Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-325-RC2, 2021

Geoscientific Model Development Discussions

Interactive comment on "Lossy Checkpoint Compression in Full Waveform Inversion" *by* Navjot Kukreja et al.

Anonymous Referee #2

Received and published: 25 March 2021

The manuscript "Lossy Checkpoint Compression in Full Waveform Inversion" by Kukreja et al. addresses the huge memory requirements of time-domain adjoint simulations that require access to the entire forward wavefield in reverse order. The excellent parallel scalability of time-domain simulations initiated a recent trend to move away from frequency-domain solutions based on the Helmholtz equation, especially on modern hardware using GPU accelerators and for visco-elastic wave propagation. The authors correctly identify the tremendous memory footprint as the major bottleneck in time-domain approaches. Realistic large-scale simulations exceed any reasonable storage capacity by far, making checkpointing and/or wavefield compression approaches inevitable.

I don't know why the industry seems somewhat slow to adopt these techniques, but I assume this is due to using primarily the acoustic wave equation and having a long history in successful inversions using a small set of discrete frequencies, which can be computed in the time-domain with the help of on-the-fly Fourier transforms. However, checkpointing has become state-of-the-art for inverting broadband seismic in continental- to global-scale seismology, and codes like SpecFEM or Salvus provide implementations of it. The authors propose to combine conventional checkpointing with additional lossy compression of the checkpoints to further reduce the memory requirements, which is a neat idea and highly desirable for the above-mentioned reasons.

My major points of criticism are:

- The idea of combining checkpointing and compression itself is not new and has been discussed by the same authors already in a previous publication. Some parts of the manuscript read more like a lab report and some parts are written quite sloppily (see examples in the minor comments below). I am also missing a discussion section that provides a bit of context for the validity of the results for other media. For instance, to what extent does the achievable compression factor depend on the medium, physics, aperture, misfit function, etc.

- I would have assumed that application-tailored compression algorithms would outperform black-box compression tools. Such a comparison would be very interesting and I find it currently missing. Examples of such techniques can be found, for instance, in Boehm et al. (Geophysics, 2016) or Weiser & Gotschel (SISC 2012).

- I am surprised that there is no adaptivity in steering the compression settings throughout the inversion. From an optimization perspective with inexact derivatives, it is well-known that more accurate gradients would be required when approaching a stationary point. I can't find this mentioned or analyzed in the manuscript.

- The manuscript would really benefit from a 3D visco-elastic example as this is where the memory bottleneck becomes a lot more prominent. Of course, I am not asking for a 3D FWI example, but I would strongly encourage the authors to provide an error analysis for a single visco-elastic gradient computation in 3D.

A few more minor comments:

- You reference eq. (1) at least twice already in the introduction, way before the equation is actually introduced. Similar premature references exist for eq. (4) and symbols like Phi(m).

- I also don't think this should be focusing on the non-dissipative acoustic wave equation. The compression approach applies to all time-dependent wave equations and it is a lot more relevant for visco-elastic media. You already hint that eq. (1) is not really the equation of interest on p.3, line 48.

- I think the statements on p.3 lines 50-56 are misleading and / or incorrect. The communication overhead of MPI-parallelized simulations can well be hidden behind computations by computing the halo first and performing asynchronous communication. But even on distributed compute architectures reducing the memory footprint is highly desired. Furthermore, many frequency-domain methods indeed have a HUGE memory footprint when factorizing the Helmholtz operator. At least, you would need to be more specific what you mean by frequency-domain here.

- p.5, line 106. Typo: There is an additional ")" in (2016)).

- Section 2. The notation mixes bold and italic symbols, for instance, for the model m and d_obs / d_sim, phi... The operator A is not introduced properly. It is the discretization of the PDE operator, and not of the equation.

- I would consider merging sections 3 and 4, because section 3 merely contains extended headings for the subsections of part 4.

- Many figures contain similar quantities and could be merged. For instance, Fig 7/8, Fig 12/13, Fig 14/15, Fig 20/21 could all be merged into a single row each.

- The maximum buffer size used in Fig 7 and 8 is fairly small and I would consider extending the line to at least twice the number of time steps.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2020-325, 2020.