Reponses to Anonymous Referee #2

General Response

We are grateful to the reviewer for his/her careful reading of the manuscript and his/her thoughtful comments. We have followed all suggestions in all cases, and provide below a point-by-point response to the reviews. These will be corrected in revised manuscript. This feedback has helped us improve and clarify the manuscript.

Point-by-point response

** In section 2.1: does "advanced time stepping" mean "adaptive time stepping"? You also mention that HOMME uses adaptive mesh refinement. Can you discuss the impact of these techniques on the accuracy of the linearization? Did you try running the TLM of an adaptive simulation? I don't think adaptivity was used in combination with the TLM, but I think this should be explicitly clarified.*

> **Authors' response**: As the reviewer point out, adaptive mesh refinement and time stepping are not used for TLM. To make it clearer, the following text is added and now reading in section the 2nd paragraph in section 2.1:
>
> "… Note that although the HOMME uses adaptive time stepping and adaptive mesh refinement, its TLM does not include such functions. ….

** Figure 2 is confusing: annotating the input and output variables with (I) and (O) isn't sensible fortran. (For example, gfortran errors with "Unexpected junk in formal argument list".) Fortran has a built-in mechanism for indicating intent, with the intent(in), intent(out) and intent(inout) attributes. I suggest that figure 2 be changed to use these instead, and to drop the unnecessarily novel (I)/(O) notation.*

> **Authors' response**: As the reviewer suggested, figure 2 is changed and followed the Fortran syntax, and the corresponding text is also changed in the 2nd paragraph in section 2.2.

```
Subroutine NL( a, b, tens )
real, intent(in) :: a, b
real, intent(out) :: tens
real :: tmp

tmp  = 3.0d0 * sin(a)
tens = tmp * b**2
End subroutine NL
```

```
Subroutine TL( a, b, tl_a, tl_b, tl_tens )
real, intent(in) :: a, b, tl_a, tl_b
real, intent(out) :: tl_tens
real :: tmp, tl_tmp

tl_tmp  = 3.0d0 * cos(a) * tl_a
tmp     = 3.0d0 * sin(a)
tl_tens = tl_tmp * b**2 + tmp * 2.0d0 * b * tl_b
End subroutine TL
```

** Section 2.4: "the high computational cost is extremely burdensome". This paper would be greatly improved with timing results — computational efficiency is crucial in 4DVAR applications of TLMs. Given that HOMME is explicit, the cost of (TLM + NLM) should be approximately twice that of the NLM, without any tricks (changing timesteps, or interpolating the original trajectory in time). Furthermore, the value of the tricks should be quantified by relating the error in the Taylor remainder (does the value of the LHS of equation (2) stay close to 1) against the computational savings (in seconds).*

Authors' response: Thanks for the excellent suggestions. We quantified total Wallclock run time and the mean relative error, defined, for different time steps, as

$$\frac{\left\|\mathbf{X}_{TL\_\Delta t}-\mathbf{X}_{NLD}\right\|}{\left\|\mathbf{X}_{NLD}\right\|},\frac{\left\|\mathbf{X}_{TL\_2\Delta t}-\mathbf{X}_{NLD}\right\|}{\left\|\mathbf{X}_{NLD}\right\|},\frac{\left\|\mathbf{X}_{TL\_3\Delta t}-\mathbf{X}_{NLD}\right\|}{\left\|\mathbf{X}_{NLD}\right\|},\frac{\left\|\mathbf{X}_{TL\_4\Delta t}-\mathbf{X}_{NLD}\right\|}{\left\|\mathbf{X}_{NLD}\right\|}$$

where $\mathbf{X}_{TL}$ is a TLM field at T=5 hour, $\mathbf{X}_{NLD}$ is the corresponding difference fields between the two nonlinear model forecasts at 5 hour, and || || is a spatial averaged norm.

The following table gives these values for the mean relative error at time T=5 hour:

Variable	**1*Δt**	**2*Δt**	**3*Δt**	**4*Δt**
u	0.0124556	0.0128355	0.0135081	0.163502
v	0.0128028	0.0120578	0.0115803	0.13647
t	0.00696689	0.00650514	0.00596657	0.104771
ps	0.00697304	0.00639369	0.00547336	0.0750567

Also, we evaluated the total wallclock run time (second) for two days simulation and showed below:

Total Wallclock time for 5 hour forecast		
1*Δt	55.490	100%
2*Δt	31.249	56%
3*Δt	20.204	36%
4*Δt	18.080	33%

We added the above results in the manuscripts, and the 2nd paragraph in section 3.3 is now reading:

"…. This can be confirmed quantitatively by considering the relative mean error, defined, for any quantity $\boldsymbol{X}$ at the time T=5 hr, as

$$|| \boldsymbol{X}_{\mathrm{TLM}} - \boldsymbol{X}_{\mathrm{NLD}} || / || \boldsymbol{X}_{\mathrm{NLD}} ||, \qquad (3)$$

where $\boldsymbol{X}_{\mathrm{TLM}}$ is a TLM field at T=5 hour, $\boldsymbol{X}_{\mathrm{NLD}}$ is the corresponding difference fields between t he two nonlinear model forecasts at 5 hour, and || || is a spatial averaged norm. Table 1 gives these values for the mean of the stat variable $\boldsymbol{X}$ at time T=5 hr. And the total wallclock time i s decreased, as the time step size is increased such that when Δt=150s is set to be 100%, 2Δt becomes 56%, 3Δt is 36%, and 4Δt for 33%...."

** The section on TAPENADE (note the misspelling in the manuscript) is quite strange. It's simply wrong that TAPENADE doesn't handle branches (see, e.g., figure 7 of 10.1145/2450153.2450158 for an example of TAPENADE applied to code with a branch). I strongly suggest that you consult with the developers of TAPENADE and make a fair comparison with AD tools (in particular in terms of computational efficiency). I expect that the hand-coded TLM will be quite a bit more efficient; quantifying this would improve the scientific value of the manuscript greatly, as it would give future developers of TLMs of other models in this area an idea of the tradeoffs.*

Authors' response: We agree with the reviewer in that many evaluations are necessary for fair comparison of TAPENADE-generated and hand-coded TLM. We leave this issue for the next study of ADM development and omit this paragraph in current version.

** The English used in the manuscript could be improved (for example, many articles are missing). I suggest having a native anglophone editor review the text and correct the minor grammatical errors.*

Authors' response: Done as suggested.

** The GMD manuscript types state that 'All papers must include a section at the end of the paper entitled "Code availability".' The code availability is discussed, but it's not in its own section as the journal requires. I suggest you review the discussion of the manuscript types and reformat appropriately.*

Authors' response: "Code availability" is now added in section 6, and reads:

"All codes in the current version of TLM are available upon the request. Any potential user interested in those modules should contact B.-J. Jung, and any feedback on them is welcome. Note that one may need help to use the TLM model optimally, but we do not have the resources to support the model in an open way. Since ADM is currently being developed based on the current version of TLM, all codes of ADM are also presumably available upon the request."

** Will the linearized version (and presumably, the future adjoint version) of HOMME be distributed with HOMME? I didn't see this mentioned in the text.*

Authors' response: This is discussed in Code availability.