Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2017-47-AC2, 2017

Geoscientific Model Development Discussions
Open Access

Interactive comment on "Shingle 2.0: generalising self-consistent and automated domain discretisation for multi-scale geophysical models" *by* Adam Candy and Julie Pietrzak

Adam Candy and Julie Pietrzak

a.s.candy@tudelft.nl

Received and published: 26 September 2017

Anonymous Referee #2

We would like to thank the reviewer for their constructive comments which have enabled us to produce a revised version which we feel is significantly improved. Parts of the review that have been included below are shown in *italics* and quotes from the paper in serif type.

General comments

This paper describes Shingle 2.0 – a Python-based library for the manipulation of spatial domains

Printer-friendly version

Discussion paper

and unstructured grids for geophysical problems. Through the use of a new XML-based file-format (BRML) and a hierarchy of publicly available software components, Shingle aims to standardise the process of managing the spatial constraints and unstructured grids associated with geophysical domains. To this end, a set of nine "tenets" for geophysical grid-generation are proposed, designed to facilitate the development of consistent and shareable frameworks for unstructured geophysical data and meshes.

The overall idea behind the Shingle library – the development of standardised approaches and formats for unstructured geophysical data – is interesting, as current methodologies are clearly ad-hoc. I do however have concerns regarding the distinction between the functionality of the Shingle package itself and the underlying libraries on which it depends. I suggest that a clear summary of the various dependencies be presented early in the paper, with an explicit delineation of functionality. Currently, it appears that:

- *The Gmsh package provides the actual meshing capabilities, based on a geometry definition created by Shingle.*
- *The Spud package is used to support the XML-based BRML file-format. It's Diamond viewer is used for GUI-based file editing.*
- *Various packages (GDAL, shapely, pyproj) are used to support geometrical operations and queries.*
- *The pydap package is used for remote data access.*

Does Shingle incorporate original algorithms and/or data processing facilities beyond those provided by the underlying libraries? If so, I suggest that these features be documented and novelty demonstrated, etc.

We agree that it is helpful to include a list of the dependencies and discuss their function, to supplement the details in the section *'5.1 Built on standard libraries'* of *'5 Lib-Shingle, the Shingle library framework'*. We have added the new table 2 there which lists the functions of Shingle that depend on external libraries. This acts as a good summary, including the details the reviewer suggested above and expanding where possible. Notably this includes reference to the Gmsh library for tessellation algorithms, Spud and Diamond for parameter management and GUI, standard libraries for geospatial operations such as GDAL and OPeNDAP libraries for remote data access.

We also took the opportunity to review the text on how dependencies may affect the reproducibility of output that appears in the conclusions at line 605. This covers both internal and external factors, including these external libraries. The new table 3 lists these dependencies, together with the potential deviations they may cause, the risk and mitigation approaches employed. In the large part this is an issue with all numerical simulation models which are linked to and use other libraries. The versions of these depend on the build environment, which can vary between systems and over time. Here we are taking the opportunity to be more explicit about these dependencies.

We see the use of external libraries a strength of the approach (cf. tenet 9). This is particularly the case where these are standard, well-regarded and well-tested. These are supported by the community, undergo strict verification testing and validated in a wide range of applications. This is the strength of joint, community efforts such as the PETSc (Portable, Extensible Toolkit for Scientific Computation, `https://www.mcs.anl.gov/petsc`) library of numerical algorithms.

In some cases it may be easier to implement an algorithm from scratch, but this would yield yet another ad hoc approach, more code and features to manage directly. Moreover, this will not benefit from future new features and support added to external libraries. These focused efforts arguably do a better job in the long term and provide a more sustainable approach than a reimplementation or development from scratch.

> *Gmsh has also been used for geophysical grid-generation in the past (e.g. Lambrechts et al., 2008: "Multiscale mesh generation on the sphere"), along with a number of other algorithms and libraries, including: Jacobsen et al., 2013: "Parallel algorithms for planar and spherical Delaunay construction with an application to centroidal Voronoi tessellations", Conroy et al., 2012: "ADMESH: An advanced, automatic unstructured mesh generator for shallow water models" and Holleman et al., 2013: (Stomel, in) "Numerical diffusion for flow-aligned unstructured grids with application to estuarine modeling", amongst others. I suggest including a brief review of these previous efforts, demonstrating the benefits of Shingle compared to existing alternatives.*

We agree that the paper would benefit from a review of previous efforts. Moreover,

Printer-friendly version

Discussion paper

whilst the advantages of the approach are expressed throughout the paper, we agree it is also helpful to collect and focus points here discussing the advantages of Shingle compared to existing alternatives. In response we have added the new section 2.4 on *'Tessellation algorithms and existing grid generation approaches'*, that we think significantly improves the manuscript on this point.

> *Additionally, I feel that the use of the Gmsh library should not be understated. While Shingle aims to overcome challenges related to the specification of the domain, geometric constraints, etc, I suggest that it is the underlying 'mesh-generation' process that is somewhat more algorithmically and computationally demanding.*

We agree that the quality of a spatial discretisation is directly dependent on both (i) the algorithm used to create a tessellation and (ii) accuracy and self-consistency of constraints under which the former operates (now stated in section 2.4, lines 230-40). Here we seek to improve the latter with the nine tenets in mind, and as the reviewer highlights, overcome the challenges related to the specification of the domain and geometric constraints.

We strongly support the reviewer's point that Gmsh has an important role in the approach and for mesh generation in general. We have emphasized this throughout the paper, and notably in lines 199, 222, 235, 286, 408, 529, 611, the new section 2.4 and new tables 2 and 3. Line 200 of section 2.4 in particular states, *'The general-purpose three-dimensional meshing library Gmsh (Geuzaine and Remacle, 2009) has been used to make significant progress in ocean modelling on unstructured meshes (e.g. see Legrand et al., 2000; White et al., 2008; van Scheltinga et al., 2010; Gourgue et al., 2013; Thomas et al., 2014)'*. It was important to facilitate the use of other libraries (line 526) in line with the ninth tenet on standardisation. Whilst use of other libraries is possible, compared to many alternatives, Gmsh does a very good job at adhering to the constraints provided, is relatively robust and provides access to multiple tessellation algorithms through a common API, again in support of tenet 9 (all points now added to section 6 from line 530).

The Candy, 2016 pre-print is referred to throughout, often to provide specific examples of functionality. I suggest that any examples referred to be included in the current paper directly. There appears to be some overlap between these papers, though the Candy, 2016 work appears to focus on more theoretical issues.

We agree with the reviewer that the paper Candy (2016) has separate, distinct aims focusing on consistency and theoretical issues. The reviewer is correct that there is necessarily some overlap. We believe this is needed, to draw connections between the works and motivate aims here, but have tried to minimise as much as possible.

A selection of example discretisations are shown in figure 1, some of which appear in the paper Candy (2016). These are included to motivate the aims of the paper – that a generalised approach is needed that is model-independent and applicable to a range of Earth Systems. The only other case also appearing in Candy (2016) is part of figure 9, which was useful to include to highlight that global domains can be considered. All other cases are new and do not appear elsewhere. This includes the full worked example of the 2010 Chile tsunami in figures 4, 5 and 7; the Caribbean Sea basin in figure 8; and new studies on selected regions of Antarctica also in figure 9.

The reviewer is correct that Candy (2016) is referenced throughout to connect the efforts. We have reviewed all of these points in the text and where appropriate referenced material is included or summarised. Notably the key results of the constraints and tenets are included in a reduced form in section 2.1 and table 1, respectively. The global case is included in figure 9 as the reviewer suggests. The only other reference back to examples is made in the sentence in lines 137-40, which is not to provide specific examples of functionality, but a broad statement is made to highlight the generalised approach and range of cases this is applicable to. We were careful to ensure this work included distinct example cases, with the 2010 Chile tsunami domain acting as the main full worked example.

Technical corrections

- *Page 2, line 32: ... [is] likely to grow.*

  The word ‘is’ added such that the sentence reads better, as suggested.

- *Page 4, line 57: “... the meshing process is broken up over multiple parallel threads (as demonstrated in Candy, 2016), ...” Does Shingle itself manage the parallel meshing process, or is this handled by the Gmsh library?*

  Yes, the Shingle library handles the discretisation over multiple parallel threads. Gmsh is currently used in serial. This not a limitation of the Shingle library. We have recently seen a paper published demonstrating Gmsh running over multiple threads at a fine grain level. This would certainly benefit the approach here and we look forward to using Gmsh in this way, parallel over multiple threads, in the future.

- *Page 5, line 102: develop[er]s*

  Corrected to ‘developers’.

Bibliography

Details of all references cited above are provided in the revised paper.

Please also note the supplement to this comment:
https://www.geosci-model-dev-discuss.net/gmd-2017-47/gmd-2017-47-AC2-supplement.pdf

---

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-47, 2017.

Printer-friendly version

Discussion paper