Supplement of Geosci. Model Dev., 14, 4939–4975, 2021
https://doi.org/10.5194/gmd-14-4939-2021-supplement

Geoscientific Model Development Open Access EGU

Supplement of

WAP-1D-VAR v1.0: development and evaluation of a one-dimensional variational data assimilation model for the marine ecosystem along the West Antarctic Peninsula

Hyewon Heather Kim et al.

Correspondence to: Hyewon Heather Kim (hkim@whoi.edu)

User Manual for WAP-1D-VAR v1.0 Model

WAP-1D-VAR (v1.0) model is a variational data assimilation 1-D marine ecosystem model combining the forward (forward-in-time) model simulation and the backward (back-in-time) model simulation for model optimization. The backward model simulation is a tangent linear adjoint version of the forward model and is used in a variational adjoint method that optimizes model parameters to adjust model outputs towards observations.

The variational adjoint method requires four components for data assimilation that WAP-1D-VAR provides within: 1) a forward model simulated by physical forcing and initial (initial conditions and model parameter guesses) and boundary conditions; 2) a cost function to evaluate misfits between the forward model results and the assimilated observations; 3) a tangent linear adjoint version of the forward model to compute the gradient of the cost function with respect to model parameters; and 4) an optimization procedure (M1QN3 3.1) to determine the direction and the optimal step size by which the model parameters should be modified to reduce the cost function based on the cost function gradient from 3). These four components are iterated sequentially to determine a set of the adjusted model parameters until present criteria are satisfied (e.g., low gradients), which then serves as an optimal numerical solution (equations with the optimized model parameters) for the final model outputs.

The model development and validation for version v1.0 of the model for Palmer Station, Antarctica, is presented in: Kim, H. H., Luo, Y.-W., Ducklow, H. W., Schofield, O. M., Steinberg, D. K., Doney, S. C (2021). WAP-1D-VAR v1.0: Development and Evaluation of a One-Dimensional Variational Data Assimilation Model for the Marine Ecosystem Along the West Antarctic Peninsula, *Geoscientific Model Development, Accepted*. Kim *et al.* (2021) describes in more detail the variables and model equations used in the forward model, physical forcing data sets used for Palmer Station, and the resulting optimized model parameters. The manuscript also provides references for previous marine ecosystem model developments that led to the creation of WAP-1D-VAR (v1.0) model, creation of the tangent linear adjoint back-in-time model, and the application of the variational adjoint method. The model inputs files and code are available from Zenodo (https://zenodo.org/record/5041139).

1. Binary input files

Raw binary input files are provided for the model scenarios for the particular growth season for Palmer Station discussed in Kim *et al.* (2021). The input, output and source code files are deposited in Zenodo (https://zenodo.org/record/5041139). and should be download to the local directory of choice (e.g., `/Users/WAP1DVAR/forcing/Palmer`). Physical forcing and the assimilative observations are stored in files `.dat`, initial conditions in `.init`, bottom boundary conditions in `bbc.dat`, information for model grids in `grid.in`, and target errors (Equation 6, Kim *et al.* 2021) of each assimilative data type in `vars.dat`. The model can be used for other years or for other locations, but this would require the construction of new binary input files. The form of the files and variables included are as follows.

1.1. Model grid

Model grid information is defined in grid.in, in which nz_site is the number of model depths, dzt_site is the level thickness (m), nt_site is the number of time steps, delt_site is time step, and ntsout_site is the interval of model output (i.e., how many steps are averaged for one output value).

1.2. Physical forcing

Six different input files are prepared for physical forcing fields, including mixed layer depth, or MLD, (mld.dat), sea surface photosynthetically active radiation, or PAR, (PAR.dat), sea-ice concentration (ice.dat), water temperature (T.dat), vertical eddy diffusivity (Kv.dat), and vertical velocity (w.dat). The first column in these files contains model time, defined as the number of days after the default model start date (i.e., 1988-06-01), in which 8035 (days) indicates 2010-06-01. The second column in these files contains data for each of the physical forcing fields. PAR is specified on the time scale of 0.5 hours, while others are specified on the time scale of 1 day. Water temperature, sea-ice concentration, vertical eddy diffusivity, and vertical velocity are set up at every vertical depth level. Unit: PAR (W m^{-2}), MLD (m), sea-ice cover (fractional), water temperature (°C), vertical eddy diffusivity (m^2 s^{-1}), and vertical velocity (m s^{-1}).

1.3. Initial conditions

The initial conditions of ecological and biogeochemical model variables are given at every vertical depth level in files .init, in which the first column contains model depth and the second column contains data (see below **5**). The initial conditions are set up for C, N, and P components of bacterial biomass (BAc.init, BAn.init, BAp.init), microzooplankton biomass (PRTc.init, PRTn.init, PRTp.init), krill biomass (MZc.init, MZn.init, MZp.init), labile and semi-labile dissolved organic matter (LDOMc.init, LDOMn.init, LDOMp.init, SDOMc.init, SDOMn.init, SDOMp.init) particulate detritus (DETc.init, DETn.init, DETp.init), sediment trap data (STc.init, STn.init, STp.init), C, N, P, and Chl components of diatoms (DAc.init, DAn.init, DAp.init, DAchl.init) and cryptophytes (CRc.init, CRn.init, CRp.init, CRchl.init), and nutrients (NH4.init, NO3.init, PO4.init). If unicellular diazotroph and Trichodesmium compartments are modeled, their initial conditions should also be set up (UNc.init, UNn.init, UNp.init, UNchl.init, TRc.init, TRn.init, TRp.init, TRchl.init). Unit: mg m^{-3} for chlorophyll, mmol m^{-3} and mmol m^{-3} d^{-1} for the rest stock and flow variables.

1.4. Bottom boundary conditions

The bottom boundary conditions values are typically set to zero at the bottom of the model vertical grids except for NO_3, PO_4, semi-labile dissolved organic matter, as a column vector in files bbc.dat. The bottom boundary conditions are set up for C, N, and P components of bacterial biomass (BAc_bbc.dat, BAn_bbc.dat, BAp_bbc.dat), microzooplankton biomass (PRTc_bbc.dat, PRTn_bbc.dat, PRTp_bbc.dat), krill biomass (MZc_bbc.dat, MZn_bbc.dat, MZp_bbc.dat), labile and semi-labile dissolved organic matter (LDOMc_bbc.dat, LDOMn_bbc.dat, LDOMp_bbc.dat, SDOMc_bbc.dat, SDOMn_bbc.dat, SDOMp_bbc.dat) particulate detritus (DETc_bbc.dat, DETn_bbc.dat, DETp_bbc.dat), sediment trap data (STc_bbc.dat, STn_bbc.dat, STp_bbc.dat), C, N,

P, and Chl components of diatoms (DAc_bbc.dat, DAn_bbc.dat, DAp_bbc.dat, DAchl_bbc.dat) and cryptophytes (CRc_bbc.dat, CRn_bbc.dat, CRp_bbc.dat, CRchl_bbc.dat), and nutrients (NH4_bbc.dat, NO3_bbc.dat, PO4_bbc.dat). If unicellular diazotroph and Trichodesmium compartments are modeled, their boundary conditions should also be set up (UNc_bbc.dat, UNn_bbc.dat, UNp_bbc.dat, UNchl_bbc.dat, TRc_bbc.dat, TRn_bbc.dat, TRp_bbc.dat, TRchl_bbc.dat). Unit: mg m^{-3} for chlorophyll, mmol m^{-3} and mmol m^{-3} d^{-1} for the rest stock and flow variables.

1.5. Assimilative observations

The to-be-assimilated observations are set up in files .dat, in which the first column contains model date, the second column contains model depth, and the third column contains data, including diatom-specific chlorophyll (DA.dat), cryptophyte-specific chlorophyll (CR.dat), bulk phytoplankton nitrogen biomass (PHY.dat), primary production (PrPr.dat), bacterial production (BPr.dat), bacterial biomass (BAC.dat), microzooplankton carbon biomass (PRT.dat), krill carbon biomass (MZ.dat), nitrate (NO3.dat), phosphate (PO4.dat), particulate organic matter (POC.dat, PON.dat, POP.dat), and semi-dissolved organic matter (sDOC.dat, sDON.dat, sDOP.dat). Unit: mg m^{-3} for chlorophyll, mmol m^{-3} and mmol m^{-3} d^{-1} for the rest stock and flow variables.

1.6. Target errors

The target error of the assimilative data type (Equation 6 in Kim *et al.* 2021) is required to calculate the cost function that is minimized by optimization, but in a file vars.dat it needs to be set up as the inverse of the target error multiplied by the square root of the total number of the assimilated data types. For instance, the target error of NO_3 is 0.80 (Table 1 in Kim *et al.* 2021) but in vars.dat it should be 4.13 ($= 0.80^{-1} \times 11^{1/2}$). The order of the target error should be identical to the order within a character array fname_cost_suffix in a Fortran code cost.f90 in the local (WAP1DVAR/src/framework) directory.

2. Model compilation

The Fortran source codes (.f90) are compiled in the local (WAP1DVAR/src/eco) directory. To compile the source codes, we use a file Makefile that pre-processes source codes, specifies compiler and optimization options, and identifies any library dependencies. The packages used for the WAP-1D-VAR codes are a Fortran compiler (mpif90), OpenMPI with the Fortran buildings, and netCDF and HDF5 Fortran libraries. If compilation finished successfully then executables driver, test_adjoint, adjoint_driver, and hessian_driver should be generated in the local (WAP1DVAR/src/eco) directory. The name lists and I/O directories of the compiled programs driver and hessian_driver are defined in files input and input_hess, respectively.

3. Forward simulation

The forward model is run by:

```
%     ./driver <input
```

This prints out the cost function value for each data type and the total cost function value that represent misfits between the forward model results (prior to optimization) and the assimilated observations (i.e., the cost function based on the initial parameter guesses or p_0; $\partial J/\partial p_0$ or J_0 in Table 1 in Kim *et al.* 2021).

4. Adjoint simulation

Manuals for building an adjoint version of the forward model and for the quasi-Newton algorithm based optimizer M1QN3 are omitted here and can be found in Lawson *et al.* (1995), Giering (1999), Hascoët & Pascual (2004), and Gilbert & Lemaréchal (2006). In essence, we use an auto-differential software, TAPENADE v3.16 (https://team.inria.fr/ecuador/en/tapenade/, developed by Institut National de Recherche en Informatique et en Automatique, France) to conduct code-to-code translation to automatically generate the adjoint codes for the forward model. M1QN3 is a Fortran program that uses the gradient information calculated from the adjoint code to minimize the cost function based on a limited-memory quasi-Newton scheme. This optimization scheme is unconstrained and there is no penalty term to the cost function, in which parameters are optimized freely to always positive values, via log-transform and exponentiation before and after optimization, respectively.

After generating the adjoint codes (**4.1**), the adjoint simulation is conducted iteratively as a three-step process (**4.2**-**4.4**). There are mainly three ways to update the list of optimizable parameters for the next optimization cycle, as explained below.[1-3]

4.1. Generating adjoint codes

The latest version of TAPENADE (e.g., v3.16) should be downloaded and installed on a local machine. `README.html` provides a web installation guide. A program tapenade is run in the local (`tapenade3.16/bin`) directory as follows:

```
%    ./tapenade
```

This will open up a window generated by JAVA where a source Fortran code is uploaded and auto-differentiated to its corresponding adjoint code. Among many, a key source code whose adjoint code should be regenerated, whenever that key source code is modified, is a code `derivs_mod.f90` that defines many of its used variables from other source codes and has some variables that are input arguments from outside. Because the Tapenade program needs a 'stand-alone' source file, a template file `derivs_tap.f90` should be modified following a modified `derivs_mod.f90` in the way that text from below `!ecosystem model parameters` to the second last line should be copied from its corresponding text in `derivs_mod.f90`.

Once the Tapenade window opens up, 1) click 'add' to add `derivs_tap.f90`, 2) click 'parse', 3) select 'adjoint mode', and 4) click 'differentiate'. This generates an adjoint code `derivs_tap_b.f90` and a log file `derivs_tap_b.msg` that contains an error message if the adjoint code generation went wrong. Files `adBuffer.f` and `adStack.C` in the local (`tapenade3.16/ADFirstAidKit`) directory should also be updated in the local (`WAP1DVAR/src/framework`) directory.

Next, as similar to copying text from derivs_mod.f90 to derivs_tap.f90, the adjoint code text generated below !ecosystem model parameters to the second last line should be copied from derivs_tap.f90 to adderivs_mod.f90, which is a final adjoint code that is being used for model optimization processes specified below (**4.2**-**4.4**).

4.2. Cost function gradient

First, the gradient of the cost function with respect to each model parameter submitted to optimization (i.e., defined in a code eco_params.f90 in the local (eco) directory) is generated by:

```
%     ./test_adjoint <input
```

This writes a file gradient.csv in the local (WAP1DVAR/output) directory and prints out the results of the cost function gradients, in which only the dJ: d close to 1 (e.g., 1.0000…, 0.9999…) indicates that the adjoint model calculates the gradients correctly. By definition dJ: d is the ratio of the gradients directly generated from the adjoint model codes to those estimated numerically from $\Delta J/e^{\Delta p}$, where $e^{\Delta p} \approx \Delta p$ for an infinitely small Δp (e.g., a 10% change of a parameter, or Δp = 10%, leads to a total cost change equivalent to 10% of the corresponding gradient), so should be near 1, unless the cost function gradients with respect to the optimized model parameters are calculated, or the adjoint models are not properly constructed.

4.3. Parameter optimization

Second, the optimization procedure (M1QN3 3.1) is executed by:

```
%     ./adjoint_driver <input
```

This prints out a series of the results indicating the changes in the cost function when the model parameters are adjusted. At the end of the simulation this writes the value of each optimized parameter in a file eco_params.in.new in the local (WAP1DVAR/output) directory. Some parameter values resulted from optimization may be scientifically unrealistic, compared to the prescribed initial guesses. Such parameters are removed from the optimization list in a code eco_params.f90 and their values are set back to their default initial guess values in a code eco_common.f90 in the local (WAP1DVAR/src/eco) directory.[1]

4.4. Parameter uncertainties

Lastly, the uncertainties of the optimized parameters are calculated by:

```
%     ./hessian_driver <input_hess
```

This generates a file that contains the Hessian matrix called hessian.out in the local (WAP1DVAR/output) directory. The inverse of the Hessian matrix is calculated using MATLAB's built-in inv function. The square roots of the corresponding diagonal elements in the inversed Hessian matrix (σ_f) represent the relative errors of those optimized parameter values. The parameters with $\sigma_f > 0.5$ are removed from the optimization list in a code eco_params.f90 but their values are updated as the optimized values in a code eco_common.f90 in the local (WAP1DVAR/src/eco) directory.[2] The off-diagonal elements represent the cross-covariance between the corresponding two parameters. Highly correlated parameters (i.e., high off-diagonal

values) cannot be optimized simultaneously, so either of the two is removed from the optimization list in a code eco_params.f90.[3]

5. Binary output files

The raw binary output files (.out) and a final netCDF file eco_out.nc are saved in the local (/WAP1DVAR/output/Palmer) directory. The initial conditions files (.init) are generated as the model results at the end of the model simulation (on June 1 of the given year) so they can be used as the initial conditions for the simulation of the following year.

MATLAB's built-in ncdisp displays all the groups, dimensions, variable definitions, and all attributes in eco_out.nc. MATLAB's built-in ncread is used to load a model variable of interest:

```
>>  ncread('eco_out.nc', 'time')
>>  ncread('eco_out.nc', 'depth')
```

These load a column vector that contains model date, defined as above **1.2**, and the column vector that contains model depths.

```
>>  ncread('eco_out.nc', 'DAc')
```

This loads diatom carbon biomass in a $m \times n$ matrix, where m is the length of a variable depth and n is the length of a variable time.

6. Model results in Kim *et al.* (2021)

After model optimization is finished as demonstrated above, in Kim *et al.* (2021) we conduct Monte Carlo experiments to calculate the impact of the optimized parameter uncertainties on the model results where we create an ensemble of parameter sets (N = 1,000) by randomly sampling values within the uncertainty ranges of the constrained parameters and then perform a model simulation using each parameter set. This is a repetitive process and therefore automated in SLURM codes Palmer4.slurm, Palmer5.slurm, and Palmer6.slurm that should be run in this order.

Palmer4.slurm runs a Fortran code montecarlo.f90 that creates 1000 different perturbed parameter sets where arrays should be filled with error values calculated the inversed Hessian matrix, below !input errors from Hessian matrix. Palmer5.slurm runs forward simulations using each of these 1000 different parameter sets. Palmer6.slurm averages the model output variables among all of the generated output values from Palmer5.slurm, using a Matlab code palmer_monte_carlo.m that generates a file Palmer_eco_mc_out_final.nc. This is a final model output file and read by Matlab scripts for plotting, including wap1dvar_contour.m, wap1dvar_flow.m, wap1dvar_forcing.m, wap1dvar_points.m, and wap1dvar_taylor.m.

References

Giering R (1999) Tangent linear and Adjoint Model Compiler, Users Manual 1.4. http://autodiff.com/tamc

Gilbert, J. C., Lemaréchal, C. (2006). The module M1QN3. *INRIA Rep., version*, *3*, 21.

Hascoët L, Pascual V (2004) TAPENADE 2.1 User's Guide. http://hal.inria.fr/inria-00069880/en/

Lawson LM, Spitz YH, Hofmann EE, Long RB (1995) A data assimilation technique applied to a predator-prey model. *Bull Math Biol* 57:593–617