

# First experiences on porting Quantum Chemistry codes to the SX-Aurora



2<sup>nd</sup> Aurora Deep Dive Workshop - RWTH Aachen

Markus Oppel – University of Vienna –  
Institute of Theoretical Chemistry

---

November 28<sup>th</sup>, 2019

# Who we are and what we do

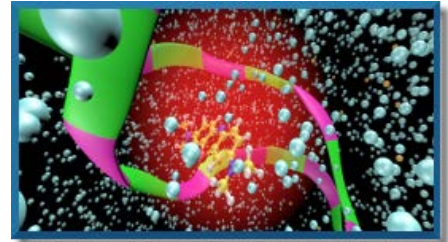


# Who we are and what we do

- **Theoretical Chemistry group at the University of Vienna - Head: Prof. Leticia González**
- **Research topic: Dynamics of excited molecules - Photophysics and photochemistry**
- **Photostability of DNA building blocks**
- **Drugs interacting with biological residues**
- **Photocatalytically materials for solar energy (artificial photosynthesis)**

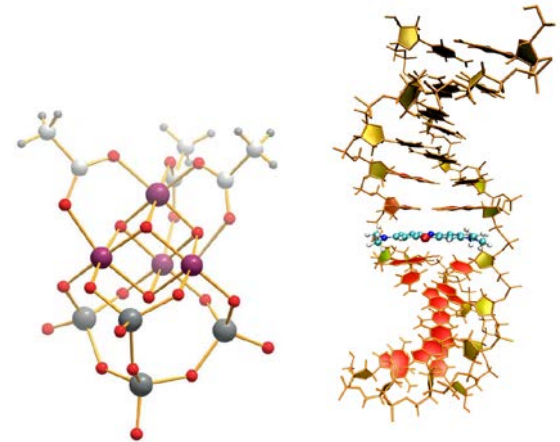
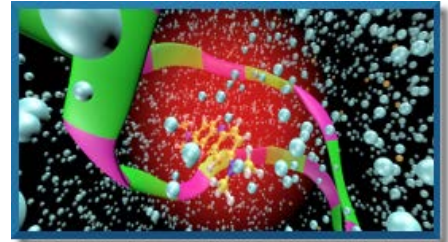
# Who we are and what we do

- **Theoretical Chemistry group at the University of Vienna - Head: Prof. Leticia González**
- **Research topic: Dynamics of excited molecules - Photophysics and photochemistry**
- **Photostability of DNA building blocks**
- **Drugs interacting with biological residues**
- **Photocatalytically materials for solar energy (artificial photosynthesis)**



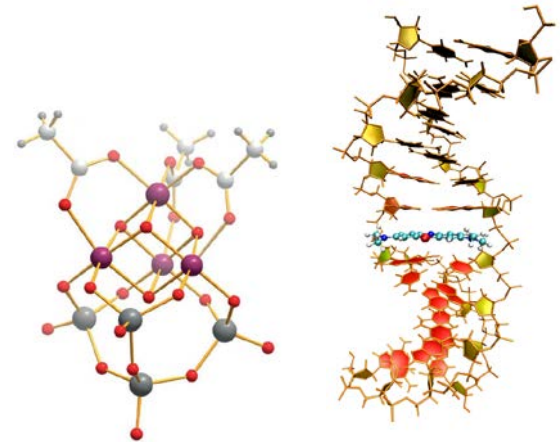
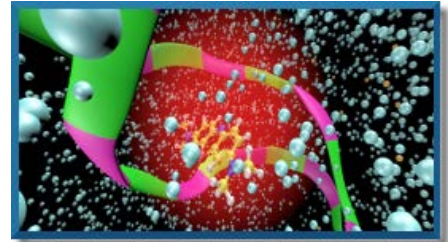
# Who we are and what we do

- **Theoretical Chemistry group at the University of Vienna - Head: Prof. Leticia González**
- **Research topic: Dynamics of excited molecules - Photophysics and photochemistry**
- **Photostability of DNA building blocks**
- **Drugs interacting with biological residues**
- **Photocatalytically materials for solar energy (artificial photosynthesis)**



# Who we are and what we do

- **Theoretical Chemistry group at the University of Vienna - Head: Prof. Leticia González**
- **Research topic: Dynamics of excited molecules - Photophysics and photochemistry**
- **Photostability of DNA building blocks**
- **Drugs interacting with biological residues**
- **Photocatalytically materials for solar energy (artificial photosynthesis)**



# How we do it

# How we do it

- Molecular dynamics simulations based on *ab initio* potential energy surfaces: “ab initio MD”
- Software package “SHARC - **S**urface **H**opping including **A**rbitrary **C**ouplings” - simulate chemical dynamics including any kind of couplings [1]

<https://github.com/sharc-md>

- Interface to Quantum Chemistry programs

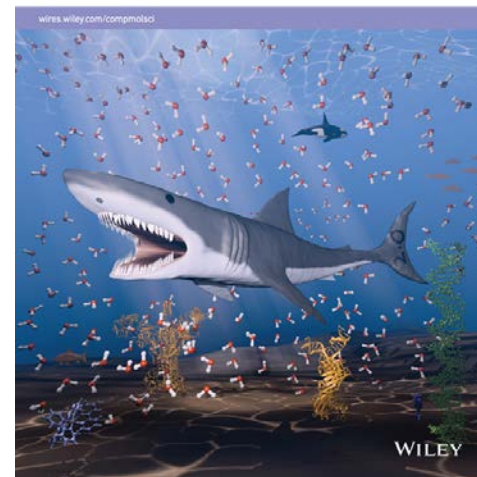


# How we do it

- Molecular dynamics simulations based on *ab initio* potential energy surfaces: “ab initio MD”
- Software package “SHARC - **S**urface **H**opping including **A**rbitrary **C**ouplings” - simulate chemical dynamics including any kind of couplings [1]

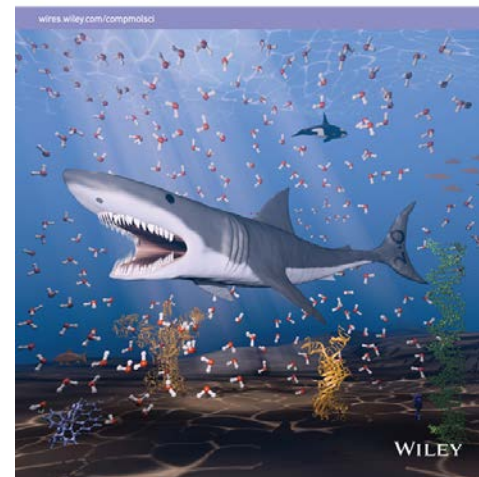
<https://github.com/sharc-md>

- Interface to Quantum Chemistry programs



# How we do it

- Molecular dynamics simulations based on *ab initio* potential energy surfaces: “ab initio MD”
- Software package “SHARC - **S**urface **H**opping including **A**rbitrary **C**ouplings” - simulate chemical dynamics including any kind of couplings [1]

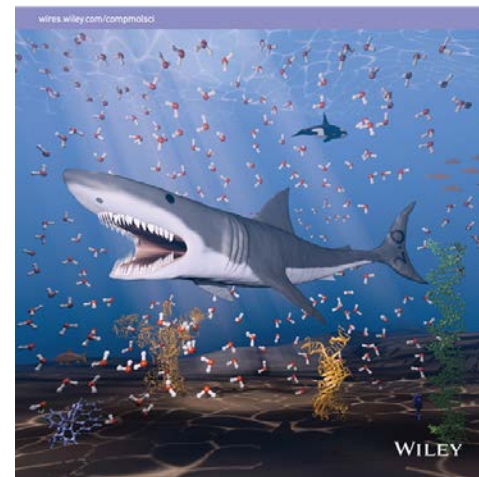


<https://github.com/sharc-md>

- Interface to Quantum Chemistry programs

# How we do it

- Molecular dynamics simulations based on *ab initio* potential energy surfaces: “ab initio MD”
- Software package “SHARC - **S**urface **H**opping including **A**rbitrary **C**ouplings” - simulate chemical dynamics including any kind of couplings [1]



<https://github.com/sharc-md>

- Interface to Quantum Chemistry programs

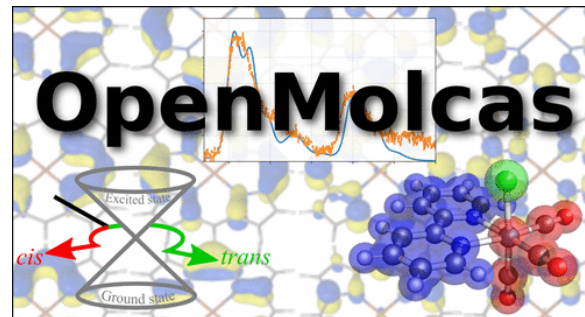
# Quantum chemistry code

# Quantum chemistry code

- **OpenMolcas: Fortran-based open source quantum chemistry code[2]**
- **Source code available on Gitlab (<https://gitlab.com/Molcas/OpenMolcas>)**
- **More than 40 single executables (for each computational (sub-) task – communication via (scratch-)files (Binary/ASCII or HDF5)**
- **Configuration done using cmake**

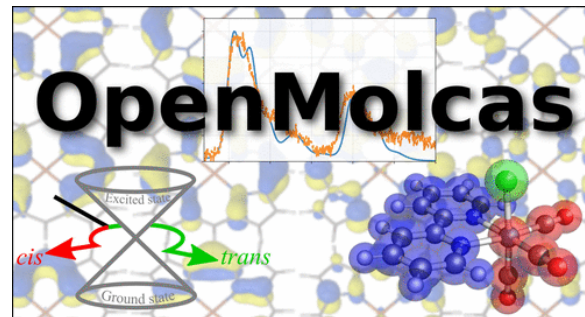
# Quantum chemistry code

- OpenMolcas: Fortran-based open source quantum chemistry code[2]
- Source code available on Gitlab (<https://gitlab.com/Molcas/OpenMolcas>)
- More than 40 single executables (for each computational (sub-) task – communication via (scratch-)files (Binary/ASCII or HDF5))
- Configuration done using cmake



# Quantum chemistry code

- OpenMolcas: Fortran-based open source quantum chemistry code[2]
- Source code available on Gitlab (<https://gitlab.com/Molcas/OpenMolcas>)
- More than 40 single executables (for each computational (sub-) task – communication via (scratch-)files (Binary/ASCII or HDF5))
- Configuration done using cmake



# OpenMolcas - Compilation



# OpenMolcas - Compilation

- Cmake toolchain file available for SX-Aurora  
<https://github.com/SX-Aurora/CMake-toolchain-file>  
– can be included into OpenMolcas' CMakeList.txt file
  
- Activate ncc, nfort and ASL (lapack and blas)

# OpenMolcas - Compilation

- Cmake toolchain file available for SX-Aurora  
<https://github.com/SX-Aurora/CMake-toolchain-file>  
– can be included into OpenMolcas' CMakeList.txt file

```
cmake -DCMAKE_TOOLCHAIN_FILE=<path_to_toolchain_file>` ...
```

- Activate ncc, nfort and ASL (lapack and blas)

# OpenMolcas – Compilation – Part II

## Compilation step successful, if:

- **Adjust the (Fortran) Compile flags: nfort flags NOT compatible with gfortran flags!**
- **Choose the right compiler version**
  - 2.5.0: seg faults (from nfort executable) with a couple of Fortran source files!
  - 2.4.1 ok
  - 2.5.1 also ok :-)

File Edit View Search Terminal Help

```

[oppel@aurora] 227 (OpenMolcas/build_nec):
[oppel@aurora] 227 (OpenMolcas/build_nec): nfort -c -fpp -w -fdefault-integer=8
-02 -I/public/aurora/scratch/oppel/molcas/OpenMolcas/src/Include -I/public/aurora/scratch/oppel/molcas/OpenMolcas/build_nec/CMakeFiles/include -I/public/aurora/scratch/oppel/molcas/OpenMolcas/build_nec/CMakeFiles/mod -I/public/aurora/scratch/oppel/molcas/OpenMolcas/src/alaska_util -I/public/aurora/scratch/oppel/molcas/OpenMolcas/build_nec/CMakeFiles/mod/alaska_util -D_I8_ -D_LINUX_ -D_MOLCAS_ /public/aurora/scratch/oppel/molcas/OpenMolcas/src/alaska_util/cmbnrf1.f
nfort: /opt/nec/ve/nfort/2.5.0/libexec/fcom is abnormally terminated by SIGSEGV
[oppel@aurora] 228 (OpenMolcas/build_nec): file core.428091
core.428091: ELF 64-bit LSB core file x86-64, version 1 (SYSV), SVR4-style, from
'/opt/nec/ve/nfort/2.5.0/libexec/fcom -I. -I/public/aurora/scratch/oppel/molcas
/', real uid: 5964, effective uid: 5964, real gid: 538, effective gid: 538, exec
fn: '/opt/nec/ve/nfort/2.5.0/libexec/fcom', platform: 'x86_64'
[oppel@aurora] 229 (OpenMolcas/build_nec): █

```

Open

Com

• Ac

W

• Ch

—

—

—

sität

le

rce

# OpenMolcas – Compilation – Part III



**Link step successful, if:**

- **Choose the right linking method**
  - Linking of .so files doesn't work
- **Linking of ASL's lapack/blas using static libraries**

Op

File Edit View Search Terminal Help

**[oppel@aurora] 273 (OpenMolcas/build\_nec):**

**[oppel@aurora] 273 (OpenMolcas/build\_nec):** /opt/nec/ve/bin/nfort -fpp -fdefault-integer=8 -O2 CMakeFiles/alaska.exe.dir/src/alaska/main.f.o CMakeFiles/alaska.dir/src/alaska/alaska.f.o CMakeFiles/alaska.dir/src/alaska/alaska\_super\_driver.f.o CMakeFiles/alaska.dir/src/alaska/chk\_numerical.f.o CMakeFiles/alaska.dir/src/alaska/cho\_alaska\_rdinp.f.o CMakeFiles/alaska.dir/src/alaska/drvdftg.f.o CMakeFiles/alaska.dir/src/alaska/drvcmbg.f.o CMakeFiles/alaska.dir/src/alaska/drvgl.f.o CMakeFiles/alaska.dir/src/alaska/drvh1\_emb.f.o CMakeFiles/alaska.dir/src/alaska/drvn1.f.o CMakeFiles/alaska.dir/src/alaska/drvn1\_emb.f.o CMakeFiles/alaska.dir/src/alaska/inputg.f.o CMakeFiles/alaska.dir/src/alaska/isoind.f.o CMakeFiles/alaska.dir/src/alaska/pritim.f.o CMakeFiles/alaska.dir/src/alaska/trnglr.f.o -o bin/alaska.exe lib/libmolcas.a /opt/nec/ve/nlc/2.0.0/lib/liblapack\_i64.so /opt/nec/ve/nlc/2.0.0/lib/libblas\_sequential\_i64.so

Link

• C

•

• Li

"liblapack\_i64.so", line 4073: error: missing ".  
 "liblapack\_i64.so", line 4104: error: missing ".  
 "liblapack\_i64.so", line 4120: error: missing ".  
 "liblapack\_i64.so", line 4128: error: missing ".  
 "liblapack\_i64.so", line 4565: error: missing '.  
 "liblapack\_i64.so", line 4573: error: missing '.  
 "liblapack\_i64.so", line 4580: error: missing '.  
 "liblapack\_i64.so", line 6590: error: missing ".  
 "liblapack\_i64.so", line 20821: error: missing ".  
 "liblapack\_i64.so", line 32605: error: missing ".  
 "liblapack\_i64.so", line 34166: error: missing ".



Op

File Edit View Search Terminal Help

```

[oppe] 280 (OpenMolcas/build_nec):
[oppe] 280 (OpenMolcas/build_nec):
[oppe] 280 (OpenMolcas/build_nec):
[oppe] 280 (OpenMolcas/build_nec): /opt/nec/ve/bin/nfort -fpp -fdefault-
integer=8 -O2 CMakeFiles/alaska.exe.dir/src/alaska/main.f.o CMakeFiles/alaska.di
r/src/alaska/alaska.f.o CMakeFiles/alaska.dir/src/alaska/alaska_super_driver.f.o
CMakeFiles/alaska.dir/src/alaska/chk_numerical.f.o CMakeFiles/alaska.dir/src/al
aska/cho_alaska_rdinp.f.o CMakeFiles/alaska.dir/src/alaska/drvdftg.f.o CMakeFile
s/alaska.dir/src/alaska/drvebg.f.o CMakeFiles/alaska.dir/src/alaska/drvgl.f.o C
MakeFiles/alaska.dir/src/alaska/drvhl_emb.f.o CMakeFiles/alaska.dir/src/alaska/d
rvnl.f.o CMakeFiles/alaska.dir/src/alaska/drvnl_emb.f.o CMakeFiles/alaska.dir/sr
c/alaska/inputg.f.o CMakeFiles/alaska.dir/src/alaska/isoind.f.o CMakeFiles/alask
a.dir/src/alaska/pritim.f.o CMakeFiles/alaska.dir/src/alaska/trnglr.f.o -o bin/a
laska.exe lib/libmolcas.a /opt/nec/ve/nlc/2.0.0/lib/liblapack_i64.a /opt/nec/ve/
nlc/2.0.0/lib/libblas_sequential_i64.a
[oppe] 281 (OpenMolcas/build_nec): ls -l bin/alaska.exe
-rwxr-xr-x 1 oppe agleti 13243136 Nov 25 11:13 bin/alaska.exe
[oppe] 282 (OpenMolcas/build_nec): █

```

Link

- C
- 
- Li

# OpenMolcas verification tests

# OpenMolcas verification tests

- **OpenMolcas comes with an extensive test suite to check the integrity of the program.**
- **Run standard verification tests (pymolcas verify 000-099)**
- **Standard compiler optimization (-O2)**
  - 46 ok, 50 failed (mostly rasscf)
- **Recompile with -O1**
  - 84 ok, 12 failed (in rasscf)

## Opt 2.2 Optimization Options

`-O[n]`

- Specifies optimization level by  $n$ . The following are available as  $n$ :
  - **4**  
Enables aggressive optimization which violates language standard.
  - **3**  
Enables optimization which causes side-effects and nested loop optimization.
  - **2**  
Enables optimization which causes side-effects. (default)
  - **1**  
Enables optimization which does not cause any side effects.
  - **0**  
Disables any optimizations, automatic vectorization, parallelization, and inlining.

he

## Op 2.2 Optimization Options

`-O[n]`

- Specifies optimization level by  $n$ . The following are available as  $n$ :
  - 4  
Enables aggressive optimization which violates language standard.
  - 3  
Enables optimization which causes side-effects and nested loop optimization.
  - 2  
Enables optimization which causes side-effects (default)
  - 1  
Enables optimization which does not cause any side effects.
  - 0  
Disables any optimizations, automatic vectorization, parallelization, and inlining.

he

## Op 2.2 Optimization Options

`-O[n]`

- ( Specifies optimization level by  $n$ . The following are available as  $n$ :
  - 4  
Enables aggressive optimization which violates language standard.
  - 3  
Enables optimization which causes side-effects and nested loop optimization.
  - 2  
Enables optimization which causes side-effects (default)
  - 1  
Enables optimization which does not cause any side effects.
  - 0  
Disables any optimizations, automatic vectorization, parallelization, and inlining.

he

# Runtime

Test example: Pyrrole

Integrals (6-31G\* vs. **AUG-CC-VDZ**), SCF, CASSCF (6,6) – 6 Roots

Module	Intel	NEC –O2	NEC –O1
Seward	2 sec (20 sec)	22 sec	22 sec (2 min 50 sec)
SCF	2 sec (22 sec)	6 sec	6 sec (1 min 36 sec)
RASSCF	7 sec (1 min 41 sec)	failure	38 sec (7 min 58 sec)

# Summary



# Summary

- **Compilation of complex program packages (quantum chemistry) possible**
- **Small problems with compiler flags, linking of libraries**
- **Numerical stability not guaranteed – needs further investigation**
- **No easy performance gain – needs more detailed investigation**

# Outlook - Wishlist

- **Compatibility of Fortran compiler flags to gfortran**
  - ncc seems to be ok
- **More tools/toolchains provided by NEC:**
  - Boost library (for C++ based codes)
  - ...
- **Remove license management for compilers (makes no sense!)**

# Acknowledgement

# Acknowledgement

- **Max Tiefenbacher (master student)**
- **University of Vienna – Faculty of Chemistry  
(Investment Project IP527022)**

# Acknowledgement

- Max Tiefenbacher (master student)
- University of Vienna – Faculty of Chemistry  
(Investment Project IP527022)



# Acknowledgement

- Max Tiefenbacher (master student)
- University of Vienna – Faculty of Chemistry  
(Investment Project IP527022)



# Acknowledgement

- Max Tiefenbacher (master student)
- University of Vienna – Faculty of Chemistry  
(Investment Project IP527022)





# Acknowledgement

- Max Tiefenbacher (master student)
- University of Vienna – Faculty of Chemistry  
(Investment Project IP527022)



2x



# Acknowledgement

- Max Tiefenbacher (master student)
- University of Vienna – Faculty of Chemistry  
(Investment Project IP527022)



2x





# Simply recompile – does it work?

# Simply recompile – does it work?

- **Works out of the box for simple programs**
- **Testing real world applications**
  - Simple Quantum Dynamics code written in Fortran77
  - Quantum Chemistry Programs: OpenMolcas

# Quantum dynamics codes

- **Fortran77 based code written by Burkhard Schmidt (FU Berlin)**
- **Two main programs**
  - Bound state calculation (time independent Schrödinger equation)
  - Propagation of nuclear wave packets (time dependent Schrödinger equation)
- **Recompilation for SX-Aurora possible without any modifications**
- **Link NEC's optimized math library for Linear Algebra and FFT**
- **Example calculations: N-H stretch of pyrrole, 2 electronic states (ab initio data points)**

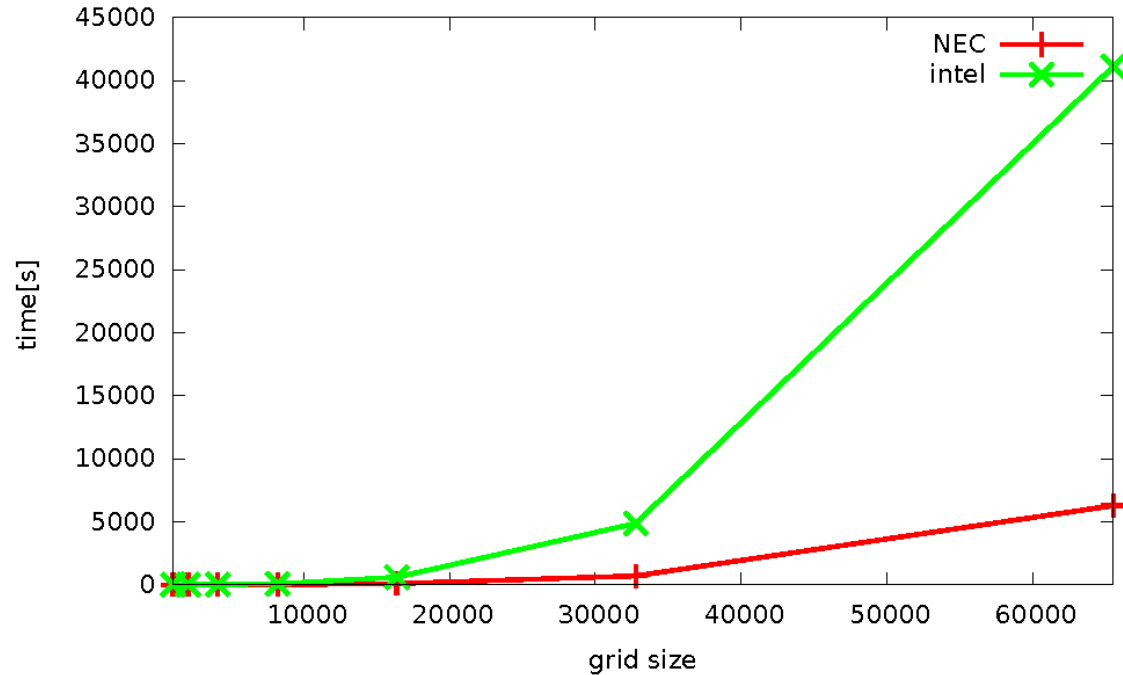
# Bound state calculations

- **Direct diagonalization of Hamiltonian on grid**
- **Number of grid points: 1024-65536**
- **Main time spend in dsyev from LAPACK**
- **Time for SX-Aurora vs. Intel Skylake Gold 6148**



# Bound state calculations

- Direct diagonalization of Hamiltonian on grid
- Number of grid points: 1024-65536
- Main time spend in dsyev from LAPACK
- Time for SX-Aurora vs. Intel Skylake Gold 6148



# Propagation

- **1d propagation on grid**
- **Split Operator**
- **Number of grid points:  
1024-65536**
- **Main time spend in  
FFT from fftw3**
- **Time for SX-Aurora vs.  
Intel Skylake Gold 6148**

# Propagation

- 1d propagation on grid
- Split Operator
- Number of grid points:  
1024-65536
- Main time spend in  
FFT from fftw3
- Time for SX-Aurora vs.  
Intel Skylake Gold 6148

