The Radial Solar Wind Variational Data Assimilation scheme- Documentation

Matthew Lang

March 2020

Contents

1 Introduction

This document outlines the Burger Radius Variational Data Assimilation scheme (BRaVDA) developed by Lang and Owens [19]. The code described by this documentation can be found online at:

https://github.com/SOJC6/BurgerRadiusDAExport.

The BRaVDA is a data assimilation (DA) scheme that merges the radial solar wind speed propagation model developed by [26] (see Section 3.1, with a variational data assimilation methodology based upon 4DVar (see 3.2.

The next section in this document gives a background into data assimilation and then the third section describes the mathematics of the solar wind propagation model and the data assimilation methodology. Full understanding of the mathematics is not required to run the data assimilation scheme and is provided for the interested reader, who wishes to understand the processes within the code. The fourth section outlines how to run the code.

2 Background into Data assimilation

Data assimilation is the study of combining prior knowledge from a model of a system with information contained in actual observations of that system in order to obtain an optimal estimate of the truth, including its uncertainty. DA methods can be used to 1) provide better model initial conditions for forecasting (e.g. [7, 10, 9]); 2) generate optimal evolution trajectories for the system to study important physical processes (e.g. [6, 21]); and 3) improve the model physics by studying model-observation misfits [18].

Variational data assimilation refers to the subset of DA methods used extensively in meteorological applications [27] that provide an optimal fit over the whole time window (the period of time over which the data assimilation is applied). Variational DA typically aims to correct the variables under consideration at the initial time by making use of all the data available over the entire time window. Sequential assimilation, on the other hand, provides an optimal fit at the end of the window by considering each observation sequentially each time a new observation becomes available. Variational data assimilation methods also tend to find the maximum of the posterior probability distribution, whereas sequential methods typically seek the mean of

the posterior probability distribution. For linear systems, this means that the sequential and variational approaches will lead to the same results at the end of the assimilation window. For non-linear systems, however, the posterior probability distribution may have multiple modes, which may lead to the variational approach getting stuck in a local maxima, as opposed to the global maxima of the system, leading to a poorer analysis. Conversely, multi-modal posterior distributions may lead to the mean of the system occurring in an area of low probability in the posterior distribution, leading to the sequential assimilation approach being sub-standard. In these cases, it is unclear what the 'optimal' estimate should be, hence there has been substantial efforts to develop Particle Filters [1, 7, 30, 29], which aim to estimate the full posterior probability distribution as opposed to a single 'optimal' estimate.

With the development of the 4DVar and adjoint model systems [12, 20], variational methodologies have become much more viable and efficient methods than optimal interpolation methods based on finite-difference methods to calculate the gradient of the cost function (and the Kalman filter methods that preceded them), particularly for applications within high-dimensional meteorological models with large quantities of observations. The 4DVar methodology is typically used to estimate the initial condition of a model given observations over a fixed time-window. However, the purpose of using a variational approach for the solar wind is to map information contained within observations back closer to the Sun, where we can then alter the inner boundary of the model and then re-compute the solar wind speed in the whole domain. Specifically, we wish to estimate the solar wind speed at all Carrington longitudes at the inner boundary ($30r_S$) using the observations at greater heliocentric distances (i.e., beyond $30r_S$ from the Sun, typically at Earth orbit, approximately $215r_S$).

3 The mathematics behind the Radial Solar Wind Variational Data Assimilation Scheme

3.1 Solar wind propagation model

In this study, we use the solar wind propagation model of **(author?)** [24], which maps the equatorial (i.e., two dimensional) solar wind speed over the

heliocentric domain from $30r_S$ to $215r_S$ from the Sun:

$$v_{i+1,j}(\phi) = v_{i,j} + \frac{\Delta r \Omega_{ROT}}{v_{i,j}} \left(\frac{v_{i,j+1} - v_{i,j}}{C\Delta\phi} \right) \tag{1}$$

where $v_{i,j}$ is the speed (in km/s) at radius, r_i (the i is the radius coordinate), and at Carrington longitude, ϕ_j (where j is the longitude coordinate). Using the same setup as [23], $\Delta r = 1r_S$ is the radial grid resolution (in km), $\Delta\phi = 2.81°$ is the latitudinal grid resolution, $C = \frac{2\pi}{180}$ is a constant representing the conversion factor from degrees to radians and $\Omega_{ROT} = \frac{2\pi}{25.38(86400)} s^{-1}$ is the solar rotational speed.

After this solution is obtained, an additional term, $v_{i,j}^{acc}$, is added to the $v_{i,j}$ to represent the acceleration of the solar wind in the domain considered, which is given by:

$$v_{i,j}^{acc} = \alpha v_{0,j}(\phi) \left(1 - e^{\frac{r_i}{r_H}} \right) \tag{2}$$

where $v_{0,j}$ is the solar wind speed at the inner-boundary and $\alpha = 0.15$ and $r_H = 50r_S$ are constants determined by [24].

Given the 2-dimensional nature of the solar wind model, in this study we must assume the ecliptic plane to be equatorial. In reality, the ecliptic is inclined by 7.25° to the heliographic equator. We note that a solar wind DA scheme in a full 3-dimensional solar wind model could relax this assumption.

3.2 The data assimilation scheme

In this section, we indicate the general definitions of data assimilation, followed by specific definitions when applied to the solar wind model. Following this, we explain the methodology behind the variational data assimilation proposed in this paper.

In DA, the variables within a numerical model, $\mathbf{x}$, are described by a N_x-dimensional state vector, which contains the values of the quantities of interest at all gridpoints, N. The variables within a state vector for meteorological applications could include the temperature field over Africa, the mean sea level pressure over the UK, wind speed/direction, etc., depending upon the purpose of the NWP model. The typical dimension of (number of variables within) the state vector in NWP models is of the order $\approx 10^9$ [8, 28].

For this application, the state vector is defined as a vector that contains the solar wind speed at each Carrington longitude, ϕ_j, in the model domain

for a given radius coordinate, i, and is written as:

$$\mathbf{v_i} = (v_{i,1}, v_{i,2}, \ldots, v_{i,N}) \tag{3}$$

where $i = 0, \ldots, 185$ correspond to radii coordinates from the Sun as $r_i = 30r_S, \ldots, 215r_S$ and $N = 128$.

For a state $\mathbf{x}$ (e.g., solar wind speed), an estimate of the initial conditions before any data assimilation is referred to as the prior (or background) state, denoted by $\mathbf{x^b}$. The prior state is generated using available prior information about the state. In meteorological applications, this typically comes from a previous forecast. The prior state is assumed to be a random perturbation away from the true state, such that:

$$\mathbf{x_0^t} = \mathbf{x^b} + \boldsymbol{\xi_0} \tag{4}$$

where $\mathbf{x_0^t}$ is an N_x-dimensional discretisation of the true initial state, and $\boldsymbol{\xi_0}$ represents the random error in the prior state.

For our solar wind model, the prior inner boundary condition, $\mathbf{v_0^b}$, is an inner boundary condition that we must provide (i.e. from a previous forecast of the inner boundary, such as from a previous coronal solution). The prior inner boundary condition is assumed to be a random perturbation from the true state, such that $\mathbf{v_0^b} \sim \mathcal{N}(\mathbf{v_0^t}, \mathbf{B})$, where $\mathbf{v_0^t}$ represents the true speed at the inner boundary and $\mathbf{B}$ represents the prior error covariance matrix (i.e. the covariance matrix of the errors at the inner boundary).

Observations within the time window provide information about the true state of the system. In order to merge the information contained by the observations with the state generated by the numerical model, it is necessary to define a function that maps from the state space to the observation space. This function is called the observation operator and is defined as:

$$\mathbf{y} = \mathcal{H}(\mathbf{x}) + \boldsymbol{\epsilon} \tag{5}$$

where $\mathcal{H} : \mathbb{R}^{N_x} \to \mathbb{R}^{N_y}$ maps the state into observation space, with $\mathbf{y}$ being an observation with the observation error given by $\boldsymbol{\epsilon}$.

Thus for the solar wind, the k^{th} vector of observations, $\mathbf{y_k}$, at radius r_k, are defined as:

$$\mathbf{y_k} = \mathcal{H}_k\left(\mathbf{v_{i_k}}\right) + \boldsymbol{\epsilon_k} \tag{6}$$

where $\mathcal{H}_k$ is the observation operator, a function that maps the model solar wind speed vector to the observation space (i.e. what the observation would

be for any given $\mathbf{v}$) and $\boldsymbol{\epsilon_k}$ is the random observation error, assumed to be normally distributed, $\boldsymbol{\epsilon_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R_k})$.

The numerical model, which approximates the dynamics of the system, is denoted by:

$$\mathbf{x_{i+1}} = f_i\left(\mathbf{x_i}\right) + \boldsymbol{\eta_i} \tag{7}$$

where $\mathbf{x_i}$ is the state vector at point i, the numerical model is represented by f_i (which here is the solar wind propagation model described in section 3.1), and $\boldsymbol{\eta_i}$ is an N_x-dimensional term representing the model error. The distribution of the model error is almost always unknown and may contain biases towards particular states (i.e. have non-zero mean) or be multi-modal, etc.

We now adopt the Strong Constraint approach and assume that the numerical model is perfect, i.e. contains no model error [11, 15]. In practice, this typically produces a poorer result than the weak constraint solution that allows for model error. But the weak constraint problem is much more complex as it is impossible to know precisely where the model is incorrect, due to missing physics, the effects of sub-grid processes etc., and hence very difficult to prescribe an accurate model error covariance matrix. In addition, including model error can lead to coupled equations that can result in different solutions, depending on which order the equations are solved [14, 18], leading to additional work being done to decouple them [5]. Therefore, as this is an introductory study to demonstrate the effectiveness of the variational approach, it is logical to start with the Strong Constraint approximation. The model evolution equation can then be rewritten as:

$$\mathbf{v_{i+1}} = f_i(\mathbf{v_i}) \tag{8}$$

$$= (f_{i,1}(\mathbf{v_i}), f_{i,2}(\mathbf{v_i}), \ldots, f_{i,N}(\mathbf{v_i})) \tag{9}$$

where

$$f_{i,j}(\mathbf{v_i}) = \begin{cases} v_{i,j} + \frac{\Delta r \Omega_{ROT}}{v_{i,j}} \left(\frac{v_{i,j+1} - v_{i,j}}{C\Delta\phi}\right) + \alpha v_{0,j} \left(e^{\frac{r_{i-1}}{r_H}} - e^{\frac{r_i}{r_H}}\right) & \text{if } r_i \neq 30 r_S \\ v_{i,j} + \frac{\Delta r \Omega_{ROT}}{v_{i,j}} \left(\frac{v_{i,j+1} - v_{i,j}}{C\Delta\phi}\right) + \alpha v_{0,j} \left(1 - e^{\frac{r_i}{r_H}}\right) & \text{Otherwise} \end{cases} \tag{10}$$

where r_i is the radial distance from the Sun at radius coordinate, $i = 0, \ldots, 185$. The model evolution equation at the inner boundary, at $30r_S$, adds the acceleration term onto the primary calculation for the speed at the next radial coordinate. However, for each subsequent radial coordinate, the

acceleration term from the previous radial point must be removed prior to the addition of the new acceleration to avoid an accumulation of the acceleration terms within the model.

Use of the Strong Constraint [17] allows the observation operator to map from the speed at the inner boundary, $\mathbf{v_0}$, to the observation location, such that:

$$\mathbf{y_k} = \mathcal{H}_k \left[f_{i_k - 1} \left(f_{i_k - 2} \left(\dots f_0 \left(\mathbf{v_0} \right) \dots \right) \right) \right] + \boldsymbol{\epsilon_k} \tag{11}$$

where k denotes the observation number and the i_k are a subset of the radial coordinates, i, and are the radial coordinates where the k^{th} observation occurs.

This allows a cost function, $\mathcal{J}$, to be written purely in terms of the inner boundary solar wind speed, such that:

$$\begin{aligned} \mathcal{J}(\mathbf{v_0}) = & \frac{1}{2} \left(\mathbf{v_0} - \mathbf{v_0^b} \right)^T \mathbf{B}^{-1} \left(\mathbf{v_0} - \mathbf{v_0^b} \right) \\ & + \frac{1}{2} \sum_{k=1}^{N_y} \left(\mathbf{y_k} - \mathcal{H}_k \left[f_{i_k - 1} \left(f_{i_k - 2} \left(\dots f_0 \left(\mathbf{v_0} \right) \dots \right) \right) \right] \right)^T \mathbf{R_k}^{-1} \left(\mathbf{y_k} - \mathcal{H}_k \left[f_{i_k - 1} \left(f_{i_k - 2} \left(\dots f_0 \left(\mathbf{v_0} \right) \dots \right) \right) \right] \right) \end{aligned} \tag{12}$$

where $\mathbf{v_0}$ is the state vector at the inner boundary (at $30 r_S$).

The cost function is the sum of the relative contributions of the errors present within the solar wind system. Its derivation is outlined in Appendix **??**. The first term in the cost function represents the prior errors, $\left(\mathbf{v_0} - \mathbf{v_0^b} \right)$, at the inner boundary weighted by the prior error covariance matrix, $\mathbf{B}^{-1}$. The second term represents the sum of the observation errors at each observed radial coordinate, $\left(\mathbf{y_k} - \mathcal{H}_k \left[f_{r_k - 1} \left(f_{r_k - 2} \left(\dots f_0 \left(\mathbf{v_0} \right) \dots \right) \right) \right] \right)$, and is weighted by the observation error covariance matrix $\mathbf{R}^{-1}$. Therefore, the optimal state, that minimises the errors in the system, is the state that minimises the cost function. The weighting by the inverse of the error covariance matrices means that if there is a high amount of certainty in, for example, the prior state compared to that of the observations, then $\mathbf{B}^{-1}$ will be much larger than $\mathbf{R}^{-1}$, meaning that the optimal state vector will have to move closer to the prior state to minimise the cost function, increasing the dominance of the prior state on the final analysis.

Obtaining the $\mathbf{v_0}$ that minimises the cost function is a non-trivial task. This can be done by evaluating the finite differences of $\mathcal{J}$ or by evaluating $\nabla_{\mathbf{v}} \mathcal{J}$ directly [3]. However, these methods are impractical for higher dimensional systems. In such situations, a more efficient method of calculating the

gradient is to use the adjoint method, a method of sensitivity analysis that efficiently computes the gradient of a function [13]. For the Strong Constraint approach, this involves using the method of Lagrange multipliers to generate a set of adjoint equations.

Firstly, we rewrite the cost function that we wish to minimise as:

$$\mathcal{J}(\mathbf{v_0}) = \frac{1}{2}\left(\mathbf{v_0} - \mathbf{v_0^b}\right)\mathbf{B}^{-1}\left(\mathbf{v_0} - \mathbf{v_0^b}\right)^T + \frac{1}{2}\sum_{k=1}^{N_y}\left(\mathbf{y_k} - \mathcal{H}_k(\mathbf{v_{i_k}})\right)\mathbf{R}^{-1}\left(\mathbf{y_k} - \mathcal{H}_k(\mathbf{v_{i_k}})\right)^T \tag{13}$$

subject to the (strong) constraint

$$\mathbf{v_{r+1}} = f_r(\mathbf{v_r}). \tag{14}$$

To minimise this cost function subject to the constraint, we must minimise the Lagrangian [2], $\mathcal{L}(\mathbf{v_r}, \boldsymbol{\lambda_r})$, which is defined by:

$$\mathcal{L}(\mathbf{v_r}, \boldsymbol{\lambda_r}) = \mathcal{J}(\mathbf{v_0}) + \sum_{i=0}^{N_r} \boldsymbol{\lambda}_{\mathbf{i+1}}^T(\mathbf{v_{i+1}} - f_r(\mathbf{v_i})). \tag{15}$$

where the $\boldsymbol{\lambda_i}$'s are the Lagrange multipliers.

By differentiating with respect to the v_r and the λ_r variables, it is possible to obtain a set of adjoint equations, given by:

$$\mathbf{v_{r+1}} = f_r(\mathbf{v_r}) \tag{16}$$

$$\boldsymbol{\lambda}_{\boldsymbol{N_r+1}} = \mathbf{0} \tag{17}$$

$$\boldsymbol{\lambda_r} = \begin{cases} \mathbf{F_r}^T\boldsymbol{\lambda_{r+1}} + \mathbf{H_k}^T\mathbf{R_k}^{-1}(\mathbf{y_k} - \mathcal{H}_k(\mathbf{v_r})) & \text{if } r = r_k \text{ is obs. radius} \\ \mathbf{F_r}^T\boldsymbol{\lambda_{r+1}} & \text{Otherwise} \end{cases} \tag{18}$$

$$\boldsymbol{\lambda_0} = \mathbf{F_0}^T\boldsymbol{\lambda_1} + \mathbf{B}^{-1}\left(\mathbf{v_0} - \mathbf{v_0^b}\right) \tag{19}$$

where $\mathbf{F_r}$ is given by the Jacobian of f_r, such that:

$$\mathbf{F_r} = \begin{pmatrix} \frac{\partial f_{r,1}}{\partial v_r(\phi_1)} & \frac{\partial f_{r,1}}{\partial v_r(\phi_2)} & \cdots & \frac{\partial f_{r,1}}{\partial v_r(\phi_N)} \\ \frac{\partial f_{r,2}}{\partial v_r(\phi_1)} & \frac{\partial f_{r,2}}{\partial v_r(\phi_2)} & \cdots & \frac{\partial f_{r,2}}{\partial v_r(\phi_N)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{r,N_r}}{\partial v_r(\phi_{N_r})} & \frac{\partial f_{r,N_r}}{\partial v_r(\phi_2)} & \cdots & \frac{\partial f_{r,N_r}}{\partial v_r(\phi_N)} \end{pmatrix} \tag{20}$$

and

$$\frac{\partial f_{r,i}}{\partial v_r(\phi_j)} = \begin{cases} 1 - \frac{\Delta r \Omega_{ROT}}{C \Delta \phi} \left(\frac{v_r(\phi_{i+1})}{v_r^2(\phi_i)} \right) & \text{if } \rho_r > 30 r_S \text{ and } j = i \\ \frac{\Delta r \Omega_{ROT}}{C \Delta \phi} \left(\frac{1}{v_r(\phi_i)} \right) & \text{if } \rho_r > 30 r_S \text{ and } j = i+1 \\ 1 - \frac{\Delta r \Omega_{ROT}}{C \Delta \phi} \left(\frac{v_0(\phi_{i+1})}{v_0^2(\phi_i)} \right) + \alpha \left(1 - e^{\frac{\rho_r}{\rho_H}} \right) & \text{if } \rho_r = 30 r_S \text{ and } j = i \\ \frac{\Delta r \Omega_{ROT}}{C \Delta \phi} \left(\frac{1}{v_r(\phi_i)} \right) + \alpha \left(1 - e^{\frac{\rho_r}{\rho_H}} \right) & \text{if } \rho_r = 30 r_S \text{ and } j = i+1 \end{cases} \tag{21}$$

The Lagrange multiplier at the inner boundary radius gives us the negative of the gradient at $\mathbf{v_0}$, the value we wish to find. This implies that:

$$\nabla_{\mathbf{v_0}} \mathcal{J}(\mathbf{v_0}) = -\boldsymbol{\lambda_0} = -\mathbf{F_0}^T \boldsymbol{\lambda_1} - \mathbf{B}^{-1} \left(\mathbf{v_0} - \mathbf{v_0^b} \right) \tag{22}$$

This gradient allows the use of a plethora of available gradient-based minimisation algorithms, from Newton's methods to steepest descent methods [4].

4 The code - documentation

Disclaimer: I am running this through Anaconda Spyder's python software

The Burger Radius Variational DA method is run by the code named *STEREOrunModelExport.py*, this should not need editting to be run. Inputs (such as parameters/directories etc.) are provided to the code via the *varSWDA.def* file which must be updated AND saved prior to running the *STEREOrunModelExport.py* file.

The code is capable of assimilating data from **01/01/2007 − 06/01/2020** (the latest MAS ensemble we currently provide, as I write this). Provided with this code are STEREO A observations from **01/01/2007 − 31/12/2018**, STEREO B observations from **01/01/2007 − 28/09/2014** and ACE observations from **01/01/2007 − 30/01/2020**.

This code uses **DA to update the inner boundary of the radial solar wind model** described in section 3.1, by propagating information contained within observations at near-Earth orbit radially inwards to the inner boundary. To do this, a cost function (see equation (13)) is specified that when it is minimised the (weighted) errors in the current estimate of the inner boundary of the solar wind and in the observations are optimally minimal. **We want to find the inner boundary condition that minimises this cost function** (see equation (13). The code does this by using

the adjoint of the radial solar wind model to find the gradient of a cost function with respect to the inner boundary condition (in essence propagating the observational information back to the inner boundary). Once we have the gradient of the cost function with respect to the inner boundary solar wind speed, we can use a wide range of (gradient-based) minimisation methods, which are typically faster and more accurate than non-gradient based minimisation methods. The code currently minimises the cost function with a $BFGS$ minimisation method ([16]), although this can be changed in the minimisation command in *STEREOrunModelExport.py* (Line $790 - 792$ as I write this, see python documentation for more information on this).

As we are updating the inner boundary of the solar wind model, **the code requires prior information about the inner boundary conditions**. In solar wind modelling, near-Sun conditions are typically fixed using empirical relations to the coronal magnetic field ([22] and [25]), which are themselves determined by extrapolation from the observed photospheric magnetic field. I have generated inner boundary conditions from the MAS model in this documentation (and provided them with this code). This code requires an ensemble of inner-boundary conditions, such that it can derive an estimate of the prior error covariance matrix, **B** (one of the 'weights' used in the cost function). I will use the term **MAS ensemble** frequently throughout this documentation to distinguish it as the ensemble used to generate the prior , however, there is no necessity for the ensemble that provides information about the inner boundary solar wind to be from MAS. Others, such as WSA can be used with no problem, as long as the number of longitude points specified in Line [15] of *varSWDA.def* is consistent with the number of longitude points in the inner boundary information provided by the other model. (I sound like someone trying to avoid product placement on the BBC "Other solar wind inner boundary conditions are available").

This code is capable of assimilating STEREO A, STEREO B and ACE data (it can be tricked into assimilating other sources, eg. Parker Probe, by setting one of the sources, eg. STERB, to the required source) over multiple 27-day periods. However, **it is currently only capable of assimilating 3 sources of observations at any one time**. It also **fixes the location of the satellite over each 27-day window at a constant position**, so issues may arise when assimilating data from a fast moving satellite, such as Parker Probe near perihelion.

The *STEREOrunModelExport.py* code calls upon many methodologies within the *STEREOmethodsExport.py* and they must be stored in the same

```
#[1] Output directory
<your Output Directory>
#[3] Initial date (inclusive) (in the form DDMMYYYY)
05062011
#[5] Number of 27-day periods (/solar Rotations) to run
1
#[7] Number of Longitude Points (will be spread evenly around 360deg)
128
#[9] Inner Radius (divided by rS, ie. a value of 30 yields an inner radius of 30rS)
30
#[11] Outer Radius (divided by rS, as above, 215 rS is Earth Radius)
215
#[13] Radial gridspace (deltaR, divided by rS as above)
1
#[15] Number of MAS ensemble members (default is 576 members)
576
#[17] File holding MJD's for start each Carrington rotation
Obs\CR_MJD\2039CRMJDstart.csv
#[19] Directory containing ensemble files used to generate B (folder contains 'line [15]'-member MAS ens,
files should be named vin_ensemble_CR####.dat)
MASens\
#[21] File containing longitudes of STEREO A for each Carrington Rotation
Obs\STEREOlocFiles\stereoaEarthDiff2020.lst
#[23] File containing longitudes of STEREO B for each Carrington Rotation
Obs\STEREOlocFiles\stereobEarthDiff2020.lst
#[25] File containing all STEREO A observations
Obs\ObservationFiles\stereoa_20070101_20201231.lst
#[27] File containing all STEREO B observations
Obs\ObservationFiles\stereob_20070101_20201231.lst
#[29] File containing all ACE observations
Obs\ObservationFiles\ace_2007_2020.lst
#[31] Which observations are to be assimilated (write a string of the letters A,B and/or C: A=STERA, B=STERB,
C=ACE) eg. ABC means assimilated all obs. streams, AC means assimilated STER A and ACE only
AB
#[33] Should R be generated as a percentage of the solar wind or as a constant supplied by the user (Write B
for background solar wind or C for constant standard deviation, then provide 3 constants seperated by a space
indicating the percentage of background (0,1] or constant>0 for STERA, STERB and ACE [even if not assim.])
B 0.1 0.1 0.1
#[35] Give localisation radius for B as a function of longitude (in degrees); 0= no localisation
15
#[37] Alpha (for use in solar wind propagation model, default=0.15)
0.15
#[39] r_H (for use in solar wind propagation model, default=50, value divided by rS as above)
50
```

Figure 1: The *varSWDA.def* file.

directory (or the appropriate path must be updated in the '`from STEREOmethodsExport import *`' line (line 17, as I write this) of *STEREOrunModelExport.py*). Similarly, the *varSWDA.def* code should be stored in the same location as *STEREOrunModelExport.py* or the line '`fvSWDA=open(currentDir+'\\varSWDA.dat','r')`' line (line 29, as I write this) should be updated with the required path. Similarly, lines $17 - 29$ of *varSWDA.def* all require paths specified from the directory *STEREOrunModelExport.py* is stored. If this cannot be done (or it is more convenient to insert path directories from the source/another directory, this should be editted within *STEREOrunModelExport.py* (lines 191 [17], 246 [19], 263 [21], 264 [23], 274 [25], 275 [27] and 276 [29] in *STEREOrunModelExport.py*, as I write this).

4.1 Outline of what code does

The basic outline of *STEREOrunModelExport.py* is as follows:

- Read in *varSWDA.def* and initialise variables

```
-----------------------------------------------------
Window no. 0

RMSE STEREO A MASMean = 196.6412822705237
RMSE STEREO A Forecast = 211.1335397136924
RMSE STEREO A Posterior= 123.01914971939041

RMSE STEREO B MASMean = 147.8133761715317
RMSE STEREO B Forecast = 165.44469764602076
RMSE STEREO B Posterior= 78.62053978228222

RMSE ACE MASMean = 94.93188430003832
RMSE ACE Forecast = 104.99995970255986
RMSE ACE Posterior= 99.59088467449632
-----------------------------------------------------
```

Figure 2: RMSEs from suggested test to make sure code is working.

- Read in and calculate MJD start-times and mid-points for all 27-day windows. Write these into files (if they don't already exist).
- Make initial ensembles covering the 27-day windows from the ensembles for each Carrington Rotation the 27-day windows cover. Write these to files if they do not already exist. (This step may introduce discontinuities where one CR ends and another starts if it's in the middle of the 27-day period, however, I have not seen any adverse effects due to this).
- Locations of the STEREO satellites is specified for each 27-day period (chosen from the nearest location information to the MJD Midpoint of the window). Observation files locations are read in from *varSWDA.def*.
- For each 27-day period:
  - The localisation radius is read from *varSWDA.def* and the mean of the prior and prior error covariance matrix, $\mathbf{B}$ are estimated from the MAS ensemble.
  - The prior solar wind speed is generated as a random perturbation (distributed by $\mathcal{N}(0, \mathbf{B})$) away from the mean of the MAS ensemble.
  - The prior and MASMean solar wind speeds are propagated out into the heliosphere using the radial propagation model described in Section 3.1.

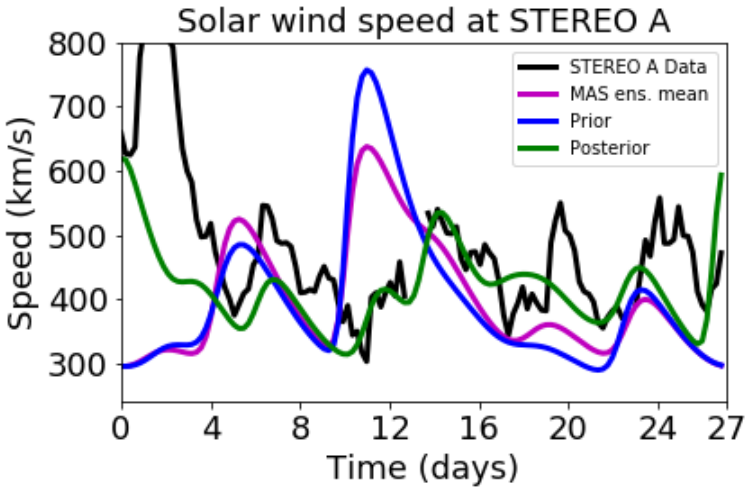

Figure 3: The solar wind speeds at STEREO A for test to ensure code is working

- The observations are read in for STEREO A, STEREO B and ACE and any unphysical/missing values are removed from the observation variables (allowing the number of observations to be calculated for each source).
- Depending upon which observations the user specified that should be assimilated, the radius at which the observations occur is set (STEREO A and B are currently set to $215r_S$ and ACE is set to $213r_S$, a future update could be to define these quantaties in *varSWDA.def*). The number of observations at each radius is also set here.
- Using the values generated in the above step, the generic observation vector, observation operator and observation error covariance matrices are initialised.
- The observation operators are generated for each observation source based upon their longitudinal locations relative to Earth.
- Depending upon how the user specified the observation error covariance matrices be generated in Line [33] of *varSWDA.def*, the observation standard deviations are generated here and then placed into a diagonal observation error covariance matrix for each ob-

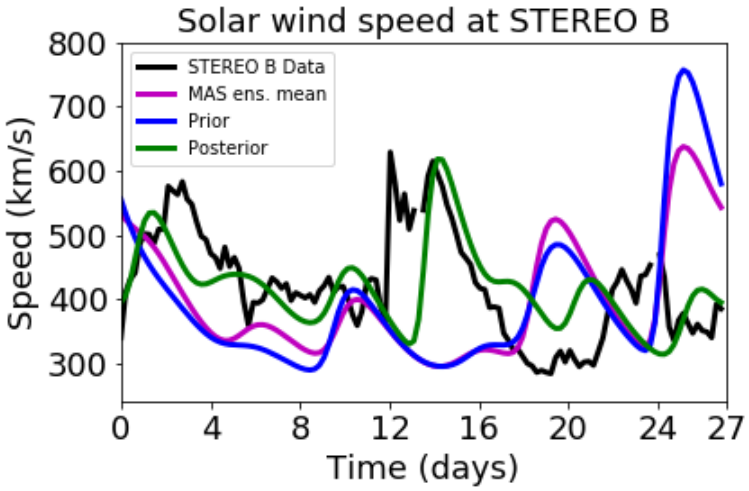

Figure 4: The solar wind speeds at STEREO B for test to ensure code is working

servation source. (A simple code for generating **R** is commented here, if the user wishes to specify another form of **R**, although this is not yet a useable option in *varSWDA.def*).

- Depending on which observations the user has chosen to assimilate, the required observations, observation operators and observation error covariance matrices are placed into the generic variables for use in the data assimilation.
- The Data Assimilation starts!!! Variables are initialised for use in the data assimilation. The number of iterations variable has been set to 1 as the cost function does not need to be written in incremental form for the radial propagation model, hence multiple outer loops are not required as we can minimise the full 'non-linear' cost function directly. This has been left in, in case the user decides to update the cost function to incremental form and wishes to easily increase the number of iterations (although, I note that I have not tested the code with multiple iterations in a long time, so bugs could be present here).
- The prior state is propagated out into the heliosphere using the radial propagation model described in Section 3.1.

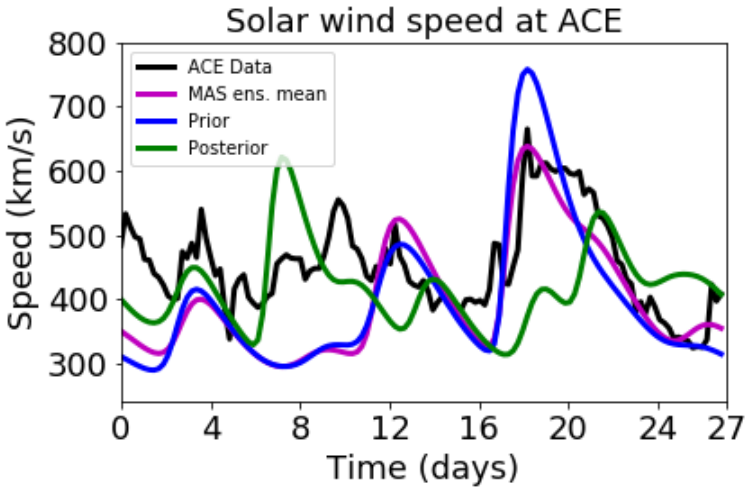

Figure 5: The solar wind speeds at ACE for test to ensure code is working

- The cost function before DA is calculated here and printed for the user.
- The cost function is minimised using the optimise.minimise function in python that takes in functions to calculate the cost function and the gradient of the cost function. This step normally takes a while for the minimisation to complete (as there are repeated runs of the model and adjoint functions out into the domain and back to the inner boundary). This can be sped up by increasing the `gtol` value, but will reduce the accuracy of the minimisation and hence produce a poorer DA analysis.
- The results of the minimisation are placed into the required variables, propagated out into the heliosphere using the model described in Section 3.1, then the cost function is calculated and printed after the DA has been performed.
- Values from the posterior (also called DA analysis), prior, MASMean and observations are collated into variables for plotting for each observation source.
- RMSEs are calculated for this 27-day window for the prior, MASMean and posterior at each observation source.
- The posterior, prior, MASMean and observations are plotted for

each observation source.

  - The RMSEs calculated two steps above are printed and written into the output file for future use
  - The prior, posterior, MASMean and observations are all written into a file for future use

- The RMSEs are plotted for each 27-day window and then written into a file for future use
- Phew...we're done...

4.2 *Skip to this if you're only interested in running the code without worrying about the mechanics of what's going on behind the scenes

This section will describe the variables needed in the *varSWDA.def* definitions file, the input files provided in the code directories and the output files generated by the *STEREOrunModelExport.py* code.

The basic steps needed to run the code are as follows:

1. Place code in a suitable directory
2. Update the *varSWDA.def* to the variables you need and SAVE IT!
3. Run the code *STEREOrunModelExport.py* in python (should work from command line, but at present I have not tested this)

The code *should* work straight out of the box after you've downloaded it.

As a test, before changing anything in the codes, I recommend that the user update *varSWDA.def* (if this is not the default) accordingly:

- Set the output directory to a relevant directory on their machine in Line [1] (I can't help with this)
- Set the initial date to 05062011 in Line [3]
- Set the number of windows to 1 in Line [5]
- Set to 'AB' (without quotes) in Line [31]

- Set to 'B 0.1 0.1 0.1' (without quotes) in Line [33]
- Set localisation radius to 15 in Line [35]

Leave everything else as the defaults for now and should be identical to Figure 1, and the *STEREOrunModelExport.py* should run.

If the code is working as it should produce the same results (or similar, depending on random number generator used in your python program) as shown in Figures 2, 3, 4 and 5.

4.3 The *varSWDA.def* definitions file

The *varSWDA.def* file (see Figure 1 for an example file) is the definitions file that provides parameters to the run file to perform solar wind data assimilation experiments. The order of the lines is VERY important as it dictates which variable each value is read into. The **order of the lines in *varSWDA.def* must not be changed** without first updating the requisite code lines within *STEREOrunModelExport.py*. In this part of the document, I will go through all the lines and explain what each one requires.

The even lines The even lines are all comments to aid the user input the correct parameters for the line below and are not used in the run script. If they are being read into the run script, then an error is occurring, check that the new-line is present at the end of all the comments. (The first line is classed as Line 0, in line with python's numbering system)

Line 1 The output directory to be set by the user, it is where all the outputs from the code will be stored (obviously). **This must be set by the user before any DA is possible**.

Line 3 The start date of the first time-window of the data assimilation. It should be written in the form Day/Month/Year in the format DDMMYYYY, eg. 01052009. The range of dates of the STEREO observations provided range from 01/01/2007-31/08/2019 (STEREO B data ends on 28/09/2014), so these will not be assimilated for dates outside this range.

Line 5 The number of 27 day periods that the DA will be run for consecutively.

Line 7 The number of longitude points. This is currently set to 128 as it must match the number of longitude points in the MAS ensemble (or whichever model is used to generate an ensemble to estimate $\mathbf{B}$, the background error covariance matrix. These longitude points must be spread evenly around 360° of the Solar Rotation.

Line 9 The inner boundary's radius as a multiple of the solar radii (default is $30r_S$). Must be less than the innermost observation of the solar wind.

Line 11 The inner boundary's radius as a multiple of the solar radii (default is $215r_S \approx$ Earth's orbit). Must be greater than the outermost observation of the solar wind.

Line 13 The radial gridspace as a multiple of the solar radii. Care must be take that the radial grid-space and the longitundal grid-space preserve the CFL criteria.

Line 15 This should specify the number of ensemble members in the MAS ensemble used to estimate the background error covariance matrix $\mathbf{B}$.

Line 17 This should specify the path to a file (starting from the code directory) holding the start-dates of each Carrington Rotation, a file is included in the directories that has these from $CR2039 - 2206$.

Line 19 The should specify the path to a directory (Starting at the code directory) that contains the files that define the MAS ensemble used to estimate the background error covariance matrix, $\mathbf{B}$. Files in this directory should be named vin_ensemble_CR****.dat (this can be changed in Lines $355 - 356$ (as I write this) of *STEREOmethodsExport.py* in the **makeVinEns27d** definition, (line 358 (as I write this) should **not** be changed however)). It is important that this is an accurate representation of the solar wind behaviours (easier said than done), as the relationship/magnitude of $\mathbf{B}$ with respect to $\mathbf{R}$, the observation error covariance matrix, determines how much we 'trust' the observation in comparison to the prior solar wind and determines how far the posterior will move from the prior towards the observed states.

Line 21 This should specify the path to a file (starting at the code directory) should contain the (longitudinal) positions of STEREO A for all Carrington Rotations to be used in the assimilation.

Line 23 This should specify the path to a file should contain the (longitudinal) positions of STEREO B for all 27-day Rotations to be used in the assimilation.

Line 25 This should specify the path to a file (starting at the code directory) that contains the observations recorded by STEREO A at all 27-day Rotations to be used in the assimilation.

Line 27 This should specify the path to a file (starting at the code directory) that contains the observations recorded by STEREO B at all 27-day Rotations to be used in the assimilation.

Line 29 This should specify the path to a file (starting at the code directory) that contains the observations recorded by ACE at all 27-day Rotations to be used in the assimilation.

Line 31 This line denotes which observation streams are to be assimilated into the model. STEREO A is denoted by 'A' (without quotes), STEREO B is denoted by 'B' (without quotes) and ACE is denoted by 'C' (without quotes). This input accepts a string of A, B and C (without deviations, hesitations or repetitions).

Line 33 This defines the observation error covariance matrix $\mathbf{R}$ as either a percentage of the background error covariance matrix, $\mathbf{B}$, or a constant value, specified by the user. This entry is made up of 4 values separated by a space. The first value is either a 'B' or a 'C' (without the quotes). 'B' means that $\mathbf{R}$ will be specified as a percentage of the background error covariance matrix to be specified by the next 3 values (for STEREO A, B and ACE respectively). 'C' means that a constant value will be used for all observations. The constant is specified for each observation type by the next 3 values (for STEREO A, B and ACE respectively). As specified above, the observation error covariance matrix is not the same as measurement error covariance matrix, which is the error inherent within the operatus taken in the meaasurement of the solar wind. Observation error contains the measurement error, but also representation error that arises as a result of representing/modelling our observation poorly (either due to computational constraints or sub-grid processes). An example of representivity error in this code is that the model is $2D$ on a plane that goes through the

Earth-Sun orbital line (the ecliptic plane), whereas our observations at STEREO A and B are frequently at different longitudes to the ecliptic, hence an error is introduced to the 'modelled' observation as a result of assuming they are constantly on the Earth-Sun line.

Line 35 This specifies the localisation radius for the estimation of the background error covariance, $\mathbf{B}$. Localisation reduces the impact of spurious correlations that arise in the estimated $\mathbf{B}$ as a result of using an ensemble to estimate the true prior error uncertainty. Localisation limits the covariances to a smaller longitude band and makes $\mathbf{B}$ 'more diagonal', too large a number will still contain spurious correlations, too small a number will restrict the impact of observations away from the observation location. Fewer (independent) ensemble members mean that localisation is increasingly needed.

Line 37 The α value to be used in (2).

Line 39 The r_H value to be used in (2).

4.4 Input files provided in varSWDAExport

All the files that are included in the varSWDAExport directory and are necessary for the smooth running of the code:

STEREOrunModelExport.py: The python file that needs to be run to run the variational DA method on the solar wind model. This should be possible to do via a shell script, but has not been tested yet.

STEREOmethodsExport.py: This python file contains all the methods/functions used in *STEREOrunModelExport.py* above. These functions include the solar wind model, its Tangent Linear Model (TLM), its adjoint and a function to generate the gradient of the cost function at the initial timestep.

varSWDA.def: This file is a definitions file used for setting parameters in the solar wind DA experiments, without having to directly interact with the main code (outside of running *STEREOrunModelExport.py*). This file is copied into the output directory when simulations are run, so that the specific definitions file for each output directory is available. The line numbers

are specific and interact with the code, henceforth the line numbers of each parameter must not be changed without altering their respective values in *STEREOrunModelExport.py*. Further details on this file will be given in a later section.

/MASens/vin_ensemble_CR**: These files contain 576-member ensemble for Carrington Rotation ****. These are provided for Carrington Rotations 2038-2225 (22/12/2005-06/01/2020).

/Obs/CR_MJD/2039CRMJDstart.csv: This file contains the start time of Carrington Rotations 2039-2228 (22/12/2005-01/03/2020). Used in Line 17 of *varSWDA.def*.

/Obs/CR_MJD/2039_2300CRMJDstart.csv: This file contains the start time of Carrington Rotations 2039-2300 (22/12/2005-06/27/2025). Used in Line 17 of *varSWDA.def*. Future Carrington Rotations (greater than 2228) are estimated as 27 day rotations.

/Obs/ObservationFiles/stereoa_2007_2020.lst: This file contains hourly averaged means for STEREOA data between 2007-2020. Used in Line 25 of *varSWDA.def*. Data was downloaded from OMNIweb and the columns are in the format: $|Year|DoY|Hour|$STERA obs (in km/s)|.

/Obs/ObservationFiles/stereob_2007_2020.lst: This file contains hourly averaged means for STEREO B solar wind speeds between 2007-2020. Used in Line 27 of *varSWDA.def*. Data was downloaded from OMNIweb and the columns are in the format: $|Year|DoY|Hour|$STERB obs (in km/s)|.

/Obs/ObservationFiles/ace_2007_2020.lst: This file contains hourly averaged means for ACE data between 2007-2020. Used in Line 29 in *varSWDA.def*. Data was downloaded from OMNIweb and the columns are in the format: $|Year|DoY|Hour|$ACEobs (in km/s)|.

/Obs/STEREOlocFiles/stereoaEarthDiff.lst: This file contains daily averaged STEREO A positions relative to ACE/Earth from 22/01/2007-31/12/2017. These files were generated from *https://omniweb.gsfc.nasa.gov/coho/helios/heli.html*. Used to generate observation operator and in Line 21 of *varSWDA.def*.

/Obs/STEREOlocFiles/stereobEarthDiff.lst: This file contains daily averaged STEREO B positions relative to ACE/Earth from 22/01/2007-31/12/2017. These files were generated from *https://omniweb.gsfc.nasa.gov/coho/helios/heli.html*. Used to generate observation operator and in Line 23 of *varSWDA.def*.

4.5 Output files provided in varSWDAExport

Output directory is specified on line [1] of the *varSWDA.def* file.

outputDir/varSWDA.def: This file is the definitions file used for setting parameters in the solar wind DA experiments, and is copied from the inputs.

outputDir/MJDfiles/MJDMidPoints_***_#####.txt**: This file contains the mid points of 27-day windows starting at MJD ***** and ending at 27-day window starting at MJD #####. This is used to place the STEREO observations for each time window.

outputDir/MJDfiles/MJD_***.txt**: This file contains the MJD's corresponding to the longitude coordinates in the time window starting at MJD *****. There will be multiple files generated here if multiple 27-day windows are used.

outputDir/MJDfiles/MJDstart_*_nWindows_%/vin_ensemble_MJDstart###.txt**: This file is the MAS ensemble generated for use with time window ###. The directory is set up for each new run (if the directory does not already exist), for the MJD start time of the initial window at *** and with % number of 27-day windows

outputDir/ACE/ACE_MJDstart***.txt**: This file is the observations of ACE that were used in the (27-day) data assimilation window starting with MJD *****. There will be multiple files in this directory if multiple windows are assimilated.

outputDir/STERA/STERA_MJDstart***.txt**: This file is the observations of STEREO A that were used in the (27-day) data assimilation

window starting with MJD *****. There will be multiple files in this directory if multiple windows are assimilated.

outputDir/STERB/STERB_MJDstart***.txt**: This file is the observations of STEREO B that were used in the (27-day) data assimilation window starting with MJD *****. There will be multiple files in this directory if multiple windows are assimilated.

outputDir/meanMAS/meanMAS_MJDstart***.txt**: This file is the mean of the MAS ensemble for each longitude point in the (27-day) data assimilation window starting with MJD *****. Longitude 0 is assumed to be the Earth-Sun line. There will be multiple files in this directory if multiple windows are assimilated. The entries in this file are the solar wind speed saved (using savetxt) in an array ordered [radCoord,lonCoord] and can be extracted into a $nRad \times nLon$ array using loadtxt(outputDir/meanMAS/meanMAS_MJDstart*****.txt) in python.

outputDir/prior/prior_MJDstart***.txt**: This file is the prior solar wind speed, selected as a random perturbation from the MAS Mean for each longitude point in the (27-day) data assimilation window starting with MJD *****. The random perturbation is normally distributed, unbiased (has mean zero) and has a covariance matrix calculated from the MAS ensemble (using the equation $Cov = \frac{1}{M-1}\sum_{m=1}^{M}\left[\left(\mathbf{x_m}-\overline{\mathbf{x}}\right)\left(\mathbf{x_m}-\overline{\mathbf{x}}\right)^T\right]$, where $\overline{\mathbf{x}} = \frac{1}{M}\sum_{m=1}^{M}\left[\mathbf{x_m}\right]$ is the MAS ensemble mean). Longitude 0 is assumed to be the Earth-Sun line and is the first value in this file. There will be multiple files in this directory if multiple windows are assimilated. The entries in this file are the solar wind speed saved (using savetxt) in an array ordered [radCoord,lonCoord] and can be extracted into a $nRad \times nLon$ array using loadtxt(outputDir/prior/prior_MJDstart*****.txt) in python.

outputDir/posterior/post_MJDstart***.txt**: This file is the posterior solar wind speed, and is solar wind speed estimate after DA has been performed. Longitude 0 is assumed to be the Earth-Sun line and is the first value in this file. There will be multiple files in this directory if multiple windows are assimilated. The entries in this file are the solar wind speed saved (using savetxt) in an array ordered [radCoord,lonCoord] and can be extracted into a $nRad \times nLon$ array using

loadtxt(outputDir/posterior/post_MJDstart*****.txt) in python.

outputDir/RMSE/RMSE_MASMean< *sat* >**_MJD*****_#####.txt**: This file contains the RMSE values over each 27-day assimilation window (between MJD***** and MJD#####) for the MASMean at satellite < *sat* > (where < *sat* >=STERA, STERB or ACE)

outputDir/RMSE/RMSE_prior< *sat* >**_MJD*****_#####.txt**: This file contains the RMSE values over each 27-day assimilation window (between MJD***** and MJD#####) for the prior state (before DA) at satellite location < *sat* > (where < *sat* >=STERA, STERB or ACE)

outputDir/RMSE/RMSE_post< *sat* >**_MJD*****_#####.txt**: This file contains the RMSE values over each 27-day assimilation window (between MJD***** and MJD#####) for the posterior state (DA analysis) at satellite location < *sat* > (where < *sat* >=STERA, STERB or ACE).

outputDir/Plots/RMSE_< *sat* >**.png**: These plots show the RMSEs of the prior, posterior and MAS Mean at satellite location < *sat* > (where < *sat* >=STERA, STERB or ACE) over each 27-day period the assimilation is performed over. The RMSEs are plotted against the MJD mid-point of each 27-day time window.

outputDir/Plots/speedsOverCR/< *sat* >**/SWspeed**< *sat* >**_MJDstart*****.png**: These plots show the solar wind speed at the satellite location < *sat* > (where < *sat* >=STERA, STERB or ACE) for the MJD window starting at MJD*****. The solar wind speed is plotted (forwards in time) against the number of days that have elapsed in the 27-day window.

5 Citation

If you wish to cite this model, please use the cite of:
Lang, M., Owens, M. J. (2019). A Variational Approach to Data Assimilation in the Solar Wind. *Space Weather*, 17(1), 59-83.

6 Problems or errors?

Are there any errors in this documentation or bugs in the code that I haven't found? Do you have ideas for ways to improve this code? Contact me at *matthew.lang@reading.ac.uk*.

References

[1] M. Ades and P. J. van Leeuwen. An exploration of the equivalent weights particle filter. *Quarterly Journal of the Royal Meteorological Society*, 139(672):820–840, apr 2013.

[2] G B Arfken, H J Weber, and F E Harris. *Mathematical methods for physicists: A comprehensive guide*. Academic press, 2011.

[3] R Bannister. Elementary 4D-Var. *DARC Technical Report No.2*, 2007.

[4] Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming Theory and Algorithms*. John Wiley & Sons, 2013.

[5] A F Bennett. *Inverse Methods in Physical Oceanography*. Arnold and Caroline Rose Monograph Series of the American So. Cambridge University Press, 1992.

[6] Grégoire Broquet, Frédéric Chevallier, Peter Rayner, Céline Aulagnier, Isabelle Pison, Michel Ramonet, Martina Schmidt, Alex T. Vermeulen, and Philippe Ciais. A European summertime CO 2 biogenic flux inversion at mesoscale from continuous in situ mixing ratio measurements. *Journal of Geophysical Research: Atmospheres*, 116(D23), dec 2011.

[7] P. A. Browne and P. J. van Leeuwen. Twin experiments with the equivalent weights particle filter and HadCM3. *Quarterly Journal of the Royal Meteorological Society*, 141(693):3399–3414, oct 2015.

[8] P A Browne and S Wilson. A simple method for integrating a complex model into an ensemble data assimilation system using MPI. *Environmental Modelling & Software*, 68:122–128, 2015.

[9] A. M. Clayton, A. C. Lorenc, and D. M. Barker. Operational implementation of a hybrid ensemble/4D-Var global data assimilation system at the Met Office. *Quarterly Journal of the Royal Meteorological Society*, 139(675):1445–1461, jul 2013.

[10] D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C. M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B.

Healy, H. Hersbach, E. V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A. P. McNally, B. M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thépaut, and F. Vitart. The ERA-Interim reanalysis: configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 137(656):553–597, apr 2011.

[11] Emanuele Di Lorenzo, Andrew M. Moore, Hernan G. Arango, Bruce D. Cornuelle, Arthur J. Miller, Brian Powell, Boon S. Chua, and Andrew F. Bennett. Weak and strong constraint data assimilation in the inverse Regional Ocean Modeling System (ROMS): Development and application for a baroclinic coastal upwelling system. *Ocean Modelling*, 16(3-4):160–187, jan 2007.

[12] F X L Dimet and O Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A*, 38A(2):97–110, 1986.

[13] R M Errico. What is an adjoint model? *Bulletin of the American Meteorological Society*, 78(11):2577–2591, 1997.

[14] G Evensen, D P Dee, and J Schröter. Parameter estimation in dynamical models. In *Ocean Modeling and Parameterization*, pages 373–398. Springer, 1998.

[15] M. Fisher, M. Leutbecher, and G. A. Kelly. On the equivalence between Kalman smoothing and weak-constraint four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 131(613):3235–3246, oct 2005.

[16] Roger Fletcher. *Practical methods of optimization*. Wiley, 1987.

[17] K. E. Howes, A. M. Fowler, and A. S. Lawless. Accounting for model error in strong-constraint 4D-Var data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(704):1227–1240, apr 2017.

[18] M Lang, P J van Leeuwen, and P A Browne. A systematic method of parameterisation estimation using data assimilation. *Tellus A*, 68(0), 2016.

[19] Matthew Lang and Mathew J Owens. A variational approach to data assimilation in the solar wind. *Space Weather*, 17(1):59–83, 2019.

[20] A C Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194, 1986.

[21] Natasha Macbean, Philippe Peylin, Frédéric Chevallier, Marko Scholze, and Gregor Schürmann. Consistent assimilation of multiple data streams in a carbon cycle data assimilation system. *Geosci. Model Dev*, 9:3569–3588, 2016.

[22] S. L. McGregor, W. J. Hughes, C. N. Arge, M. J. Owens, and D. Odstrcil. The distribution of solar wind speeds during solar minimum: Calibration for numerical solar wind modeling constraints on the source of the slow solar wind. *Journal of Geophysical Research: Space Physics*, 116(A3), mar 2011.

[23] Mathew J. Owens and Pete Riley. Probabilistic Solar Wind Forecasting Using Large Ensembles of Near-Sun Conditions With a Simple One-Dimensional "Upwind" Scheme. *Space Weather*, 15(11):1461–1474, nov 2017.

[24] P. Riley and R. Lionello. Mapping Solar Wind Streams from the Sun to 1 AU: A Comparison of Techniques. *Solar Physics*, 270(2):575–592, jun 2011.

[25] Pete Riley, Jon A. Linker, and C. Nick Arge. On the role played by magnetic expansion factor in the prediction of solar wind speed. *Space Weather*, 13(3):154–169, mar 2015.

[26] Pete Riley, Jon A. Linker, R. Lionello, and Z. Mikic. Corotating interaction regions during the recent solar minimum: The power and limitations of global MHD modeling. *Journal of Atmospheric and Solar-Terrestrial Physics*, 83:1–10, jul 2012.

[27] Yoshikazu Sasaki. Numerical Variational Analysis formulated under the constraints as determined by longwave equations and a Low-pass filter. *Monthly Weather Review*, 98(12):884–898, dec 1970.

[28] Y Trémolet. Accounting for an imperfect model in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 132(621):2483–2504, 2006.

[29] P J van Leeuwen. Particle filters for the geosciences. *Advanced Data Assimilation for Geosciences: Lecture Notes of the Les Houches School of Physics: Special Issue, June 2012*, page 291, 2014.

[30] M Zhu, P J van Leeuwen, and J Amezcua. Implicit Equal-Weights Particle Filter. *Quarterly Journal of the Royal Meteorological Society*, 2016.