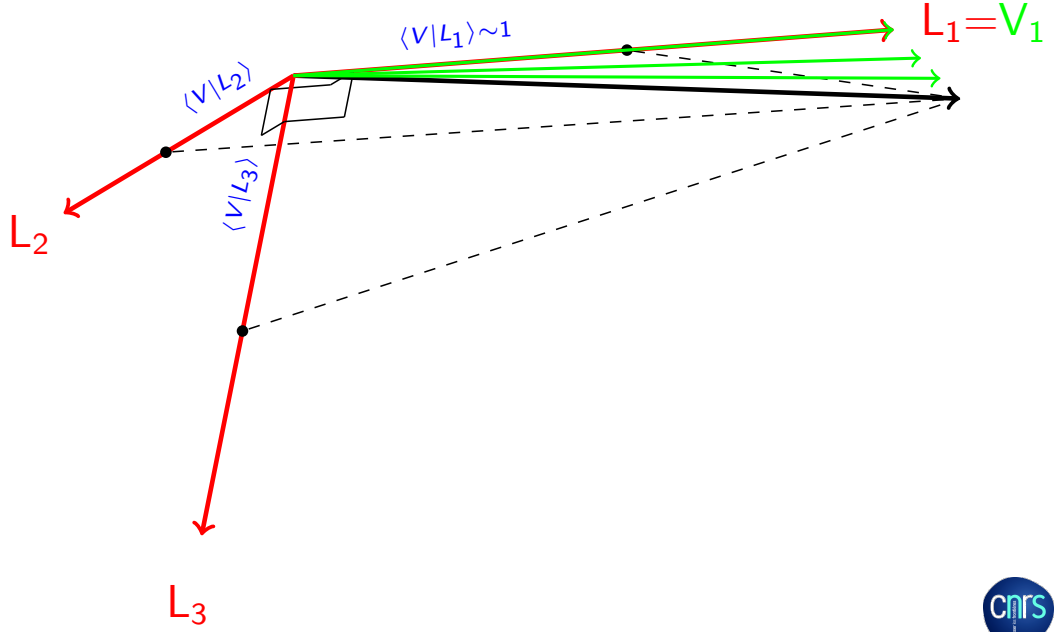
1rst Lanczos call:

$\mathbf{L_1}$ = random

i-th Lanczos call:

$\mathbf{L_1}$ = converged at call$_{i-1}$





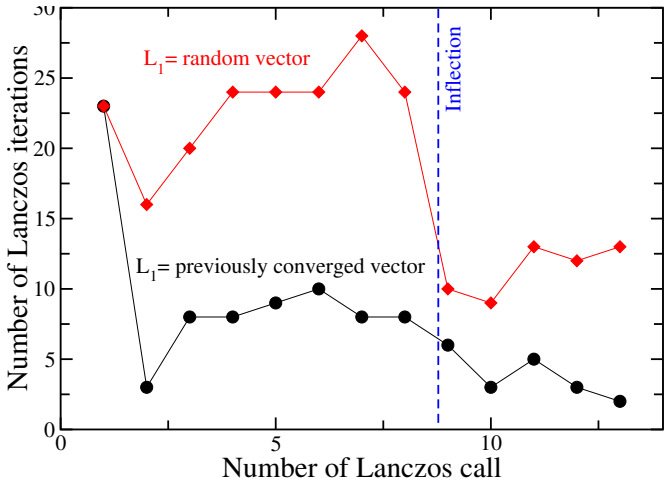
Lanczos: Algorithm

1rst Lanczos call:

$\mathbf{L_1}$ = random

i-th Lanczos call:

$\mathbf{L_1}$ = converged at call$_{i-1}$



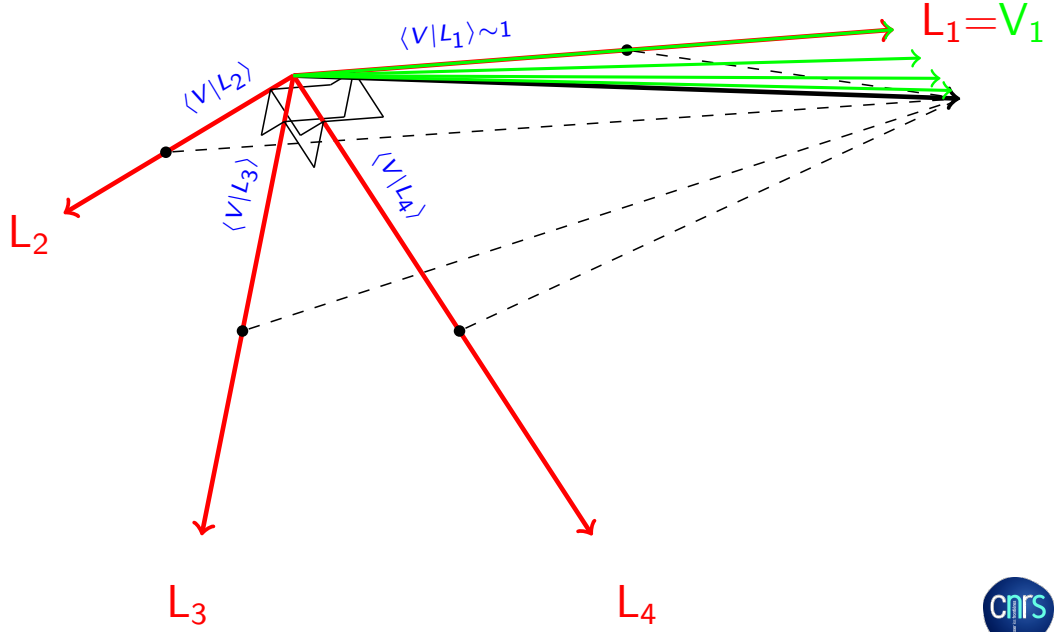
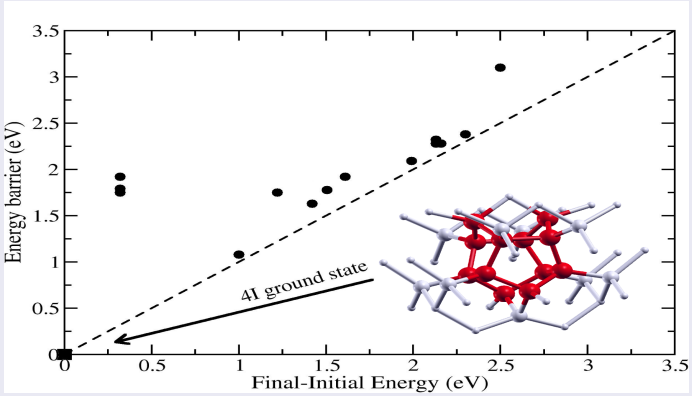


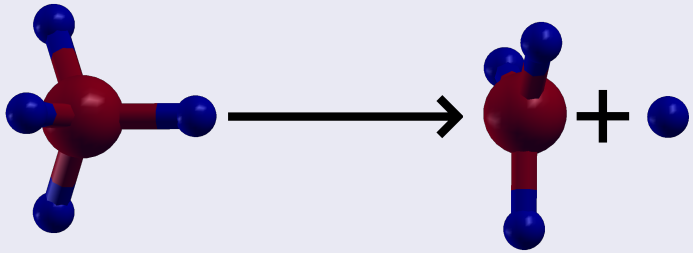
Table of Contents

Applications

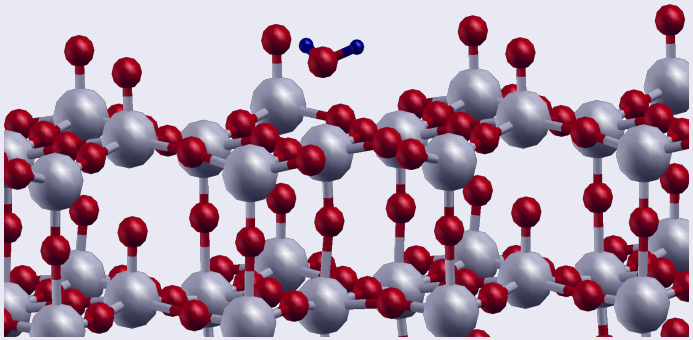
Metastable defects : 4I



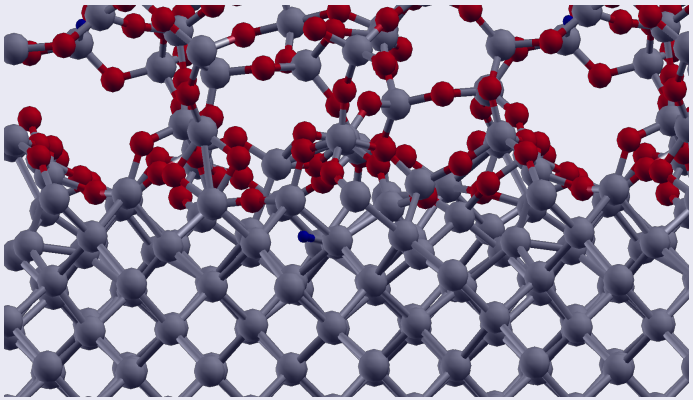
Molecules : $CH_4 \rightarrow CH_3 + H$



Surfaces : H_2O on WO_3

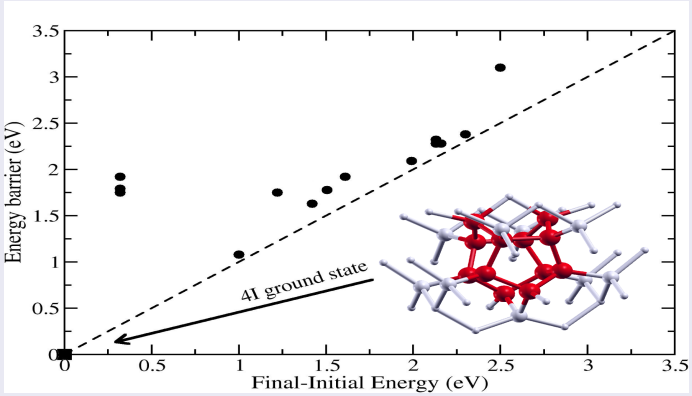


Interfaces : Si/SiO_2

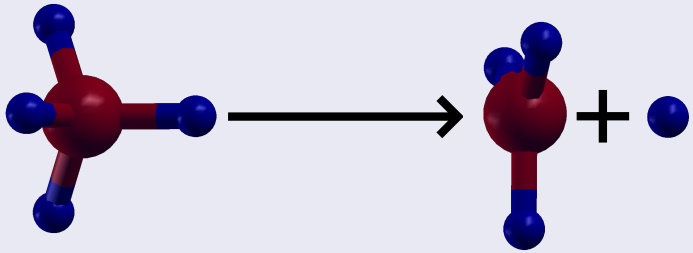


Applications

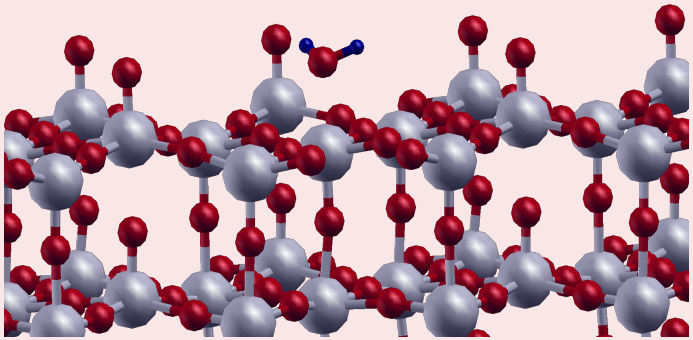
Metastable defects : 4I



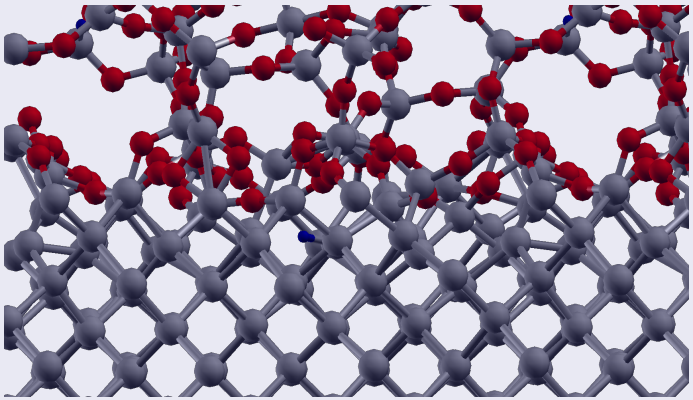
Molecules : $CH_4 \rightarrow CH_3 + H$



Surfaces : H_2O on WO_3

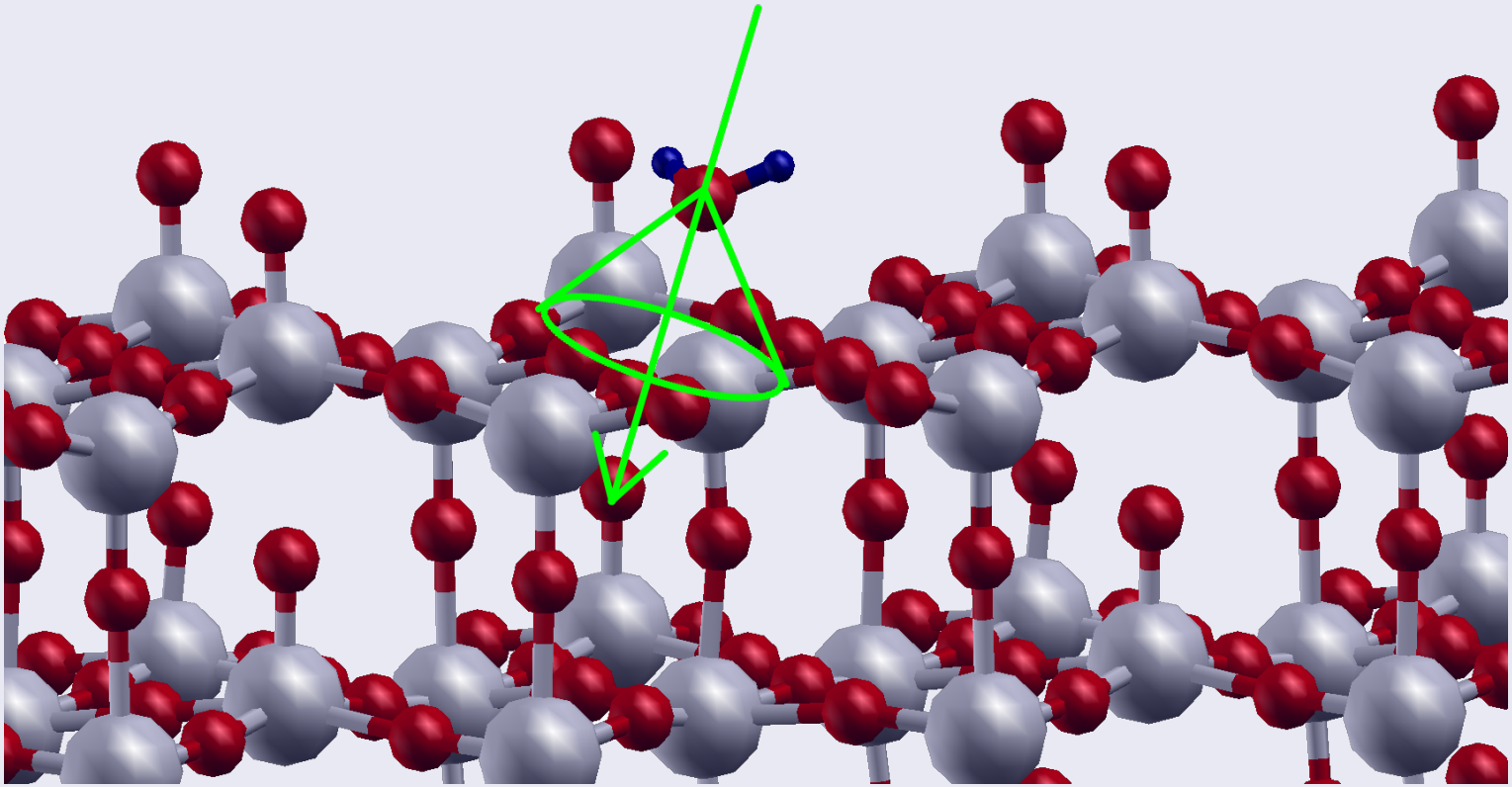


Interfaces : Si/SiO_2



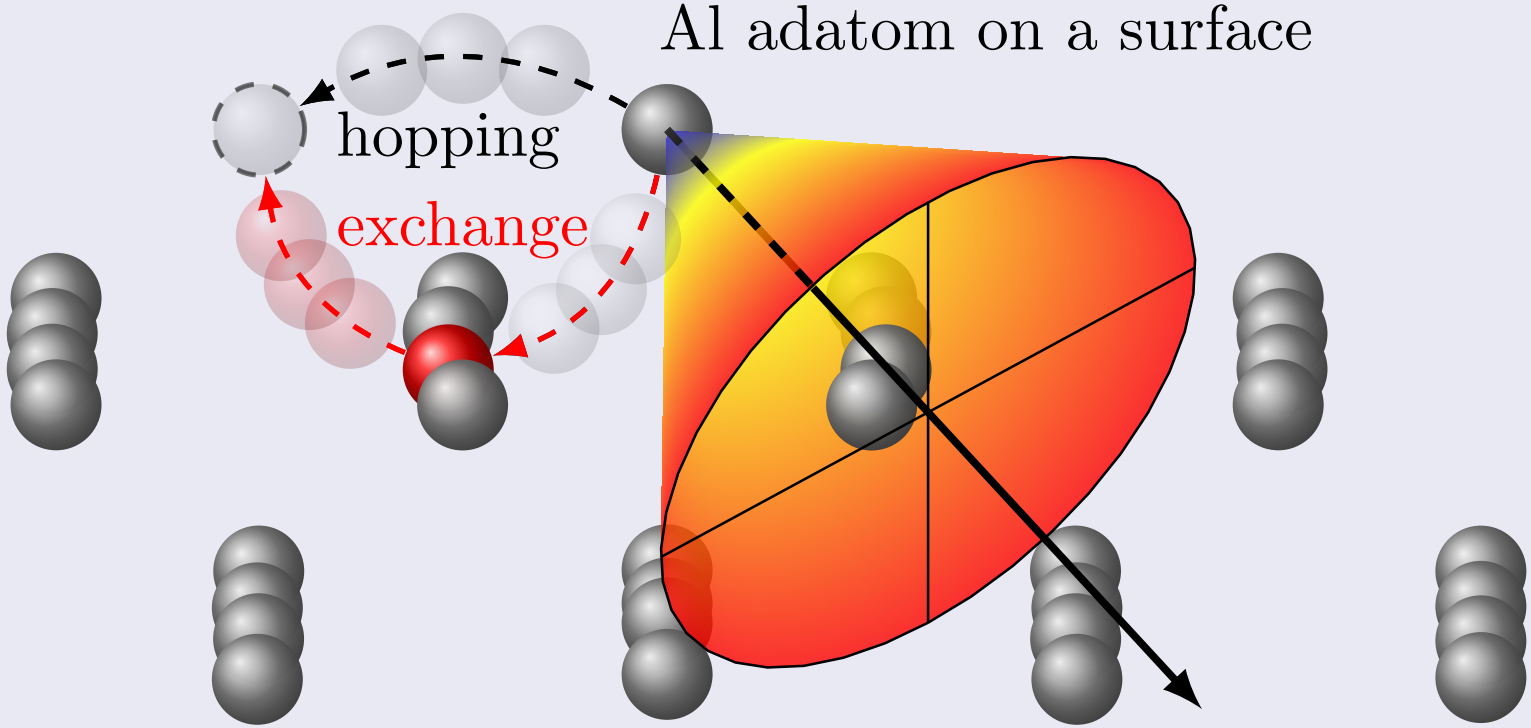
Restrict dimensions

Reaction on Surfaces: User guess



Restrict dimensions

Impose random conical direction



Restrict dimensions

Impose central atom(s)

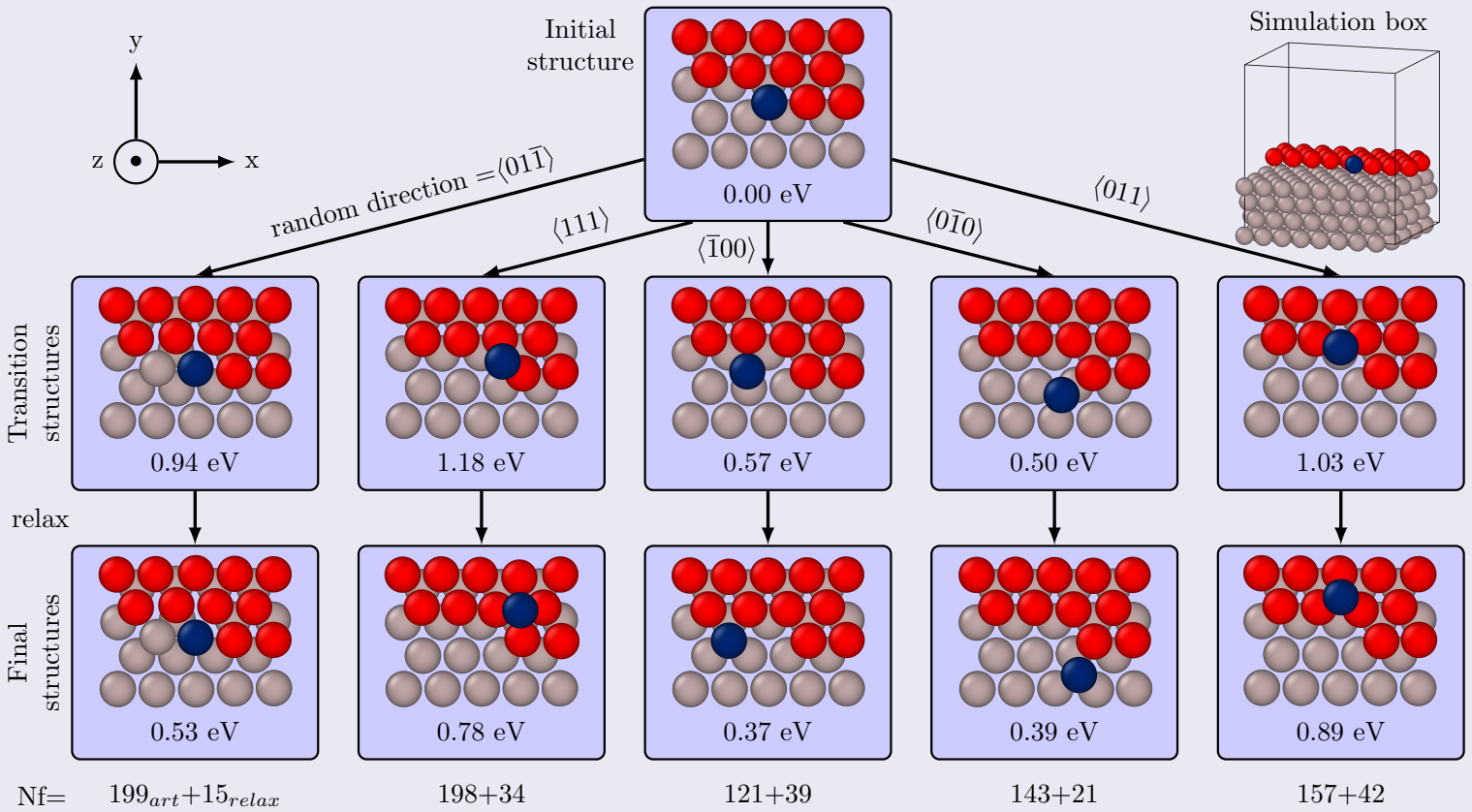
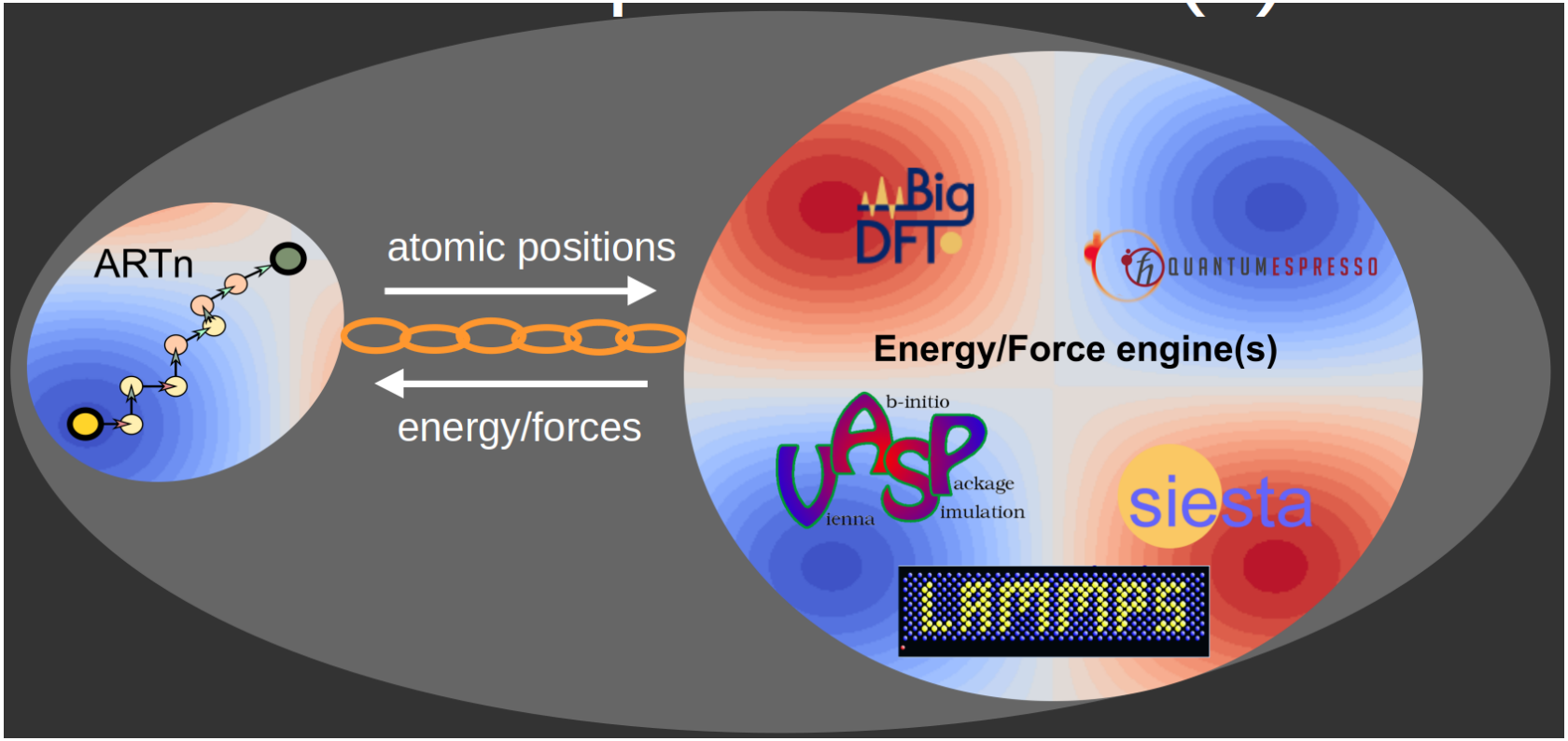
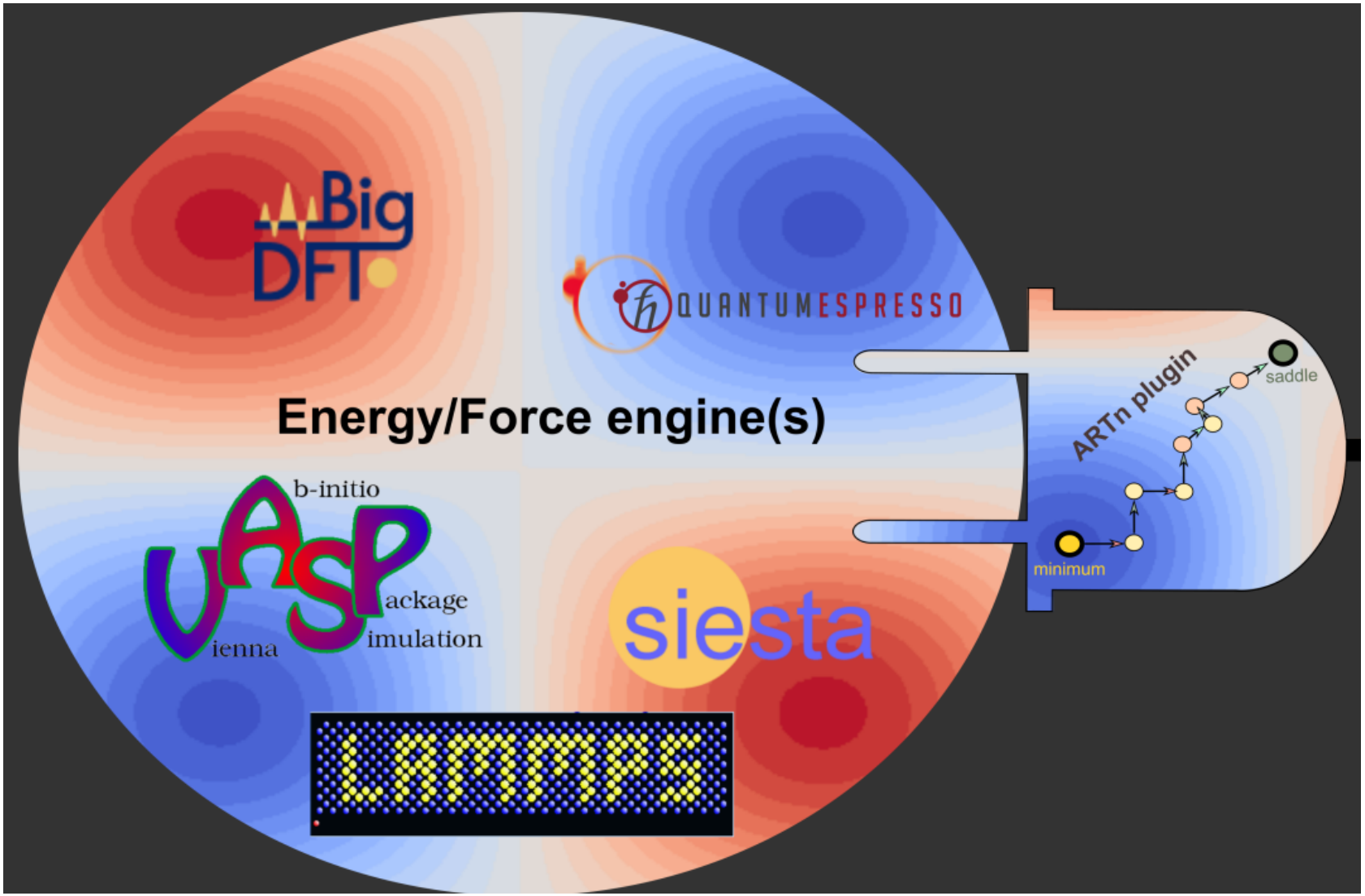


Table of Contents

Implementation

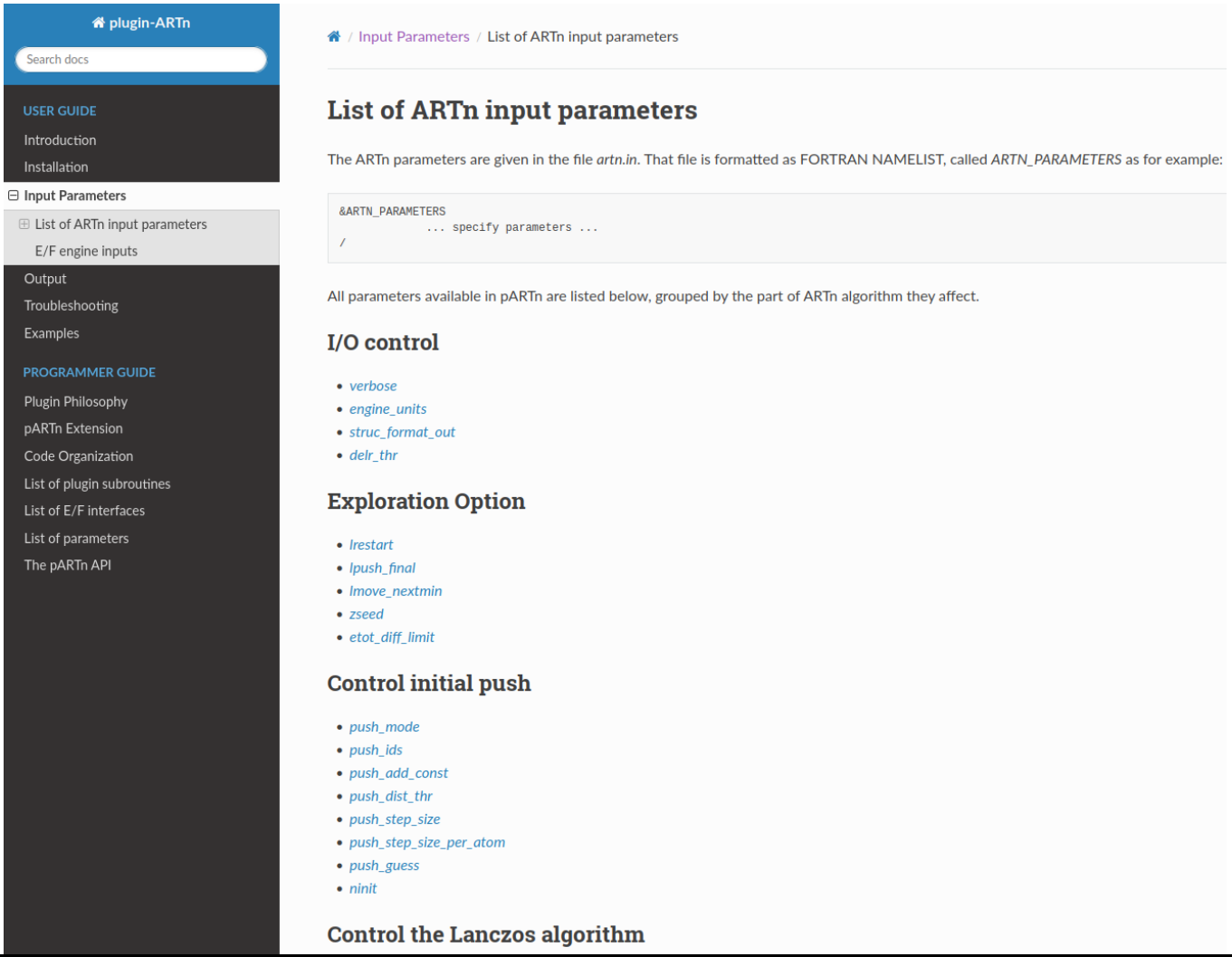


Implementation



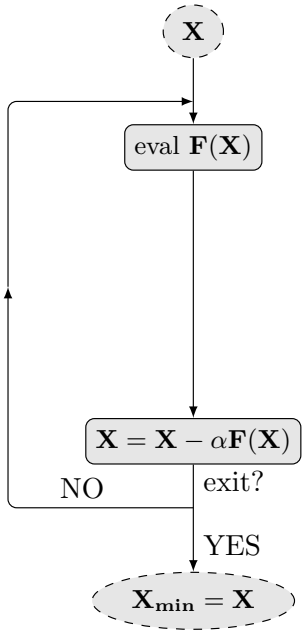
Free access to pART code and doc

https://gitlab.com/mammasmias/artn-plugin/

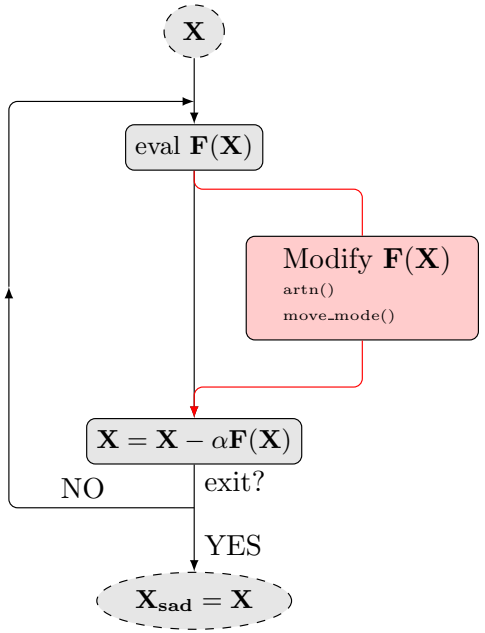


Add plug to get SP: biased minimization

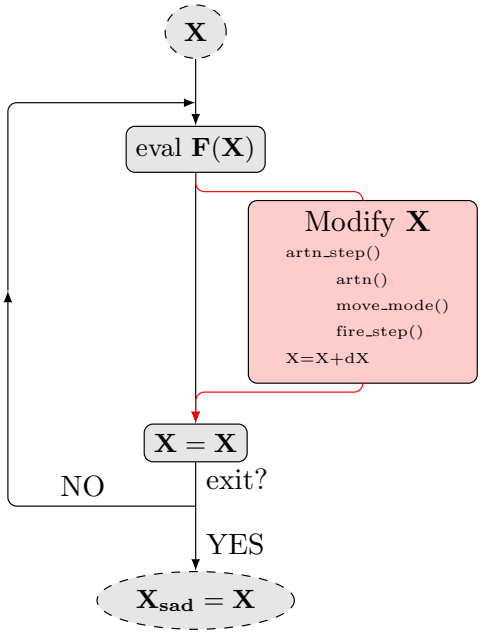
Force engine



integrator = FIRE



integrator = no



Minimal modification of the force engine:

- 1 call to routine
- 1 flag for the compilation
- 1 include location of the library

typical scripts

LAMMPS

```
units metal
dimension 3
boundary p p p
atom_style atomic
atom_modify sort 0 1

read_data Al_vac.data

pair_style eam/alloy
pair_coeff * * AlO.eam.alloy Al

plugin load ../../lib/libartn-lmp.so
fix 10 all artn alpha0 0.2 dmax 5.0

min_style fire
minimize 1e-4 1e-5 1000 10000
```

Execution:

```
mpirun -np 4 lmp_mpi -in lammps.in
```

Quantum Espresso

```
&CONTROL
calculation = 'relax',
/
&SYSTEM
celldm(1) = 7.6533908,...
/
&ELECTRONS
/
&IONS
ion_dynamics = 'fire',
/
ATOMIC_SPECIES
Al 1.0 Al.pz-vbc.UPF
ATOMIC_POSITIONS (angstrom)
Al 0.01 0.00 -0.07
...
K_POINTS gamma
```

Execution:

```
mpirun -np 4 pw.x -partn < relax.in
```

Table of Contents

Install Force engines

Install LAMMPS:

```
git clone -b release https://github.com/lammps/lammps.git mylammps
cd mylammps/src/
make yes-MANYBODY yes-reaxff yes-plugin yes-MEAM yes-misc yes-EXTRA-PAIR
make -j4 mode=shared mpi
```

Give the relative path:

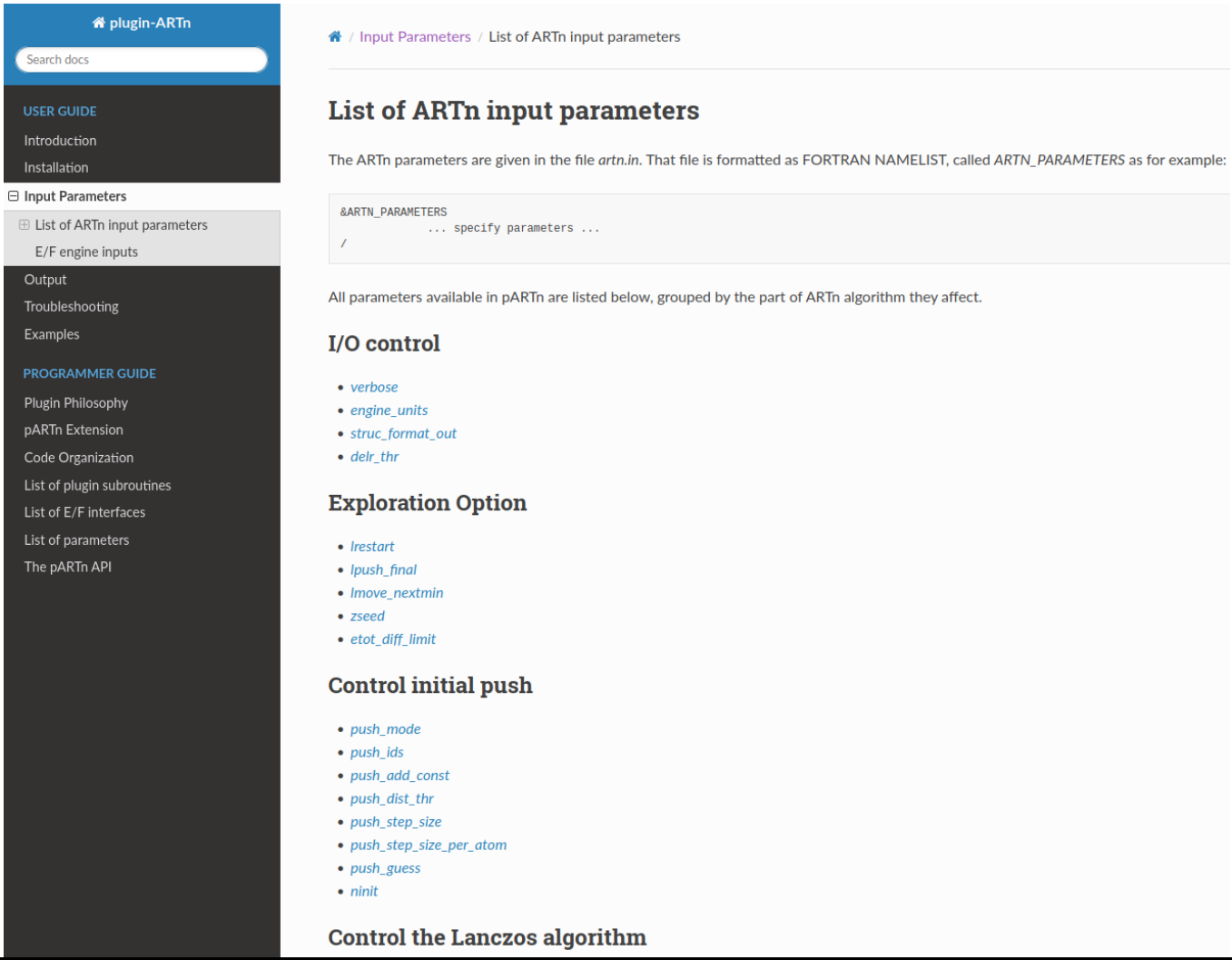
```
vi ~/.bashrc
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/me/mylammps/src/
source ~/.bashrc
```

Install QuantumEspresso-7.3.1
Inscription + Download at
https://www.quantum-espresso.org/download-page/

```
tar -zxvf qe-7.3-ReleasePack.tar.gz
cd qe-7.3/
./configure --enable-legacy_plugins
make -j4 pw
```

Free access to pART code and doc

https://gitlab.com/mammasmias/artn-plugin/



Download pART

```
git clone https://gitlab.com/mammasmias/artn-plugin/
cd artn-plugin
ll
```

```
ajay@lillou:~/Programmes$ cd artn-plugin/
ajay@lillou:~/Programmes/artn-plugin$ ll
total 116
-rw-r--r--  1 ajay m3  5768 Jul  4 09:14 CMakeLists.txt
drwxr-xr-x  2 ajay m3  4096 Jul 30 15:53 Files_LAMMPS/
drwxr-xr-x  4 ajay m3  4096 Jul  4 09:14 Files_QE/
drwxr-xr-x  2 ajay m3  4096 Jul  4 09:14 Files_Siesta/
-rw-r--r--  1 ajay m3 10175 Jul  4 09:14 LICENSE_Apache_v2.txt
-rw-r--r--  1 ajay m3 35149 Jul  4 09:14 LICENSE_gpl-3.0.txt
-rw-r--r--  1 ajay m3  2242 Jul  4 09:14 MANUAL.rst
-rw-r--r--  1 ajay m3  3236 Jul  4 09:14 Makefile
-rw-r--r--  1 ajay m3  3872 Jul  4 09:14 README.rst
-rw-r--r--  1 ajay m3   655 Jul  4 09:14 TERMS_OF_USE
-rw-r--r--  1 ajay m3  1640 Jul  4 09:14 TODO.md
drwxr-xr-x  2 ajay m3  4096 Jul  4 09:14 cmake/
drwxr-xr-x  9 ajay m3  4096 Jul 31 09:20 docs/
-rw-r--r--  1 ajay m3   616 Jul 30 15:53 environment_variables
drwxr-xr-x 21 ajay m3  4096 Jul  4 13:37 examples/
drwxr-xr-x  3 ajay m3  4096 Jul 31 09:52 interface/
drwxr-xr-x  2 ajay m3  4096 Jul 30 15:53 lib/
drwxr-xr-x  3 ajay m3  4096 Jul 30 15:53 src/
```

Fill compilers and paths

```
vi environment_variables
```

```
# The following scripts is need for the user to set environment variables

# COMPILER
F90=gfortran -pedantic -std=f2018
CXX=mpicxx
CC=mpicc

#
# ARTn PATH
ART_PATH=`pwd`
#
# Path to the Energy and Force engines
#
# Quantum ESPRESSO
QE_PATH=/home/ajay/Programmes/qe-7.3/

# LAMMPS
LAMMPS_PATH=/home/ajay/Programmes/mylammps/


# prefix for running the examples in parallel, i.e., PARA_PREFIX = "mpirun -np 4"
PARA_PREFIX="mpirun -np 4"


## Path for the blas and fortran library
# BLAS
BLAS_LIB= /usr/lib/x86_64-linux-gnu/libopenblas.a  -lpthread

# FORTAN LIB
FORT_LIB= /usr/lib/x86_64-linux-gnu/libgfortran.so.5
```

```
pwd
sudo apt-get install libopenblas-dev
find /usr/lib/ /usr/local/lib -name "*openblas*"      (blas+lapack+cblas)
find /usr/lib/ /usr/local/lib -name "libgfortran*"
```

Compile artn library and caller routines

```
make clean
make lib
```

```
a - Obj/block_pushover.o
a - Obj/block_finalize.o
a - Obj/start_guess.o
a - Obj/diag.o
a - Obj/restart.o
a - Obj/serialize.o
a - Obj/m_fire.o
a - Obj/artn_api2.o
a - Obj/artn_siesta.o
a - Obj/artn_step.o
a - Obj/artn_QE.o

LIBRARY COMPILATION: SUCCESS!!

gfortran -pedantic -std=f2018 -shared  -o libartn.so Obj/precision.o Obj/m_tools.o Obj/units.o Obj/m_artn_data.o Obj/m_error.o Obj/string_tools.o Obj/permute.o Obj/art
n_params_mod.o Obj/convert_units.o Obj/m_datainfo.o Obj/m_block_lanczos.o Obj/m_artn_report.o Obj/m_option.o Obj/write_report.o Obj/set_params.o Obj/get_params.o Obj/s
et_runparam.o Obj/get_runparam.o Obj/set_data.o Obj/get_data.o Obj/fill_param_step.o Obj/check_force_convergence.o Obj/displacement_validation.o Obj/pbc.o Obj/compute_
delr.o Obj/center.o Obj/sum_force.o Obj/perpforce.o Obj/smooth_interpol.o Obj/push_over_procedure.o Obj/move_mode.o Obj/write_struct.o Obj/save_step_data.o Obj/nextmin
.o Obj/make_filename.o Obj/nperp_limitation.o Obj/random.o Obj/check_artn_params.o Obj/block_lanczos.o Obj/lanczos.o Obj/m_setup_artn.o Obj/clean_artn.o Obj/artn.o Obj
/push_init.o Obj/block_pushinit.o Obj/block_perprelax.o Obj/block_pusheigen.o Obj/block_pushover.o Obj/block_finalize.o Obj/start_guess.o Obj/diag.o Obj/restart.o Obj/
serialize.o Obj/m_fire.o Obj/artn_api2.o Obj/artn_siesta.o Obj/artn_step.o Obj/artn_QE.o /usr/lib/x86_64-linux-gnu/libopenblas.a  -lpthread  # Needs BLAS_LIB for compi
lation on macos
make[1]: Leaving directory '/home/ajay/Programmes/artn-plugin/src'
/home/ajay/Programmes/artn-plugin
ln -sf ../src/libartn.a ./lib/libartn.a
ln -sf ../src/libartn.so ./lib/libartn.so
ln -sf ../src/libartn.a ./lib/libartn-qe.a
```

```
make lmplib
```

```
( cd src; make; cd - )
make[1]: Entering directory '/home/ajay/Programmes/artn-plugin/src'
make[1]: Leaving directory '/home/ajay/Programmes/artn-plugin/src'
/home/ajay/Programmes/artn-plugin
ln -sf ../src/libartn.a ./lib/libartn.a
ln -sf ../src/libartn.so ./lib/libartn.so
ln -sf ../src/libartn.a ./lib/libartn-qe.a
( cd Files_LAMMPS; make lmplib; cd - )
make[1]: Entering directory '/home/ajay/Programmes/artn-plugin/Files_LAMMPS'

>>>> environment_variable verification
<<<< OK

cxx is: mpicxx

>>>> Shared library build...
mpicxx -shared -rdynamic -o  ./libartn-lmp.so  ../src/Obj/*.o  ./*.o  /usr/lib/x86_64-linux-gnu/libgfortran.so.5 /usr/lib/x86_64-linux-gnu/libopenblas.a  -lpthread

>>>> Shared library link...
ln -sf ../Files_LAMMPS/libartn-lmp.so ../lib/libartn-lmp.so

>>>> Shared library libartn-lmp.so is in lib/

 1) In LAMMPS Package PLUGIN must be loaded
 2) LAMMPS must be compiled with mode=shared
 3) To launch LAMMPS, lammps/src path should be loaded in LD_LIBRARY_PATH
 4) Enjoy ;)

make[1]: Leaving directory '/home/ajay/Programmes/artn-plugin/Files_LAMMPS'
/home/ajay/Programmes/artn-plugin
```

```
make patch-
     qe
```

```
        rism1d.o ../src/libpw.a ../../KS_Solvers/libks_solvers.a ../../dft-d3/libdftd3qe.a /home/ajay/Programmes/qe-7.3//Modules/libqemod.a /home/ajay/Programmes/qe-7.
3//upflib/libupf.a /home/ajay/Programmes/qe-7.3//XClib/xc_lib.a /home/ajay/Programmes/qe-7.3//FFTXlib/src/libqefft.a /home/ajay/Programmes/qe-7.3//LAXlib/libqela.a /ho
me/ajay/Programmes/qe-7.3//UtilXlib/libutil.a /home/ajay/Programmes/qe-7.3//MBD/libmbd.a /home/ajay/Programmes/artn-plugin/src/libartn.a /home/ajay/Programmes/artn-plu
gin/src/libartn.so /home/ajay/Programmes/artn-plugin/src/libartn.so /home/ajay/Programmes/artn-plugin/src/libartn.so /home/ajay/Programmes/artn-plugin/src/libartn.so /
home/ajay/Programmes/artn-plugin/src/libartn.so /home/ajay/Programmes/artn-plugin/src/libartn.so /home/ajay/Programmes/artn-plugin/src/libartn.so /home/ajay/Programmes
/artn-plugin/src/libartn.so /home/ajay/Programmes/artn-plugin/src/libartn.so  -L/home/ajay/Programmes/qe-7.3//external/devxlib/src -ldevXlib  -L/home/ajay/Programmes/m
ylammps/src/  -lopenblas   -lfftw3  -lopenblas      /home/ajay/Programmes/artn-plugin/src/libartn.a /home/ajay/Programmes/artn-plugin/src/libartn.so /home/ajay/Progra
mmes/artn-plugin/src/libartn.so /home/ajay/Programmes/artn-plugin/src/libartn.so /home/ajay/Programmes/artn-plugin/src/libartn.so /home/ajay/Programmes/artn-plugin/src
/libartn.so /home/ajay/Programmes/artn-plugin/src/libartn.so /home/ajay/Programmes/artn-plugin/src/libartn.so /home/ajay/Programmes/artn-plugin/src/libartn.so /home/aj
ay/Programmes/artn-plugin/src/libartn.so
( cd ../../bin ; ln -fs ../PW/tools/rism1d.x . )
make[3]: Leaving directory '/home/ajay/Programmes/qe-7.3/PW/tools'
make[2]: Leaving directory '/home/ajay/Programmes/qe-7.3/PW'
make[1]: Leaving directory '/home/ajay/Programmes/qe-7.3'
/home/ajay/Programmes/artn-plugin
```

Table of Contents

Many examples for QE and LAMMPS

```
cd examples
ll
```

```
ajay@lillou:~/Programmes/artn-plugin$ cd examples/
ajay@lillou:~/Programmes/artn-plugin/examples$ ll
total 84
drwxr-xr-x 3 ajay m3 4096 Aug  1 12:37 Al-vac-EAM.LAMMPS.d/
drwxr-xr-x 4 ajay m3 4096 Jul 31 11:36 Al-vacancy.QE.d/
drwxr-xr-x 3 ajay m3 4096 Jul  4 09:14 Alad.Al100.QE.d/
drwxr-xr-x 5 ajay m3 4096 Jul  4 09:14 COUPLE/
drwxr-xr-x 3 ajay m3 4096 Jul  4 09:14 ClCH3+Cl.QE.d/
drwxr-xr-x 3 ajay m3 4096 Jul 31 11:21 Evolution_a-Si.LAMMPS.d/
drwxr-xr-x 3 ajay m3 4096 Jul 30 15:59 LJ.SaddleRefine.LAMMPS.d/
drwxr-xr-x 3 ajay m3 4096 Jul  4 09:14 Li-migration.QE.d/
drwxr-xr-x 2 ajay m3 4096 Jul 24 17:08 Monkey_SP.d/
drwxr-xr-x 4 ajay m3 4096 Jul 30 16:02 NH3.QE.d/
drwxr-xr-x 2 ajay m3 4096 Jul  4 09:14 OptBench_SaddleSearch.d/
drwxr-xr-x 6 ajay m3 4096 Jul 30 16:04 Oxydation.ReaxFF.LAMMPS.d/
drwxr-xr-x 3 ajay m3 4096 Jul 30 16:05 Pt111.LAMMPS.d/
-rw-r--r-- 1 ajay m3  503 Jul  4 09:14 README.md
-rw-r--r-- 1 ajay m3 2659 Jul  4 09:14 README.rst
drwxr-xr-x 2 ajay m3 4096 Jul  4 09:14 Si-vac.LAMMPS.d/
drwxr-xr-x 3 ajay m3 4096 Jul  4 09:14 Si-vac.QE.d/
drwxr-xr-x 2 ajay m3 4096 Jul  4 09:14 Si-vac.Siesta.d/
drwxr-xr-x 2 ajay m3 4096 Jul  4 09:14 a-Si.LAMMPS.d/
drwxr-xr-x 2 ajay m3 4096 Jul  4 09:14 a-Si.LAMMPS_Annealing_bash.d/
drwxr-xr-x 3 ajay m3 4096 Jul  4 09:14 graphene.QE.d/
```

Input files

```
cd Al-vac-EAM.LAMMPS.d/
ll
```

```
ajay@lillou:~/Programmes/artn-plugin/examples$ cd Al-vac-EAM.LAMMPS.d/
ajay@lillou:~/Programmes/artn-plugin/examples/Al-vac-EAM.LAMMPS.d$ ll
total 196
-rw-r--r-- 1 ajay m3 170699 Jul  4 09:14 AlO.eam.alloy
-rw-r--r-- 1 ajay m3   7686 Jul  4 09:14 Al_vac.data
-rw-r--r-- 1 ajay m3   1053 Jul  4 09:14 README.md
-rw-r--r-- 1 ajay m3    643 Jul  8 15:02 artn.in
-rw-r--r-- 1 ajay m3   1863 Jul  4 09:14 lammps.in
drwxr-xr-x 2 ajay m3   4096 Jul  4 09:14 reference.d/
-rwxr-xr-x 1 ajay m3    154 Jul  4 09:14 run_example.sh*
```

- The empirical potential: **AlO.eam.alloy**
- The input structure: **Al_vac.data**
- The lammps commands: **lammps.in**
- The artn parameters: **artn.in**
- The execution command: run_example.sh
- A directory with the attemped results: reference.d/
- An explanation of the example: README.md

Ouput files

```
./run_example
ls -ltr
```

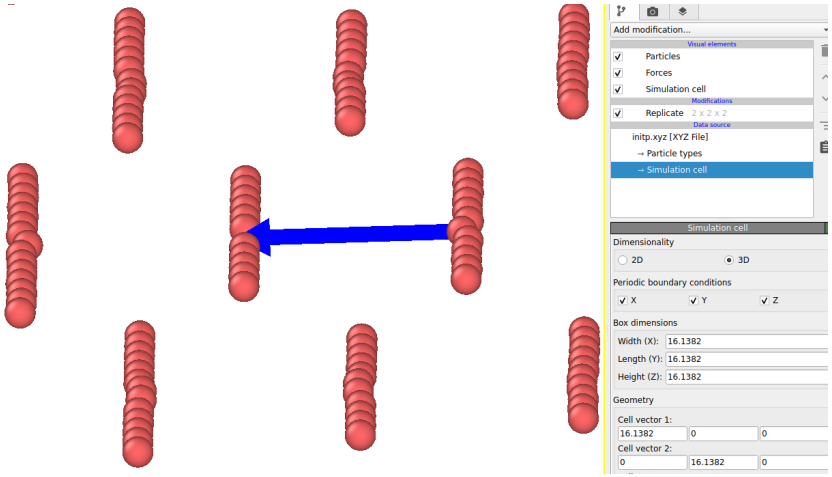
```
ajay@lillou:~/Programmes/artn-plugin/examples/Al-vac-EAM.LAMMPS.d$ ls -ltr
total 4132
-rwxr-xr-x 1 ajay m3     154 Jul  4 09:14 run_example.sh
drwxr-xr-x 2 ajay m3    4096 Jul  4 09:14 reference.d
-rw-r--r-- 1 ajay m3    1863 Jul  4 09:14 lammps.in
-rw-r--r-- 1 ajay m3    1053 Jul  4 09:14 README.md
-rw-r--r-- 1 ajay m3    7686 Jul  4 09:14 Al_vac.data
-rw-r--r-- 1 ajay m3  170699 Jul  4 09:14 AlO.eam.alloy
-rw-r--r-- 1 ajay m3     643 Jul  8 15:02 artn.in
-rw-r--r-- 1 ajay m3   22208 Aug  1 15:41 initp.xyz
-rw-r--r-- 1 ajay m3   22295 Aug  1 15:41 latest_eigenvec.xyz
-rw-r--r-- 1 ajay m3      16 Aug  1 15:41 sadcounter
-rw-r--r-- 1 ajay m3   22219 Aug  1 15:41 sad1.xyz
-rw-r--r-- 1 ajay m3   22223 Aug  1 15:41 min1.xyz
-rw-r--r-- 1 ajay m3      16 Aug  1 15:41 mincounter
-rw-r--r-- 1 ajay m3   22219 Aug  1 15:41 min2.xyz
-rw-r--r-- 1 ajay m3 3871129 Aug  1 15:41 config.dmp
-rw-r--r-- 1 ajay m3   10107 Aug  1 15:41 log.lammps
-rw-r--r-- 1 ajay m3    9752 Aug  1 15:41 artn.out
```

- The initial structure and displacement: **initp.xyz**
- The latest eigenvector: **latest_eigenvec.xyz**
- The target structures: **sad1.xyz**, **min1.xyz**, **min2.xyz**
- Their counters: **sadcounter** and **mincounter**
- The intermediate structures from dump: **config.dmp**
- The LAMMPS information : **log.lammps**
- The information about the path to SP: **artn.out**

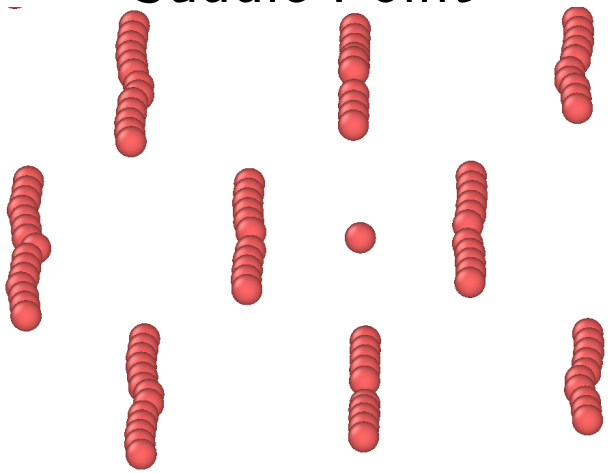
Visualization

```
ovito initp.xyz sad1.xyz min*.xyz
```

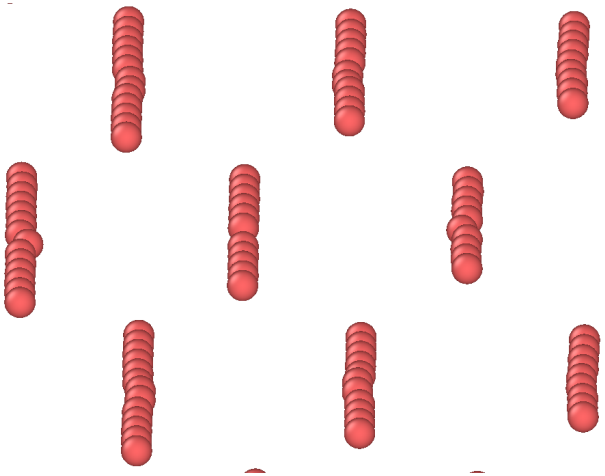
Initial displ



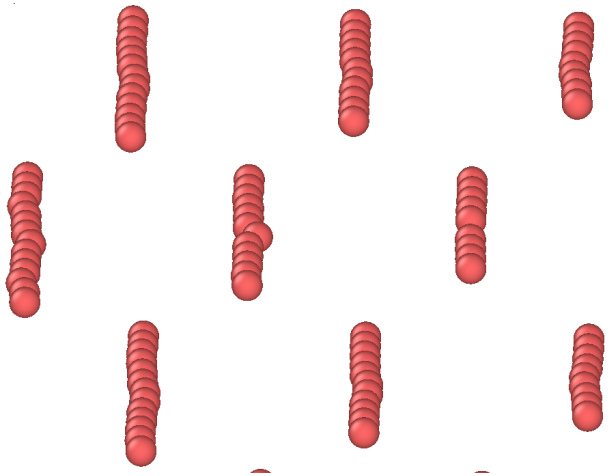
Saddle Point



Minimum 1



Minimum 2



Basic parameters, see documentation

```
vi artn.in
```

- **engine_units** : mandatory
- **verbose** : 0/1/2/3
- **lpush_final** : don't stop at saddle?
- **ninit** : n steps without lanczos
- **nsmooth** : n smooth steps
- **forc_thr** : definition of saddle point
- **push_step_size** : size of the random push
- **push_mode** : list/rad/all
- **push_ids** : indices of the pushed atoms
- **push_add_const(:,251)** : random cone = vector+angle
- **lanczos_disp** : size for derivative
- **lanczos_max_size** : max size lanczos matrix
- **eigen_step_size** : max push size along eigenvector
- **push_over** : Once at SP, scale of the eigen_step_size
- **zseed** : random seed

Follow the path to SP

header

```
-------------------------------------------------------
|     ____          /\    ______________              |
|    / _  |        /  \  |  __ \__   __|             |
|   | (_) | ___   / /\ \ | |__) ||  |                 |
|   |  __/ |___| / /  \ \|  _  / |  |                 |
|   | |         / ______ \ | \ \ |  |                 |
|   |_|        /_/      \_\_|  \_\|__|                |
|                                        ARTn plugin |
-------------------------------------------------------
Launched on (dd.mm.yyyy): 1.8.2024 at: 15:41:41

               INPUT PARAMETERS
-------------------------------------------------------
engine_units: lammps/metal
Verbosity Level:  2
Zseed          : 207981888
-------------------------------------------------------
Simulation Parameters:
-------------------------------------------------------
        * Iterators Parameter:
          ninit            = 3
          nevalf_max       = 9999
          nperp_limitation = 4 8 12 16 -1
          neigen           = 1
          nsmooth          = 2
          nnewchance       = 0
        * Threshold Parameter:
          converge_property = maxval
          forc_thr          =   0.010  eV/Ang
          eigval_thr        =  -0.486  eV/Ang**2
          eigval_thr_nounit =  -0.010
          delr_thr          =   0.100  Ang
        * Step size Parameter:
          push_step_size  =   0.30  Ang
          eigen_step_size =   0.20  Ang
          push_over       =   1.00  fraction of eigen_step_size
          push_mode       =  list
          alpha_mix_cr    =   0.20
          push_ids      = 251
          eigenvec_guess  = BBBB
-------------------------------------------------------
Lanczos algorithm:
-------------------------------------------------------
          lanczos_min_size      =      3
          lanczos_max_size      =     16
          lanczos_disp          =   0.020  Ang
          lanczos_eval_conv_thr =   0.010
-------------------------------------------------------
In/out file preferences:
-------------------------------------------------------
          filin              = artn.in
          filout             = artn.out
          struc_format_out   = xyz
          prefix_sad         = sad
          prefix_min         = min
-------------------------------------------------------
```

```
vi artn.out
ovito config.dmp
```

path details

```
|> ARTn research : 0 0
--------------------------------------------------
istep    ART_step    Etot     init/eign/perp/lanc/relx     Ftot       Fperp       Fpara      eigval      delr  npart evalf  a1
                     [eV]                              -----------[eV/Ang]-----------  [eV/Ang**2]  [Ang]
|> First PUSH vectors almost RANDOM
|> First EIGEN vectors RANDOM
   0   Bstep void    0.0000     0    0    0    0    0     0.0003     0.0003     0.0000   Infinity    0.0000     0     1   0.00
   1   Bstep/init    0.1103     1    0    4    0    0     0.5123     0.1961     0.4891   Infinity    0.3000     1     6   0.00
   2   Bstep/init    0.3563     2    0    4    0    0     0.7810     0.4559     0.7122   Infinity    0.6000     1    11   0.00
   3   Bstep/init    0.6275     3    0    4    0    0     0.7900     0.7275     0.6465   Infinity    0.9001     1    16   0.00
   4   Sstep/smth    0.6957     3    1    4   12    0     0.7764     0.8024     0.3276    -1.5141    1.0380     1    33   0.01
   5   Sstep/smth    0.6829     3    1    8    4    0     0.7127     0.7225     0.0719    -1.8321    1.1512     1    46   1.00
   6   Sstep/eign    0.6200     3    1   12    4    0     0.5524     0.5533     0.0125    -1.9052    1.2770     1    63   1.00
   7   Sstep/eign    0.5133     3    1   16    4    0     0.2917     0.2920     0.0109    -1.7901    1.3454     5    84   1.00
   8   Sstep/eign    0.3759     3    1   35    3    0     0.0171     0.0112     0.0140    -1.3111    1.5311    10   123   1.00
   9   Sstep/eign    0.3751     3    1   14    5    0     0.0099     0.0099     0.0011    -0.6933    1.5570    10   143   0.99
--------------------------------------------------
|> ARTn found a potential saddle point | E_saddle - E_initial =     0.37508 eV
|> Stored in Configuration Files: * Start: initp.xyz | sad1.xyz
--------------------------------------------------
|> DEBRIEF(SADDLE) | dE = 0.37508 eV | F_{tot,parap,perp} = 0.98796E-2  0.10653E-2  0.98783E-2  eV/Ang | EigenVal = -0.69331 eV/
--------------------------------------------------
|> Pushing forward to a minimum  ***
--------------------------------------------------
|> number of steps: 142
   9   Rstep relx    0.1374     3    1    0    0    5     0.3381     0.1823     0.3626    -0.6933    0.0000     0   149   0.99
  ...
   9   Rstep/relx    0.0028     3    1    0    0   41     0.0098     0.0086     0.0082    -0.6933    2.7442     1   185   0.99
     |> Stop relax because force < forc_thr       :     0.0098 <     0.0100              maxval
--------------------------------------------------
|> ARTn converged to a forward minimum | backward E_act =     0.37223 eV
--------------------------------------------------
|> number of steps: 184
|> DEBRIEF(RLX :1) | dE = 0.28443E-2 eV | F_{tot,parap,perp} = 0.98109E-2  0.81584E-2  0.86394E-2  eV/Ang | EigenVal = -0.69331
--------------------------------------------------
   9   Sstep relx    0.3759     3    1    0    0    0     0.0109     0.0109     0.0006    -0.6933    0.0000     0   186   0.99
  ...
   9   Rstep/relx    0.0094     3    1    0    0   62     0.0095     0.0095     0.0062    -0.6933    0.3096     0   249   0.99
     |> Stop relax because force < forc_thr       :     0.0095 <     0.0100              maxval
--------------------------------------------------
|> ARTn converged to a backward minimum::
--------------------------------------------------
        forward  E_act =     0.36565 eV
        backward E_act =     0.37223 eV
        reaction dE    =     0.00658 eV
        dEinit - dEfinal     =    -0.00943 eV
--------------------------------------------------
|> Configuration Files: * Start: initp.xyz | sad1.xyz | min1.xyz | min2.xyz
--------------------------------------------------
|> DEBRIEF(RLX :2) | dE = 0.94282E-2 eV | F_{tot,parap,perp} = 0.94954E-2  0.61824E-2  0.94999E-2  eV/Ang | EigenVal = -0.69331
--------------------------------------------------
ifail:  0 * Start: initp.xyz | sad1.xyz | min1.xyz | min2.xyz
|> BLOCK FINALIZE..
|> number of steps: 248
|> Enter in ARTn but already finished, RETURN
|> BLOCK FINALIZE..
|> number of steps: 249
!> CLEANING ARTn | Fail: 0
```

See only the important information

```
grep DEBRIEF artn.out
```

```
illou:~/Programmes/artn-plugin/examples/Al-vac-EAM.LAMMPS.d$ grep DEBRIEF artn.out
 |> DEBRIEF(SADDLE) | dE = 0.37518 eV    | F_{tot,parap,perp} = 0.99874E-2  0.11294E-2  0.99855E-2  eV/Ang | EigenVal = -0.69709 eV/Ang**2 | npart = 10 | delr = 1.5572 Ang  | evalf = 143
 |> DEBRIEF(RLX :1) | dE = 0.77427E-2 eV | F_{tot,parap,perp} = 0.84573E-2  0.15900E-2  0.84081E-2  eV/Ang | EigenVal = -0.69709 eV/Ang**2 | npart = 3  | delr = 2.7176 Ang  | evalf = 189
 |> DEBRIEF(RLX :2) | dE = 0.15075E-2 eV | F_{tot,parap,perp} = 0.87244E-2  0.17609E-2  0.87269E-2  eV/Ang | EigenVal = -0.69709 eV/Ang**2 | npart = 0  | delr = 0.11916 Ang | evalf = 235
```

```
grep "␣␣.step" artn.out
```

```
illou:~/Programmes/artn-plugin/examples/Al-vac-EAM.LAMMPS.d$ grep "  .step" artn.out
istep    ART_step    Etot     init/eign/perp/lanc/relx     Ftot      Fperp      Fpara      eigval      delr  npart evalf  a1
   0   Bstep void   0.0000     0    0    0    0    0     0.0003     0.0003     0.0001   Infinity    0.0000      0     1    0.00
   1   Bstep/init   0.1128     1    0    4    0    0     0.6282     0.2317     0.5664   Infinity    0.3000      1     6    0.00
   2   Bstep/init   0.3743     2    0    4    0    0     1.0419     0.5719     0.8651   Infinity    0.6000      1    11    0.00
   3   Bstep/init   0.6830     3    0    4    0    0     1.2345     0.9768     0.8737   Infinity    0.9001      1    16    0.00
   4   Sstep/smth   0.7587     3    1    4   12    0     1.1371     1.0947     0.4266    -1.3122    1.0357      1    33    0.01
   5   Sstep/smth   0.7319     3    1    8    4    0     0.9442     0.9584     0.0835    -1.8785    1.1490      1    46    0.99
   6   Sstep/eign   0.6444     3    1   12    4    0     0.6739     0.6755     0.0180    -1.9146    1.2770      3    63    1.00
   7   Sstep/eign   0.5197     3    1   16    4    0     0.2891     0.2897     0.0153    -1.7714    1.3507      3    84    1.00
   8   Sstep/eign   0.3765     3    1   35    3    0     0.0198     0.0114     0.0122    -1.4330    1.5339     10   123    1.00
   9   Sstep/eign   0.3754     3    1   15    5    0     0.0099     0.0099     0.0007    -0.6967    1.5569     10   144    0.99
   9   Rstep relx   0.1696     3    1    0    0    5     0.3438     0.2650     0.3207    -0.6967    0.0000      0   150    0.99
   9   Sstep relx   0.3763     3    1    0    0    0     0.0129     0.0129     0.0014    -0.6967    0.0000      0   188    0.99
```

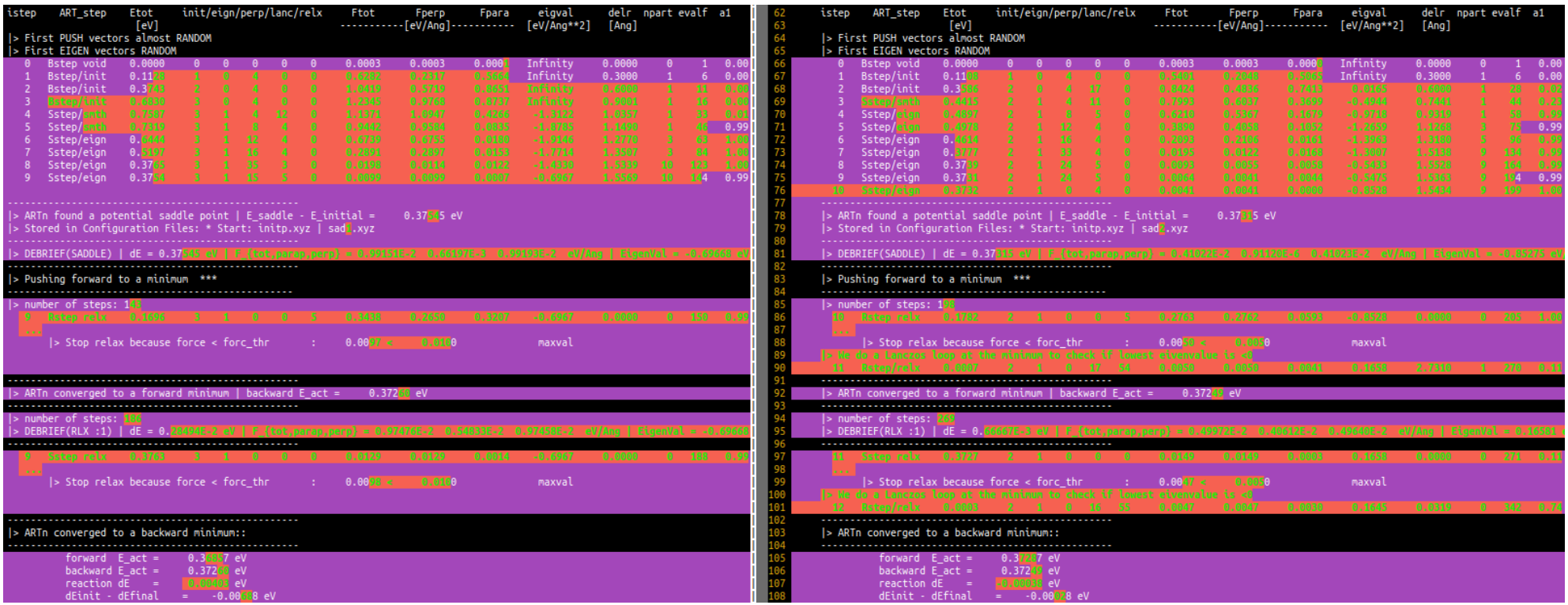
Modify some parameters

```
mv artn.out artn.out_save
vi artn.in
```

- forc_thr = 0.005
- ninit = 1
- nsmooth = 1
- lanczos_at_min =.true.

run and see the difference

```
mpirun -mp 4 ../../../mylammps/src/lmp_mpi
vimdiff artn.out artn.out_save
```



Input files

```
cd ../Al-vacancy-QE.d/
ll
```

```
ajay@lillou:~/Programmes/artn-plugin/examples/Al-vacancy.QE.d$ ll
total 48
-rw-r--r-- 1 ajay m3 26928 Jul  4 09:14 Al.pz-vbc.UPF
-rw-r--r-- 1 ajay m3   985 Jul  4 09:14 README.md
-rw-r--r-- 1 ajay m3   465 Jul 31 11:36 artn.in
drwxr-xr-x 2 ajay m3  4096 Jul  4 09:14 reference.d/
-rw-r--r-- 1 ajay m3  2872 Jul  4 09:14 relax.Al-vacancy.in
-rwxr-xr-x 1 ajay m3   190 Jul  4 09:14 run_example.sh*
ajay@lillou:~/Programmes/artn-plugin/examples/Al-vacancy.QE.d$
```

- The pseudopotential: **Al.pz-vbc.UPF**
- The QE commands: **relax.Al-vacancy.in**
- The artn parameters: **artn.in**
- The execution command: run_example.sh
- A directory with the attemped results: reference.d/
- An explanation of the example: README.md

Ouput files

```
./run_example (30s on 4 cores)
ls -ltr
```

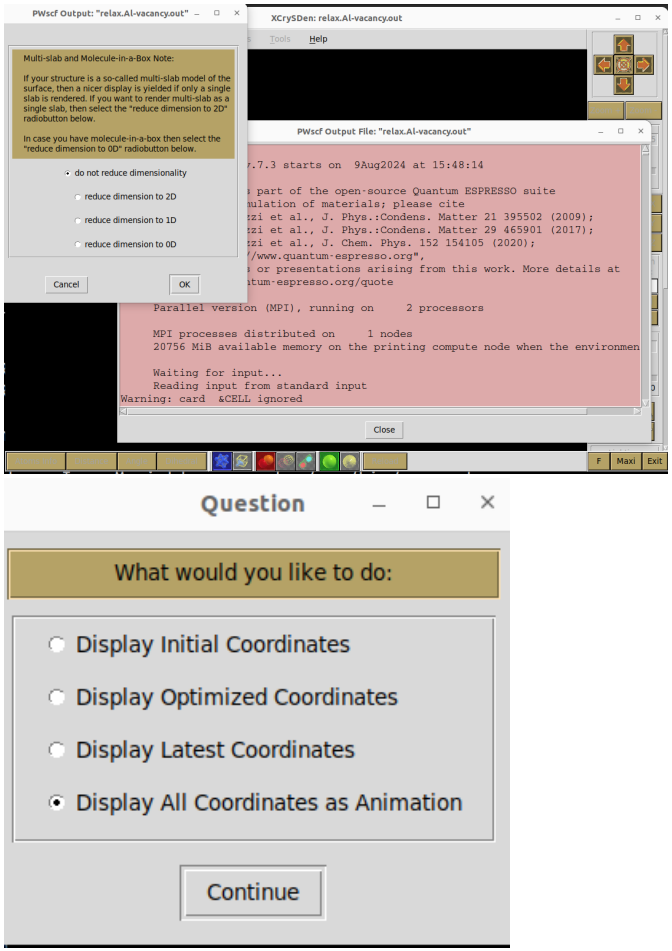
```
ajay@lillou:~/Programmes/artn-plugin/examples/Al-vacancy.QE.d$ ls -ltr
total 2084
-rw-r--r-- 1 ajay m3     985 Jul  4 09:14 README.md
-rw-r--r-- 1 ajay m3   26928 Jul  4 09:14 Al.pz-vbc.UPF
-rwxr-xr-x 1 ajay m3     190 Jul  4 09:14 run_example.sh
-rw-r--r-- 1 ajay m3    2872 Jul  4 09:14 relax.Al-vacancy.in
drwxr-xr-x 2 ajay m3    4096 Jul  4 09:14 reference.d
-rw-r--r-- 1 ajay m3    3199 Aug  9 15:48 initp.xsf
-rw-r--r-- 1 ajay m3    3199 Aug  9 15:48 latest_eigenvec.xsf
-rw-r--r-- 1 ajay m3      16 Aug  9 15:48 sadcounter
-rw-r--r-- 1 ajay m3    3199 Aug  9 15:48 sad1.xsf
-rw-r--r-- 1 ajay m3    3199 Aug  9 15:48 min1.xsf
-rw-r--r-- 1 ajay m3     104 Aug  9 15:48 pwscf.fire
-rw-r--r-- 1 ajay m3      16 Aug  9 15:48 mincounter
-rw-r--r-- 1 ajay m3    3199 Aug  9 15:48 min2.xsf
-rw-r--r-- 1 ajay m3   25988 Aug  9 15:48 artn.out
-rw-r--r-- 1 ajay m3  983292 Aug  9 15:48 pwscf.xml
drwxr-xr-x 2 ajay m3    4096 Aug  9 15:48 pwscf.save
-rw-r--r-- 1 ajay m3 1028870 Aug  9 15:48 relax.Al-vacancy.out
-rw-r--r-- 1 ajay m3     465 Aug  9 16:08 artn.in
```

- The initial structure and displacement: **initp.xsf**
- The latest eigenvector: **latest_eigenvec.xsf**
- The target structures: **sad1.xsf**, **min1.xsf**, **min2.xsf**
- Their counters: **sadcounter** and **mincounter**
- The charge density and restart QE files: **pwscf***
- The QE informations : **relax.Al-vacancy.out**
- The informations about the path to SP: **artn.out**

Visualization

```
xcrysden --pwo relax.Al-vacancy.out
```

press Ok and All



Try to recognize:
push-perp_relax-Lanczos
2 final relaxations from SP

Understand artn.out (verbose=3)

```
istep    ART_step    Etot     init/eign/perp/lanc/relx     Ftot       Fperp      Fpara      eigval      delr  npart evalf  a1
                      [Ry]                                ----------[Ry/a.u.]----------  [Ry/a.u.**2]   [a.u.]
|> First PUSH vectors almost RANDOM
|> PUSH Vector is:
|> atom       1 dx= 0.2121 dy= 0.0000 dz= 0.2121
|> First EIGEN vectors RANDOM
   0    Bstep void    0.0000     0    0    0    0    0     0.0001     0.0001     0.0000 **********     0.0000     0     1   0.00
   1    Bstep init    0.0031     1    0    0    0    0     0.0141     0.0044     0.0141 **********     0.0000     0     2   0.00
   1    Bstep init    0.0030     1    0    1    0    0     0.0139     0.0040     0.0139 **********     0.0000     0     3   0.00
   1    Bstep init    0.0029     1    0    2    0    0     0.0134     0.0033     0.0134 **********     0.0000     0     4   0.00
   1    Bstep init    0.0027     1    0    3    0    0     0.0127     0.0020     0.0127 **********     0.0000     0     5   0.00
   1    Bstep init    0.0026     1    0    4    0    0     0.0118     0.0013     0.0118 **********     0.0000     0     6   0.00
   1    Bstep/init    0.0026     1    0    4    0    0     0.0118     0.0013     0.0118 **********     0.3101     1     6   0.00
        |> Stop perp relax because iperp = nperp max : 4 = 4
   2    Bstep init    0.0099     2    0    0    0    0     0.0221     0.0052     0.0221 **********     0.0000     0     7   0.00
   2    Bstep init    0.0098     2    0    1    0    0     0.0219     0.0049     0.0219 **********     0.0000     0     8   0.00
   2    Bstep init    0.0095     2    0    2    0    0     0.0214     0.0042     0.0214 **********     0.0000     0     9   0.00
   2    Bstep init    0.0092     2    0    3    0    0     0.0205     0.0030     0.0205 **********     0.0000     0    10   0.00
   2    Bstep init    0.0090     2    0    4    0    0     0.0195     0.0019     0.0195 **********     0.0000     0    11   0.00
   2    Bstep/init    0.0090     2    0    4    0    0     0.0195     0.0019     0.0195 **********     0.6227     1    11   0.00
        |> Stop perp relax because iperp = nperp max : 4 = 4
   3    Bstep init    0.0188     3    0    0    0    0     0.0263     0.0066     0.0263 **********     0.0000     0    12   0.00
   3    Bstep init    0.0186     3    0    1    0    0     0.0260     0.0062     0.0260 **********     0.0000     0    13   0.00
   3    Bstep init    0.0183     3    0    2    0    0     0.0254     0.0052     0.0254 **********     0.0000     0    14   0.00
   3    Bstep init    0.0179     3    0    3    0    0     0.0244     0.0037     0.0244 **********     0.0000     0    15   0.00
   3    Bstep init    0.0175     3    0    4    0    0     0.0231     0.0025     0.0231 **********     0.0000     0    16   0.00
   3    Bstep/init    0.0175     3    0    4    0    0     0.0231     0.0025     0.0231 **********     0.9398     5    16   0.00
        |> Stop perp relax because iperp = nperp max : 4 = 4
   4    Bstep init    0.0175     3    0    0    1    0     0.0232     0.0025     0.0232 **********     0.0000     0    17   0.00
   4    Bstep init    0.0176     3    0    0    2    0     0.0232     0.0025     0.0232     0.1168     0.0000     0    18   0.00
   4    Bstep init    0.0174     3    0    0    3    0     0.0231     0.0027     0.0231     0.0782     0.0000     0    19   0.00
   4    Bstep init    0.0176     3    0    0    4    0     0.0232     0.0029     0.0230     0.0526     0.0000     0    20   0.00
   4    Bstep init    0.0174     3    0    0    5    0     0.0233     0.0029     0.0230     0.0248     0.0000     0    21   0.00
   4    Bstep init    0.0175     3    0    0    6    0     0.0233     0.0027     0.0231     0.0095     0.0000     0    22   0.00
   4    Bstep init    0.0176     3    0    0    7    0     0.0232     0.0025     0.0232     0.0048     0.0000     0    23   0.00
   4    Bstep init    0.0174     3    0    0    8    0     0.0232     0.0025     0.0231     0.0038     0.0000     0    24   0.00
   4    Bstep init    0.0176     3    0    0    9    0     0.0233     0.0028     0.0231     0.0034     0.0000     0    25   0.00
   4    Bstep init    0.0174     3    0    0   10    0     0.0233     0.0028     0.0231     0.0030     0.0000     0    26   0.00
   4    Bstep init    0.0176     3    0    0   11    0     0.0233     0.0027     0.0231     0.0025     0.0000     0    27   0.00
   4    Bstep init    0.0174     3    0    0   12    0     0.0231     0.0028     0.0230     0.0020     0.0000     0    28   0.00
   4    Bstep init    0.0175     3    0    0   13    0     0.0231     0.0027     0.0231     0.0018     0.0000     0    29   0.00
   4    Bstep init    0.0175     3    0    0   14    0     0.0234     0.0027     0.0232     0.0018     0.0000     0    30   0.00
   4    Bstep init    0.0175     3    0    0   15    0     0.0232     0.0025     0.0232     0.0018     0.0000     0    31   0.03
   4    Bstep init    0.0282     4    0    0    0    0     0.0266     0.0070     0.0266     0.0018     0.0000     0    32   0.03
   4    Bstep init    0.0280     4    0    1    0    0     0.0263     0.0065     0.0263     0.0018     0.0000     0    33   0.03
   4    Bstep init    0.0276     4    0    2    0    0     0.0256     0.0055     0.0256     0.0018     0.0000     0    34   0.03
   4    Bstep init    0.0272     4    0    3    0    0     0.0247     0.0038     0.0247     0.0018     0.0000     0    35   0.03
   4    Bstep init    0.0268     4    0    4    0    0     0.0234     0.0027     0.0234     0.0018     0.0000     0    36   0.03
   4    Bstep/init    0.0268     4    0    4   15    0     0.0234     0.0027     0.0234     0.0018     1.2584     5    36   0.03
        |> Stop perp relax because iperp = nperp max : 4 = 4
   5    Bstep init    0.0267     4    0    0    1    0     0.0234     0.0027     0.0234     0.0018     0.0000     0    37   0.00
   5    Bstep init    0.0267     4    0    0    2    0     0.0234     0.0028     0.0233     0.0002     0.0000     0    38   0.00
   5    Bstep init    0.0267     4    0    0    3    0     0.0235     0.0030     0.0234    -0.0007     0.0000     0    39   0.00
   5    Bstep init    0.0270     4    0    0    4    0     0.0235     0.0027     0.0235    -0.0007     0.0000     0    40   0.00
   5    Bstep init    0.0266     4    0    0    5    0     0.0236     0.0022     0.0236    -0.0008     0.0000     0    41   0.00
   5    Bstep init    0.0268     4    0    0    6    0     0.0236     0.0024     0.0235    -0.0009     0.0000     0    42   0.00
```

Understand artn.out (verbose=3)

```
   5   Bstep init    0.0268     4    0    0    6    0      0.0236      0.0024      0.0235     -0.0009      0.0000     0    42    0.00
   5   Bstep init    0.0268     4    0    0    7    0      0.0236      0.0029      0.0235     -0.0016      0.0000     0    43    0.00
   5   Bstep init    0.0268     4    0    0    8    0      0.0236      0.0028      0.0234     -0.0026      0.0000     0    44    0.00
   5   Bstep init    0.0267     4    0    0    9    0      0.0236      0.0030      0.0234     -0.0028      0.0000     0    45    0.00
   5   Bstep init    0.0268     4    0    0   10    0      0.0236      0.0032      0.0234     -0.0029      0.0000     0    46    0.00
   5   Bstep init    0.0268     4    0    0   11    0      0.0234      0.0027      0.0234     -0.0029      0.0000     0    47    0.48
   5   Bstep init    0.0369     5    0    0    0    0      0.0239      0.0065      0.0239     -0.0029      0.0000     0    48    0.48
   5   Bstep init    0.0367     5    0    1    0    0      0.0236      0.0061      0.0236     -0.0029      0.0000     0    49    0.48
   5   Bstep init    0.0364     5    0    2    0    0      0.0231      0.0050      0.0231     -0.0029      0.0000     0    50    0.48
   5   Bstep init    0.0360     5    0    3    0    0      0.0223      0.0034      0.0223     -0.0029      0.0000     0    51    0.48
   5   Bstep init    0.0356     5    0    4    0    0      0.0213      0.0027      0.0213     -0.0029      0.0000     0    52    0.48
   5   Bstep/init    0.0356     5    0    4   11    0      0.0213      0.0027      0.0213     -0.0029      1.5753     5    52    0.48
       |> Stop perp relax because iperp = nperp max : 4 = 4
   6   Bstep init    0.0359     5    0    0    1    0      0.0212      0.0027      0.0212     -0.0029      0.0000     0    53    0.00
   6   Bstep init    0.0355     5    0    0    2    0      0.0211      0.0031      0.0211     -0.0089      0.0000     0    54    0.00
   6   Bstep init    0.0357     5    0    0    3    0      0.0211      0.0033      0.0211     -0.0105      0.0000     0    55    0.00
   6   Bstep init    0.0356     5    0    0    4    0      0.0213      0.0030      0.0213     -0.0110      0.0000     0    56    0.00
   6   Bstep init    0.0357     5    0    0    5    0      0.0213      0.0025      0.0212     -0.0120      0.0000     0    57    0.00
   6   Bstep init    0.0356     5    0    0    6    0      0.0213      0.0028      0.0213     -0.0127      0.0000     0    58    0.00
   6   Bstep init    0.0356     5    0    0    7    0      0.0213      0.0029      0.0213     -0.0129      0.0000     0    59    0.00
   6   Sstep init    0.0356     5    0    0    8    0      0.0213      0.0027      0.0213     -0.0129      0.0000     0    60    0.88
   6   Sstep eign    0.0435     5    1    0    0    0      0.0177      0.0039      0.0158     -0.0129      0.0000     0    61    0.88
   6   Sstep/eign    0.0435     5    1    0    8    0      0.0177      0.0039      0.0158     -0.0129      1.8618     5    61    0.88
       |> Stop perp relax because fperp < fpara     :      0.0039 <     0.0158              maxval
       |> No perp relax because fperp < fpara       :      0.0039 <     0.0158              maxval
   7   Sstep eign    0.0437     5    1    0    1    0      0.0175      0.0038      0.0157     -0.0129      0.0000     0    62    0.00
   7   Sstep eign    0.0435     5    1    0    2    0      0.0176      0.0036      0.0159     -0.0168      0.0000     0    63    0.00
   7   Sstep eign    0.0436     5    1    0    3    0      0.0176      0.0038      0.0159     -0.0187      0.0000     0    64    0.00
   7   Sstep eign    0.0435     5    0    0    4    0      0.0177      0.0039      0.0159     -0.0188      0.0000     0    65    0.99
   7   Sstep eign    0.0498     5    1    0    0    0      0.0126      0.0031      0.0117     -0.0188      0.0000     0    66    0.99
   7   Sstep/eign    0.0498     5    1    0    4    0      0.0126      0.0031      0.0117     -0.0188      2.1494     5    66    0.99
       |> Stop perp relax because fperp < fpara     :      0.0031 <     0.0117              maxval
       |> No perp relax because fperp < fpara       :      0.0031 <     0.0117              maxval
   8   Sstep eign    0.0499     5    1    0    1    0      0.0125      0.0031      0.0116     -0.0188      0.0000     0    67    0.00
   8   Sstep eign    0.0498     5    1    0    2    0      0.0126      0.0029      0.0119     -0.0192      0.0000     0    68    0.00
   8   Sstep eign    0.0498     5    1    0    3    0      0.0126      0.0032      0.0118     -0.0223      0.0000     0    69    0.00
   8   Sstep eign    0.0498     5    1    0    4    0      0.0127      0.0031      0.0118     -0.0227      0.0000     0    70    0.00
   8   Sstep eign    0.0498     5    0    0    5    0      0.0127      0.0031      0.0118     -0.0227      0.0000     0    71    0.99
   8   Sstep eign    0.0536     5    1    0    0    0      0.0065      0.0027      0.0058     -0.0227      0.0000     0    72    0.99
   8   Sstep/eign    0.0536     5    1    0    5    0      0.0065      0.0027      0.0058     -0.0227      2.4393     5    72    0.99
       |> Stop perp relax because fperp < fpara     :      0.0027 <     0.0058              maxval
       |> No perp relax because fperp < fpara       :      0.0027 <     0.0058              maxval
   9   Sstep eign    0.0537     5    1    0    1    0      0.0063      0.0027      0.0055     -0.0227      0.0000     0    73    0.00
   9   Sstep eign    0.0536     5    1    0    2    0      0.0065      0.0028      0.0058     -0.0309      0.0000     0    74    0.00
   9   Sstep eign    0.0536     5    1    0    3    0      0.0065      0.0033      0.0058     -0.0333      0.0000     0    75    0.00
   9   Sstep eign    0.0536     5    0    0    4    0      0.0065      0.0027      0.0058     -0.0333      0.0000     0    76    0.99
   9   Sstep eign    0.0547     5    1    0    0    0      0.0031      0.0031      0.0005     -0.0333      0.0000     0    77    0.99
   9   Sstep eign    0.0547     5    1    1    0    0      0.0028      0.0028      0.0005     -0.0333      0.0000     0    78    0.99
   9   Sstep eign    0.0546     5    1    2    0    0      0.0021      0.0021      0.0005     -0.0333      0.0000     0    79    0.99
   9   Sstep eign    0.0544     5    1    3    0    0      0.0013      0.0013      0.0005     -0.0333      0.0000     0    80    0.99
   9   Sstep eign    0.0543     5    1    4    0    0      0.0011      0.0011      0.0005     -0.0333      0.0000     0    81    0.99
   9   Sstep eign    0.0543     5    1    5    0    0      0.0010      0.0010      0.0006     -0.0333      0.0000     0    82    0.99
   9   Sstep/eign    0.0543     5    1    5    4    0      0.0010      0.0010      0.0006     -0.0333      2.6838     7    82    0.99
 ---------------------------------------------
 |> ARTn found a potential saddle point | E_saddle - E_initial =     0.05427 Ry
 |> Stored in Configuration Files: * Start: initp.xsf | sad1.xsf
```

Table of Contents

Conclusions

- You want to refine a saddle point?
  $\rightarrow$ Use ARTn
- You have no money (= CPU time)?
  $\rightarrow$ Use ARTn
- You want to explore the PES?
  $\rightarrow$ Use ARTn
- No arbitrary convergence
  $\rightarrow$ can reach any forces
- Use the DFT coupling, Quantum Espresso now
  $\rightarrow$ Plugin $\forall$ softwares in development (VASP, Abinit$\cdots$)
- Empirical potentials: good for harmonic areas, less good for saddle points
  $\rightarrow$ Perfect for Machine Learning training

Conclusions

- You want to refine a saddle point?
  $\rightarrow$ Use ARTn
- You have no money (= CPU time)?
  $\rightarrow$ Use ARTn
- You want to explore the PES?
  $\rightarrow$ Use ARTn
- No arbitrary convergence
  $\rightarrow$ can reach any forces
- Use the DFT coupling, Quantum Espresso now
  $\rightarrow$ Plugin $\forall$ softwares in development (VASP, Abinit$\cdots$)
- Empirical potentials: good for harmonic areas, less good for saddle points
  $\rightarrow$ Perfect for Machine Learning training

Conclusions

- You want to refine a saddle point?
  $\rightarrow$ Use ARTn
- You have no money (= CPU time)?
  $\rightarrow$ Use ARTn
- You want to explore the PES?
  $\rightarrow$ Use ARTn
- No arbitrary convergence
  $\rightarrow$ can reach any forces
- Use the DFT coupling, Quantum Espresso now
  $\rightarrow$ Plugin $\forall$ softwares in development (VASP, Abinit$\cdots$)
- Empirical potentials: good for harmonic areas, less good for saddle points
  $\rightarrow$ Perfect for Machine Learning training

Conclusions

- You want to refine a saddle point?
  $\rightarrow$ Use ARTn
- You have no money (= CPU time)?
  $\rightarrow$ Use ARTn
- You want to explore the PES?
  $\rightarrow$ Use ARTn
- No arbitrary convergence
  $\rightarrow$ can reach any forces
- Use the DFT coupling, Quantum Espresso now
  $\rightarrow$ Plugin $\forall$ softwares in development (VASP, Abinit$\cdots$)
- Empirical potentials: good for harmonic areas, less good for saddle points
  $\rightarrow$ Perfect for Machine Learning training

Conclusions

- You want to refine a saddle point?
  $\rightarrow$ Use ARTn
- You have no money (= CPU time)?
  $\rightarrow$ Use ARTn
- You want to explore the PES?
  $\rightarrow$ Use ARTn
- No arbitrary convergence
  $\rightarrow$ can reach any forces
- Use the DFT coupling, Quantum Espresso now
  $\rightarrow$ Plugin $\forall$ softwares in development (VASP, Abinit$\cdots$)
- Empirical potentials: good for harmonic areas, less good for saddle points
  $\rightarrow$ Perfect for Machine Learning training

Conclusions

- You want to refine a saddle point?
  $\rightarrow$ Use ARTn
- You have no money (= CPU time)?
  $\rightarrow$ Use ARTn
- You want to explore the PES?
  $\rightarrow$ Use ARTn
- No arbitrary convergence
  $\rightarrow$ can reach any forces
- Use the DFT coupling, Quantum Espresso now
  $\rightarrow$ Plugin $\forall$ softwares in development (VASP, Abinit$\cdots$)
- Empirical potentials: good for harmonic areas, less good for saddle points
  $\rightarrow$ Perfect for Machine Learning training

When Not using ARTn

PES badly defined:

- Not smooth empirical potentials
- Many small minima→ increase Lanczos *dr* or refine SP
- Flat energy surface: dissociation
- Too small molecules: Lanczos is useless

When Not using ARTn

PES badly defined:

- Not smooth empirical potentials
- Many small minima$\rightarrow$ increase Lanczos *dr* or refine SP
- Flat energy surface: dissociation
- Too small molecules: Lanczos is useless

When Not using ARTn

PES badly defined:

- Not smooth empirical potentials
- Many small minima$\rightarrow$ increase Lanczos *dr* or refine SP
- Flat energy surface: dissociation
- Too small molecules: Lanczos is useless

When Not using ARTn

PES badly defined:

- Not smooth empirical potentials
- Many small minima$\rightarrow$ increase Lanczos *dr* or refine SP
- Flat energy surface: dissociation
- Too small molecules: Lanczos is useless

Questions time

Canada

Italy

France

Jay *et al.*, J. Chem. Theory Comput. 16, 2020

Jay *et al.*, Comp. Mat. Sc. 209, 2022

Poberznik *et al.*, Comp. Phys. Comm. 295, 2024

Gunde *et al.*, J. Chem. Phys. 23, 2024

Thank you for your attention

questions suggestions

What is your favorite PES?

Smooth

DFT
slow explo, accurate

Unsmooth

Empirical potentials
fast explo, not accurate