Verification & Validation (VV) – ideas suggested at Webex on 26 June 2019

Seb, Andrew, Nicolas, Pierre, Claire, Julien, Mike, Simon

These notes summarise points made in the meeting. An attempt has been made to put the points into a coherent order.

SETTE system

The current system is good because it is quite lightweight (it "just" launches tests) so it is easy to set up and run. Further work could make it easier to use. Weaknesses at present are that not all the NEMO code is tested and only some aspects such as restartability are tested. SETTE does not test whether results are reproduced as before or conservation properties. Only a basic set of decompositions is tested (4x8 and 8x4); odd numbers (e.g. 3x7) can uncover additional problems; memory and performance checks/tests are not included. It would also be good to be able to add a new configuration to SETTE.

The large number of compilations involved in SETTE does create a bottleneck but parallelising would complicate the system. The suppression of cpp keys has reduced the number of compilations required to test the standard configurations.

Unit testing and focused testing

Unit testing usually means that each function (or subroutine) in a system is provided with a driver which tests that the outputs for given inputs are as expected. It can mean that every path through a function is tested to ensure that the outputs are as expected.

There are tools which assist unit testing and help to assess what fraction of (paths through) the code are being test.

Some ideas for specific / focused tests:

- Rotation through 90^o or 180^o should not affect the results. There are tools to manipulate netcdf files to facilitate these tests
- Uniform T & S initial fields should stay the same
- Verify conservation properties
- Unit tests of individual modules – e.g. bdy code; advection code
- Outputs from idealised configurations

The outputs from tests need to be in a form that can be automatically checked (yes/no answer).

NEMO is quite modular so it might be possible to build a driver that can be used to test a large number of NEMO's subroutines.

Development & testing processes and approach

The trusting tool enables automated production of web-pages for the results from SETTE. This should be considered to be a distinct functionality (separate from the SETTE tool). The trusting tool has been implemented at CNRS but is currently not running there. Implementing it on another system would be a task in itself. Outputs from some other versioning tools (e.g. git rather than svn) may be easier to share. Trusting could enable each consortium member to run one or two reference configurations (e.g. ORCA025) with the head of the trunk every night.

Some developers are committing before testing. This issue is partly a matter of discipline. The development process/system could be designed to make it more difficult (or impossible) to commit before testing. A development and its testing are not sufficiently connected. We may not be exploiting tools for unit testing and continuous integration as well as we could.

Some of the issues with our current approach to development & verification are :

- the different levels (or types) of testing that we should do (multi-core & single core; across different machines; to ensure soundness of physics) are not very clearly articulated
- developers' short-cuts can compromise the robustness of the development process
- developers own the task of solving problems arising from their developments (as they should) but the "process" for providing assistance to them is not very clear and falls to the same people all the time – the coordination of and recognition for this work should be improved.

The process for validation of large scale configurations is not very clear (it's a complicated issue); [The observation operator is not used for this task as much as it should be.] Validation seems to be "only" done in a rush at a late stage in the development of a new version(?) Repeated validation of large configurations would require significant computational resources.

Resources / Motivation

The NEMO System Team has limited resources for VV work at present so has had to be "pragmatic" in its ambitions. There is a danger that ambitions to improve VV will become long-standing dreams (similar to those for documentation). Nonetheless there are clear opportunities to improve rapidly in testing some specific functionalities (see specific / focused tests above). CNRS is also committed to do some work on NEMO's continuous integration process in IMMERSE.

VV is central to the quality of NEMO (it is the quality assurance process). It is arguable that 50% of the NEMO work-plan should be devoted to it. If we can clearly articulate its benefits (with examples of current deficiencies) and develop a work-plan for its improvement a good case could be made for specific funding for it within an EC project. A first step towards this would be to place our VV activities within a suitable VV framework / structure with some ideas for short-term and long-term developments to our current systems and practices (aka a roadmap). [The first version of this should not be more than a few pages.] It would be good for a small team to do this. It should include pragmatists and vision people. Mike asked for volunteers to email him.