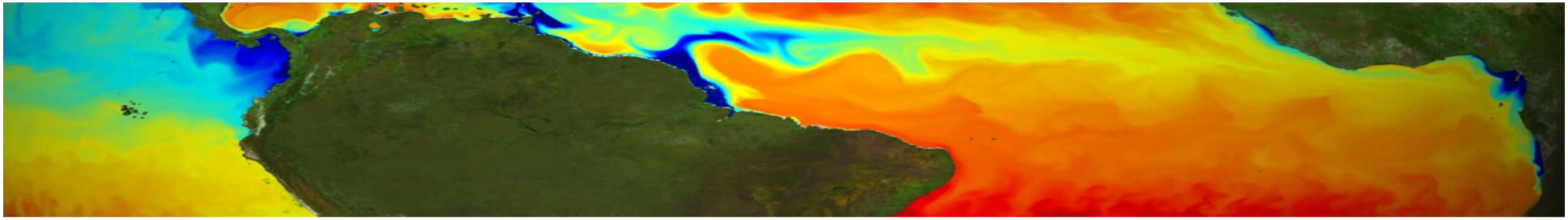

Verification & Validation of NEMO

NEMO Developers' Committee

12th Sept 2019

Contents

- Summary of main points from two webex discussion meetings
  - What do we mean by verification & validation (V&V) ?
  - What do we have now ?
    - SETTE & Trusting tests – strengths & weaknesses
  - What could we have ?
    - Unit testing & other focussed tests
  - What could we afford ?
    - Pragmatism & processes
    - Importance & resources
- Discussion of way ahead

What do we mean by V&V ?

- Verification : confirming a code does what it is intended to do: e.g.
  - Unit testing (checks outputs as expected for given inputs)
  - Repeatability (results do not change over time)
  - Restartability (results after a restart same as a continuous run)
  - Reproduceability (results don't depend on domain decompositions)
  - Bit comparison when expected across versions

- Validation: scientific quality as expected for:
  - a specific process (e.g. an advection scheme)
  - an aspect of simulation (e.g. energy conservation)
  - overall skill/quality of a system (e.g. comparison with observations)

What do we have?: SETTE tool

- What it does:
  - Allows some configurations (both “real” & test cases) to be checked for
    - Successful compilation
    - Reproduceability and restartability
    - [check agreement with a previous version (new functionality)]

- Some strengths:
  - Light-weight – easy to install & use
  - Familiar to the NEMO community
  - Immediate validation of new developments

- Some current limitations:
  - It is possible to commit to the trunk without passing SETTE tests
  - It is not possible at the moment to compare against other peoples’ results
  - Somewhat slow to run and difficult to extend
  - Not all debug options are tested (e.g. out of bounds tests)
  - Not all paths through code are tested (inevitable but “uncontrolled”)
  - Only basic decompositions are tested (e.g. 4 x 8 but not 3 x 7)

What do we have?: Trusting tool

- What it does:
  - Enables automated production of web pages from SETTE results
  - Only repeatability over revisions at the moment
  - (Restartability & reproducibility plans in IMMERSE)

- Strengths
  - Can be run automatically (by “cron” job) so help track code evolution
  - Tools can help pin issues to particular fields or commits
  - Can be run at several centres for several configs & branches (no hard limits)

- Weaknesses / issues
  - It flags up issues but does not solve them; someone working in part A of the code generates an issue in part B. This is not easy to solve
  - The tool was turned off to sort out other issues …
  - Still in development (early version)

What could we have?

- Unit testing
  - Usually means each subroutine is provided with a driver
  - Checks outputs are as expected for given inputs
  - Can mean every path through code must be tested
  - NEMO is quite modular so a generic driver might be possible?
  - Needs to be set up to re-run as code evolves

- Examples of focused testing
  - Rotation through 180^{o} / 360^{o} should not change results
  - Uniform T & S should stay the same; energy conservation
  - Unit tests of individual modules
  - Outputs from idealised configurations

- Outputs need to give a pass/fail answer

What could we afford?

Pragmatism & Processes

- The System Team has taken a pragmatic approach to V&V: time constraints limit the work done
- We do not want impractical visions for V&V

- However, several processes could be more clearly articulated: e.g.
  - The levels/types of testing performed
  - What tests are covered and what are not
  - What developer short-cuts (avoiding SETTE) should not be allowed
  - Coordination & recognition of solution of problems flagged by Trusting
  - How large configurations should be validated before releases

- MOM6 has developed interesting processes (e.g. faster release turn-round)

What could we afford?

Importance & Resources

- Importance
  - How confident are we about the V&V of the core code?
  - If we put more effort into V&V what returns do we expect?

- Resources: what fraction of System Team time
  - is devoted to this already?
  - should be devoted to this?

Proposed next steps

- Discuss MOM6 approach to V&V with Alistair Adcroft (IMMERSE action)
  - Are there other code systems we could learn from

- Generate a roadmap for future V&V activities
  - Place our V&V activities within a suitable structure / framework
  - Outline ideas for short-term & long-term developments
  - First draft should not be longer than a few pages and be written in 3-4 months
  - Need a small (@ 5 member) team of volunteers
    - Mike Bell, Nicolas Martin & Simon Muller have volunteered