State of the Union Octopus

Micael Oliveira

Max Planck Institute for the Structure and Dynamics of Matter, Hamburg, Germany

Octopus Developers Meeting, September 29, 2021
Back in 2001...

commit 822cf8dccad0d4225bdef51546400ba7c68cc824
Author: Miguel Marques <marques@tddft.org>
Date:   Tue Dec 4 03:51:19 2001 +0000

Very preliminary version of new tddft code
now baptized octopus ;))

Octopus is 20 years old!
Major milestones since last meeting (June 2019)

- Two major releases (10 and 11) and six bugfix releases
- 822 approved merge requests, more than 5000 commits
- (Numbers from previous meeting: 470 MR and 2700 commits)
Major milestones since last meeting (June 2019)

- Improved GPU support (see talk by Sebastian)
- Improved ground-state calculations
- Many optimizations and bugfixes
- Introduction of OOP features of Fortran 2003/2008
- Major refactoring of core components in progress
Major milestones since last meeting (June 2019)

- New multi-system framework (see talk later today)
- New Maxwell and DFTB+ systems (see talk by Franco)
- Magnons from real-time TDDFT (see talk by Nicolas)
- Hybrid functionals:
  - Support for solids
  - ACE operator and improved parallelization
  - Range-separated functionals
- Several run modes extended to handle photons within QEDFT (see talks by Davis and Johannes)

Full changelog: https://octopus-code.org/wiki/Changes
Ground-state improvements

Eigensolvers:
- Conjugate gradients
  - Corrected implementation: order of steps
  - Introduce normalizations
  - Adapt convergence criteria of eigensolver loop
- RMMDIIS
  - several corrections and bugfixes

Improvement in convergence
- Before, difficulties below $10^{-8}$ in relative densities
- Now, convergence to $10^{-15}$ possible for many systems
Ground-state improvements

Broyden mixing:
- Remove normalization of intermediate quantities
- Enable fast convergence to very high accuracy
- Implement restarting: mixing history is reset after a while, helps finding the right minimum

Preconditioners:
- Filter preconditioner
  - Theoretical understanding in terms of Jacobi iterations
  - Generalization to non-orthogonal cells
- Generalized multigrid preconditioner
Ground-state improvements

Octopus 8.4

Octopus, improved CG + Mixing

Relative density change

Iterations

10^{-16} 10^{-14} 10^{-12} 10^{-10} 10^{-8} 10^{-6} 10^{-4} 10^{-2} 10^{0}

0 25 50 75 100 125 150 175 200

Iterations

1ALA ZALA 2Dharmonic 3ALA 4ALA ASC NiO Si TiO_2 WSe_2 benzene methane oxyluciferin
Performance optimizations

- Casida for huge matrices (up to 100k), used for photon coupling
- Improvements of the OpenMP parallelization
  - More pragmas for loops, some are more complicated (e.g. norm)
  - Good scaling up to about 12 threads per rank
  - Only good for large grids, can substitute domain parallelization
- Application of phase for periodic systems
  - Only once at beginning and end of time step
  - Use phase correction for boundary points
  - Can save up to 50-80% of the computing time
- Allocate aligned memory for better vectorization, introduce AVX512 instructions
- Improve finite-difference kernels by using non-temporal store instructions
  - Directly store to memory without cache
  - Improve performance of kernel
How much faster does the Octopus swim?

![Bar chart showing the complete time in seconds for different Octopus versions.](chart.png)
How much faster does the Octopus swim?

![Bar graph showing complete time in seconds for different Octopus versions]

Cubic Si, 32 k-points - GS run

Speed-up: 2.1

<table>
<thead>
<tr>
<th>Octopus version</th>
<th>Complete time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0 (2016)</td>
<td>10</td>
</tr>
<tr>
<td>7.2 (2017)</td>
<td>12</td>
</tr>
<tr>
<td>8.4 (2018)</td>
<td>10</td>
</tr>
<tr>
<td>9.2 (2019)</td>
<td>8</td>
</tr>
<tr>
<td>10.3 (2020)</td>
<td>6</td>
</tr>
<tr>
<td>develop (2021-01)</td>
<td>5</td>
</tr>
</tbody>
</table>
How much faster does the Octopus swim?

Cubic Si, 32 k-points - TD run

- 6.0 (2016)
- 7.2 (2017)
- 8.4 (2018)
- 9.2 (2019)
- 10.3 (2020)
- develop (2021-01)
How much faster does the Octopus swim?
Test-suites and test-farm news

Regression test-suite:
- New conditional execution and matches
- Allow for expected execution failures in tests

Performance test-suite:
- Two new Buildbot builders on dedicated hardware
- New web app to visualize results
- Still under development:
  - Only “unit” tests are run, no main run modes (gs, td)
  - Parameter space needs to be reduced (tests take too long)
  - How to determine success/failure?
  - Too large dispersion of timings
Test-suites and test-farm news

Buildbot and test-farm:
- Two new GPU machines with 10 GeForce RTX 2080 Ti cards each
- Next generation of dedicated machines will be available soon
- New builders to check GCC compiler warnings
- New OpenMP+GCC builders
Test-suites and test-farm wishlist

- New compilers?
- Refactorize testsuite to improve coverage and speed
- Meta-tests for regression tests:
  - Detect when tolerances are much too large
  - Detect when test dispersion is too large
- Better integration of EasyBuild with Buildbot
- Builders for all major Linux distributions
Other news

- New website under construction (see talk by Martin)
- MR reviews mandatory since last meeting
- Martin is the new release manager
- Two on-line courses in September:
  - Octopus basics: learn how to use Octopus (39 participants)
  - Octopus advanced: learn how to develop Octopus (25 participants)
- Weekly meetings on Tuesdays at 9:00 am
- Regular hackathons
Other news

Things that did not go so well:

- Still 157 features marked as experimental (159 in Octopus 9.0)
- 211 open issues; 289 closed
- Web server was hacked in July 2019
- Mailing lists are still down