Scientific workflow design: theoretical and practical issues
Terpstra, F.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 6

Data Assimilation

6.1 Introduction

In the upcoming chapters we will study workflows in practice with the help of two case studies. One case study deals with the prediction of bird migration, while the other is concerned with the prediction of road traffic. Both of these case studies use the technique of data assimilation to enable continuous predictions. In this chapter we will describe the origins of this technique. An overview of the essentials of data assimilation is given. Both the history and essentials of data assimilation are derived mainly from the books of Kalnay[83] and Daley[56]. The chapter is concluded with a discussion on the suitability of current toolkits for use in e-Science.

6.2 Weather Prediction

The term data assimilation originated in the field of weather prediction. Numerical weather prediction was first made a practical proposition in the mid 1950’s by the realization in 1904 by Bjerkness[44] that there were no simple causal relationships that relate the state of the atmosphere at one instant of time to that at another. Thus he defined weather prediction as nothing less than the integration of the equations of motion of the atmosphere. This was followed in 1922 by a practical method of performing this integration numerically. The predictions using this method at the time failed because of the inadequacies of the available observations. In 1950 aided by adequate observations and - of more interest to us - one of the first computers, the ENIAC, Charney, Fjörtoft and von Neuman computed the first successful one day numerical weather forecast[50]. One of the largest challenge for achieving a good prediction was the estimation of the initial state of the predictive model. While the observations were “adequate” they did not match the grid points in the model. Furthermore there were many more grid points in the model than there were observations. Finally observations
CHAPTER 6. DATA ASSIMILATION

were not equally distributed, with many more observations in Eurasia and North America than in other areas. For the first experiments the values for these grid points were interpolated manually from the available observations. These grid points were then manually digitized, a very labor intensive process, that also lacked objectivity because human interpretation was involved in the manual interpolation. Soon weather prediction switched to a process where interpolation was done by computer. To compensate for the lack of observations a first guess of the state of the atmosphere was needed. At first this was based on climatological data, later upon short range forecasts. At each prediction cycle this state estimate was adjusted based on the observations available. The prediction of the model could be used as a first guess of the state for the next cycle. The resulting process as illustrated in figure 6.1 is now known as data assimilation. In 1997 data assimilation was defined by Talagrand[116] as: “Assimilation of meteorological or oceanographical observations can be described as the process through which all available information is used in order to estimate as accurately as possible the state of the atmospheric or oceanic flow. The available information essentially consists of the observations proper, and of the physical laws that govern the evolution of the flow. The latter are available in practice under the form of a numerical model. The existing assimilation algorithms can be described as either sequential or variational.” Since the 1950’s weather prediction has grown more sophisticated employing better models with increasing resolution using more computing power. With this sophistication have come better predictions: 72 hour predictions are now of the same quality as 36 hour predictions of 15 years ago, and current 36 hour predictions
are considered perfect by the standards of the mid 1950’s. Interestingly the 
performance of subjective weather predictions made by experts continue to 
improve as well. In fact the improvements follow the same pattern as that 
of numerical forecasts with subjective forecasts always having slightly better 
performance.

To understand what is meant by sequential and variational algorithms, 
and more generally how these algorithms adjust the state estimate we need 
to go into some more detail as explained in the next section

6.3 Data Assimilation Algorithms

From the history of weather prediction it has become clear that data assim-
ilation concerns itself with the minimization of error in the initial conditions 
of the model. Historically it was first used to minimize errors in observation 
and thereby the estimate of the current state. The model which is used 
for predictions can also lack accuracy and therefore need adjustment. Data 
assimilation can also be employed to minimize model error by way of per-
forming parameter optimization. The observant reader will notice that the 
estimator algorithms used for minimizing model and observation error also 
recur in fields other than weather prediction that do not use the term data 
assimilation to describe their process. For instance some of the estimators 
used also occur in (electronic) control theory as well as many other fields 
where error minimization is important. Although The term data assimila-
tion is not used universally, it is commonly used within field related to earth 
sciences such as atmospheric research (including weather prediction)[56], 
oceanography, hydrology. What sets data assimilation apart from most re-
search which does not use this term for error minimization is the use of a 
relatively large computational model and the large amount of input data 
required to run this model.

To more clearly explain the data assimilation process we will first enu-
merate its main components and then discuss each in detail with the empha-
sis on estimation. The main components in the data assimilation process 
are:

- Observation Data
- Computational Model
- State estimate
- Prediction
- Estimator
6.3.1 Observation Data

Data from observations is usually not in the form that the computational model needs. Like observations used in any scientific process the data can suffer several problems. Noise through inaccuracy inherent in the sensors employed. Noise through human errors caused by some transformation. For instance wrong date or wrong unit of measurement which can potentially result in extreme values far out of normal range. More often than not observational data is not a direct measurement of what the computational model needs as input. For instance if a model needs a flow rate of objects as an input whereas only observations of individual objects are available, these observations need to be aggregated in order to derive a flow rate. Another possibility is that observations contain information on several phenomena at different timescales and only one of them is of interest to the model. In this case others have to be filtered out.
6.3.2 Computational Model

The computational model incorporates the available theories on which prediction is based. It takes the estimate of the current state as input and in addition can have parameters which affect the working of the model. These can be useful when there is some uncertainty on how two modeled phenomena influence each other. The estimator can adjust these parameters to minimize model error.

6.3.3 State Estimate

The state estimate consists of all data the computational model needs to perform a prediction. The initial state estimate can be based on transformed observation data or be derived from a different, possibly less accurate, model if not enough observations are available to provide a complete estimate. State estimates of subsequent cycles are based on previous predictions. They are adjusted through the estimator based on current observational data.

6.3.4 Prediction

The prediction is the output of the computational model. It is both output of the data assimilation process as well as data used by the estimator to adjust both state estimate and model parameters.

6.3.5 Estimator

The estimator can be used to estimate two things:

- State: minimizing the error in the estimate of the current state, by weighting the influence of observations in the state estimate according to their error co-variance.

- Parameter: minimizing the error introduced by the computational model through weighting the various parameters in the model.

Most data assimilation systems are used for the prediction of unstable systems and employ state estimation. It has been shown by Lorenz[92] that even the slightest error in the initial state estimate for weather prediction will eventually result in totally chaotic predictions. There is a theoretical maximum of about two weeks to weather prediction. Thus for the prediction of unstable systems an accurate state estimate is of great importance. Optimal methods for parameter estimation can be very expensive computationally as it brings a cost increase in the same order as the degrees of freedom in a computational model. This means that in practice for complex models which have to run under real time constraints, non optimal estimators are employed. Especially for weather prediction Kalman filters are not
a practical proposition and therefore less complex but also less optimal estimators are used. In other areas such as hydrology where models are either less complex or real time constraints are less demanding Kalman filters are employed to achieve the best results. What follows is an overview of common estimators, the first three are more computationally efficient and the last three have increasing computational complexity but higher accuracy.

- **3dVar**: three dimensional variational analysis, uses a cost function to minimize state error.
- **4dVar**: four dimensional variational analysis, uses a cost function to minimize state error allows for observations within a time interval instead of just one discreet point in time.
- **Optimal Interpolation**: does state estimation using an error covariance matrix. It is proven to be equivalent to 3dVar with certain assumptions.
- **Kalman Filter**: does state estimation using an error covariance matrix.
- **Extended Kalman Filter**: does optimal state and parameter estimation using an error covariance matrix.
- **Ensemble Kalman Filter**: does a non optimal more efficient way of parameter estimation using an error covariance matrix.

### 6.3.6 Use of ensembles

In the prediction of unstable system where even slight error in the initial state estimate can have a large impact on longer term predictions ensembles are often employed. An ensemble consists of several runs of the computational model each with slightly different initial state estimates. This gives more insight in the effect of error propagation in the final prediction as it shows a whole possible range of outcomes. It also allows the most likely outcome can be chosen by taking an average of all the runs in the ensemble.

### 6.4 Data assimilation toolkits

Within this section an overview is given of toolkits for data assimilation. We will go into detail on the two toolkits used in the case studies, CAPTAIN and SOS, but we will also look at other available toolkits. By means of a comparison of available features and the intended uses of these toolkits, it is our aim to show how data assimilation can benefit from implementation inside an e-Science environment.
6.4. DATA ASSIMILATION TOOLKITS

6.4.1 Overview of Toolkits

Data assimilation tools can be divided into two types, those dedicated to modeling one domain specific phenomenon, and those that provide a more generic framework. Within weather prediction and oceanography many national weather and climate institutes provide access to their model for free.

6.4.2 Toolkits in detail

In this section a detailed view of the toolkits used in the case studies is provided.

Captain Toolbox

Captain[133] is a matlab toolbox for non-stationary time series analysis and forecasting. It was developed at the university of Lancaster and can be obtained directly from there for a license fee. It contains several different modeling methods, all based on Unobserved Components. These modeling methods assume that a time series is composed of a combination of different additive or multiplicative components which cannot be observed directly. It also includes algorithms for data pre processing, system identification and model validation as well as a manual explaining the proper use of the included tools and their theoretical background. The Captain toolbox is very well documented a comprehensive manual is provided with the toolkit. The development of this manual as well as providing support in the use of this toolkit is the main reason that this software is licensed.

SOS toolkit

The SOS Toolkit[128] was developed by Heemink and Verlaan at the Technical University of Delft. Its main use has been in the analysis of various new estimators and it has several estimators build in: a reduced rank square root estimator, an ensemble Kalman filter as well as several estimators implementing a hybrid of both. The details of these algorithms as described and implemented by Verlaan and Heemink can be found in [28]. Like the Captain toolbox it was written in matlab, unlike Captain the source is available making it easier to develop it into services. The toolkit provides a matlab based GUI for running experiments and viewing results. One standard interface exists which allows a model to be used with all available estimators when implemented. Similarly more estimators can be added by implementing their side of the interface. No manual or support is available for this toolkit, only related papers and documentation inside of the matlab code.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported Models</td>
<td>Geophysics</td>
<td>Weather Prediction</td>
<td>Hydrological modeling</td>
<td>Ocean modeling</td>
<td>Time Series Analysis</td>
<td>DA Algorithm</td>
</tr>
<tr>
<td>Supported DA Algorithms</td>
<td>many geophysical models</td>
<td>advanced research wrf and non-hydrostatic mesoscale model</td>
<td>various hydrological models</td>
<td>various ocean and ocean-atmosphere models</td>
<td>many</td>
<td>many</td>
</tr>
<tr>
<td>Model Interface</td>
<td>various ensemble kalman filters</td>
<td>3d-Var/4d-var</td>
<td>ensemble kalman filter</td>
<td>numerous smoothing estimators</td>
<td>4d-var</td>
<td>reduced rank square root, ensemble kalman filter and combinations of both</td>
</tr>
<tr>
<td>GUI</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Programming Language</td>
<td>fortran90</td>
<td>fortran90</td>
<td>matlab</td>
<td>fortran</td>
<td>matlab</td>
<td>matlab</td>
</tr>
<tr>
<td>Parallel Computing</td>
<td>MPI</td>
<td>MPI</td>
<td>no</td>
<td>MPI</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Current Development</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>License</td>
<td>Open Source</td>
<td>Open Source</td>
<td>Open Source</td>
<td>Open Source</td>
<td>closed source</td>
<td>Open Source</td>
</tr>
<tr>
<td>Documentation</td>
<td>only tutorial</td>
<td>only tutorial</td>
<td>manual</td>
<td>manual</td>
<td>manual</td>
<td>no</td>
</tr>
<tr>
<td>Grid Support</td>
<td>no</td>
<td>yes, teragrid</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Problem Solving Environment</td>
<td>no</td>
<td>custom workflow system</td>
<td>matlab</td>
<td>no</td>
<td>matlab</td>
<td>matlab</td>
</tr>
</tbody>
</table>

Table 6.1: Data assimilation toolkits compared
6.4.3 Grid use

In this section we look into more detail at what it takes to run any of the toolkits from the overview, given in table 6.1, on the grid. There is a clear dichotomy to be made in the toolkits reviewed here. SOS, Captain and Daihm\[60\] are toolkits which are used in the development of either models or estimators or both. IOM\[53\], WRF-VAR\[40\] and Dart\[4\] are tools which are used in a setup where a continuous prediction is needed and are aimed at specific applications such as weather prediction. These are more of a production based system where performance and meeting realtime constraints is more important than exploring new model or estimator possibilities. When performance becomes an issue any overhead or inefficiencies, brought about by using the grid and especially existing e-Science workflow systems, become an important reason to be conservative and look for more efficient alternatives to workflow systems. Let us look for instance at the example of using ensembles in data assimilation. For each prediction step the data used for the calculation of each member in an ensemble changes relatively little. Also each ensemble calculation finishes in more or less the same time. After each prediction step there is a synchronization of the data in all steps and then the next prediction step is started. To do this efficiently each of the nodes calculating an ensemble should keep its state and only the information that has changed for the next step should be altered in this state. For greatest efficiency the ensemble calculation should be kept in memory ready to start immediately with the state update also being performed directly in memory. This is something that can be done easier and more efficiently in a dedicated cluster than in a grid environment. The underlying reason is that until now, there is relatively little support for optimization in the scheduling and especially synchronization of more complicated workflows running on the grid.

Matlab toolkits on the grid

Running the Captain toolbox as a set of services is not easy. The toolkit consists of compiled closed source functions which are not easily converted to stand alone programs through the use the Matlab C compiler. Nor can they be run in Octave, an open source alternative to Matlab. Running in Octave is attractive because it saves on license costs when using multiple instances at the same time. The SOS toolkit is able to run in Octave in a limited form after some modification to the code. The SOS toolkit uses some Matlab functions and libraries for which no equivalent is available in Octave. In the clusters available to the VL-e project there is a limited number of Matlab licenses available while Octave is installed on all nodes: its use is therefore unlimited. The only available option for Captain is a wrapper around Matlab which is able to call Matlab functions, for instance
using software such as JMatlink\(^1\). This means that an instance of Matlab has to run for each service that is in use.

## 6.5 Conclusion

Data assimilation is suitable for prediction problems where the phenomenon one is trying to predict is unstable. Unlike phenomena like tides and sunrises which have clear periodicity. For the former problems error propagation will in the end always lead to completely chaotic predictions if the prediction is a certain amount of (time) steps ahead. Another prerequisite is that the computational model available is accurate. If the model error is such that it eclipses errors in state estimation then the sensible approach is to develop a better model. Data assimilation is very much suited to an e-Science environment because it deals with massive amounts of data and massive computation, certainly in the case of ensembles or parameter optimization. As illustrated in this introduction, it is also a technique that can be shared between very different scientific domains like: atmospheric research\(^{[40]}\), hydrology\(^{[60]}\), oceanography\(^{[53]}\), geophysics\(^{[4]}\), and as the case studies in the next chapter will show biology and traffic research.

\(^1\)http://jmatlink.sourceforge.net