1

A Model-Independent Data Assimilation (MIDA) module and its applications in ecology

3

Xin Huang[1,2], Dan Lu[3], Daniel M. Ricciuto[4], Paul J. Hanson[4], Andrew D. Richardson[1,2], Xuehe Lu[5], Ensheng Weng[6,7], Sheng Nie[8], Lifen Jiang[1], Enqing Hou[1], Igor F. Steinmacher[2], Yiqi Luo[1,2,9]

7

1 Center for Ecosystem Science and Society, Northern Arizona University, Flagstaff, AZ, USA

2 School of informatics, Computing, and Cyber Systems, Northern Arizona University, Flagstaff, AZ, USA

3 Computational Sciences and Engineering Division, Climate Change Science Institute, Oak Ridge National Laboratory, Oak Ridge, TN, USA

4 Environmental Sciences Division, Climate Change Science Institute, Oak Ridge National Laboratory, Oak Ridge, TN, USA

5 International Institute for Earth System Science, Nanjing University, Nanjing, China

6 Center for Climate Systems Research, Columbia University, New York, USA

7 NASA Goddard Institute for Space Studies, New York, USA

8 Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China

9 Department of Biological Sciences, Northern Arizona University, Flagstaff, AZ, USA

21

Correspondence to: Xin Huang (xh59@nau.edu)

23

ABSTRACT

Models are an important tool to predict Earth system dynamics. An accurate prediction of future states of ecosystems depends on not only model structures but also parameterizations. Model parameters can be constrained by data assimilation. However, applications of data assimilation to ecology are restricted by highly technical requirements such as model-dependent coding. To alleviate this technical burden, we developed a model-independent data assimilation (MIDA) module. MIDA works in three steps including data preparation, execution of data assimilation, and visualization. The first step prepares prior ranges of parameter values, a defined number of iterations, and directory paths to access files of observations and models. The execution step calibrates parameter values to best fit the observations and estimates the parameter posterior distributions. The final step automatically visualizes the calibration performance and posterior distributions. MIDA is model independent and modelers can use MIDA for an accurate and efficient data assimilation in a simple and interactive way without modification of their original models. We applied MIDA to four types of ecological models: the data assimilation linked ecosystem carbon (DALEC) model, a surrogate-based energy exascale earth system model: the land component (ELM), nine phenological models and a stand-alone biome ecological strategy simulator (BiomeE). The applications indicate that MIDA can effectively solve data assimilation problems for different ecological models. Additionally, the easy implementation and model-independent feature of MIDA breaks the technical barrier of applications of data-model fusion in ecology. MIDA facilitates the assimilation of various observations into models for uncertainty reduction in ecological modeling and forecasting.

1. Introduction

Ecological models require a large number of parameters to simulate biogeophysical and biogeochemical processes (Bonan, 2019; Ciais et al., 2013; Friedlingstein et al., 2006), and specify model behaviors (Luo et al., 2016; Luo and Schuur, 2020). Parameter values in ecological models are mostly determined in some *ad hoc* fashions (Luo et al., 2001), leading to considerable biases in predictions (Tao et al., 2020). The situation becomes even worse when more detailed processes are incorporated into models (De Kauwe et al., 2017; Lawrence et al., 2019). Data assimilation (DA), a statistically rigorous method to integrate observations and models, is gaining increasing attention for parameter estimation and uncertainty evaluation. It has been successfully applied to many ecological models (Fox et al., 2009; Keenan et al., 2012; Richardson et al., 2010; Safta et al., 2015; Wang et al., 2009; Williams et al., 2005; Zobitz et al., 2011). However, almost all those DA studies require model-dependent, invasive coding (Walls et al., 2005). This requires a DA algorithm to be programmed for a specific model. Such model-dependent coding creates a large technical barrier for ecologists to use DA to solve prediction and uncertainty quantification problems in ecology. Thus a model-independent DA toolkit is required to facilitate the use of DA technique in ecology.

DA is a powerful approach to combine models with observations and can be used to improve ecological research in several ways (Luo et al., 2011). First, DA can be used for parameter estimation (Bloom et al., 2016; Hararuk et al., 2015; Hou et al., 2019; Ise and Moorcroft, 2006; Ma et al., 2017; Ricciuto et al., 2011; Scholze et al., 2007). It enables the optimization of parameter values across sites, time and treatments (Li et al., 2018; Luo and Schuur, 2020). For example, Hararuk and his colleagues applied DA to a global land model and substantially improved the explanability of the global variation in soil organic carbon (SOC)

70   from 27% to 41% (Hararuk et al., 2014). When DA was combined with deep learning to improve

71   spatial distributions of estimated parameter values, for example, the Community Land Model

72   version 5 (CLM5) predicted the SOC distribution in the US continent with much higher R^2 of

73   0.62 than CLM5 with default parameters ($R^2 = 0.32$) (Tao et al., 2020). Second, DA can be used

74   to select alternative model structures to better represent ecological processes (Liang et al., 2018;

75   Van Oijen et al., 2011; Shi et al., 2018; Smith et al., 2013; Williams et al., 2009). In the study by

76   Liang et al. (2018), DA was used to evaluate four models. And a two-pool interactive model was

77   selected after DA to best represent SOC decomposition with priming. Additionally, DA can be

78   applied to locate the most informative data to reduce uncertainty, thus guiding the sensor

79   network design. (Keenan et al., 2013; Raupach et al., 2005; Shi et al., 2018; Williams et al.,

80   2005). One DA study at Harvard Forest (Keenan et al., 2013) indicated that only a few data

81   sources contributed to the significant reduction in parameter uncertainty. In spite of powerful

82   applications of DA to ecological research, computational cost is a major hurdle, especially with

83   complex models. Fer et al. (2018) developed a Bayesian model emulation to reduce the time cost

84   of DA from 112h to 6h with the simplified Photosynthesis and Evapotranspiration model.

85   Overall, DA is essential for ecological modeling and forecasting (Jiang et al., 2018) and is

86   helpful for evaluation of different inversion methods (Fox et al., 2009).

87   Applications of traditional DA to ecological research require highly technical skills of

88   users. A successful DA application usually involves model-dependent coding to integrate

89   observations into models. This requires users to have knowledge about model programing. For

90   example, if a complex model (e.g., the community land model) is used in DA, users need to

91   know the programming language (e.g., Fortran) of the model and its internal content to write DA

92   algorithm into the model source code before DA can be conducted. The learning curve for model

93   programing is steep for general ecologists. Furthermore, users often need to update the

94   programming knowledge when a different model is used in DA. For example, scientists who

95   implemented the DA algorithm coded in MATLAB (Xu et al., 2006) to an ecosystem carbon

96   cycle model programmed in Fortran (e.g., TECO) need to understand both MATLAB and

97   Fortran (Ma et al., 2017). Moreover, DA often involves reading observation files about a specific

98   study site. As a result, users usually have to update the codes of model-dependent DA to read

99   new observations from every new study site.

100   A number of tools have been developed to facilitate DA applications (Table 1) but many

101   of them are model dependent, such as the Carbon Cycle Data Assimilation Systems (CCDAS)

102   (Rayner et al., 2005; Scholze et al., 2007), the Carbon Data Model Framework (CARDAMOM)

103   (Bloom et al., 2016), the Ecological Platform for Assimilating Data (EcoPAD) into model

104   (Huang et al. 2019) and Predictive Ecosystem Analyzer (PEcAn) (LeBauer et al., 2013). These

105   tools combine DA algorithms with a specific model. For example, CCDAS specified the DA

106   algorithm to the Biosphere Energy Transfer Hydrology (BETHY) model (Rayner et al., 2005).

107   The hardcoding feature of aforementioned tools make them inflexible to be applied to different

108   models.

109   There are some model independent DA tools that are not tailored to a specific model,

110   such as Data Assimilation Research Testbed (DART) (Anderson et al., 2009), the open Data

111   Assimilation library (openDA) (Ridler et al., 2014), the Parallel Data Assimilation Framework

112   (PDAF) (Nerger and Hiller, 2013) and Parameter Estimation & Uncertainty Analysis software

113   suit (PEST) (Doherty, 2004).

114   However, these model-independent tools suffer from some limitations for a general and

115   flexible DA application. For example, openDA requires users to code three functions to initialize

116 a Java class (Ridler et al., 2014) (Table 1). DART enables incorporating a new model through a

117 range of interfaces (Anderson et al., 2009). It has been successfully applied to atmospheric and

118 oceanic models with currently available interfaces (Anderson et al., 2009; Raeder et al., 2012)

119 and recently to the community land model (Fox et al., 2018). It is likely that users may need to

120 prepare new interfaces for new ecological models to use DART. DART and PDAF adopted the

121 Ensemble Kalman Filter (EnKF) method (Evensen, 2003), which may makes it difficult to obey

122 mass conservation for biogeochemical models. This is because the parameter values estimated by

123 EnKF change each time when new data sets are assimilated (Allen et al., 2003; Gao et al., 2011;

124 Trudinger et al., 2007). The sudden changes in estimated parameter values at time points when

125 data are assimilated by EnKF usually do not reflect reality of biogeochemical cycles in the real

126 world. PEST utilizes Levenberg-Marquardt method (Levenberg, 1944) which is a local

127 optimization method for parameter estimation. If the relationship between simulation outputs and

128 parameters are highly nonlinear, which is common in ecological models, this method may trap

129 into a locally optimization solution (Doherty, 2004).

130   In this work, we developed a model-independent DA module (MIDA) to enable a general

131 and flexible application of DA in ecology. MIDA was designed as a highly modular tool,

132 independent of specific models, and friendly to users with limited programming skills and/or

133 technical knowledge of DA algorithms. Additionally, MIDA implemented advanced Markov

134 Chain Monte Carlo (MCMC) algorithms for DA analysis which can accurately quantify the

135 parameter uncertainty with informative posterior distribution. The anticipated user community in

136 this initial phase of MIDA development is the biogeochemical modelers who are looking for

137 appropriate parameter estimation methods. In the following Section 2, we first introduce the

138 development details of MIDA and its usage. In Section 3, we demonstrate the application of

139 MIDA to four different types of ecological models. In Section 4, we discuss the strengths and

140 weaknesses of MIDA in ecological modelling and lastly we give our concluding remarks in

141 Section 5.

142

143 **2. Model-independent data assimilation (MIDA)**

144 **2.1 Bayes's theorem and DA**

145 Based on Bayes' theorem, DA is a statistical approach to constrain parameter values and

146 estimate their posterior density distributions through assimilating observations into a model. The

147 posterior density distributions $p(C|Z)$ of parameters C for a given observation Z can be obtained

148 from *prior* density distributions $p(C)$ and the likelihood function $p(Z|C)$:

149 $$p(C|Z) \propto p(Z|C)p(C) \qquad (1)$$

150 The *prior* density distribution $p(C)$ is assumed as a uniform distribution over the parameter

151 range. And the likelihood function is negatively proportional to a cost function, J as:

152 $$p(Z|C) \propto exp(-J) \qquad (2)$$

153 The cost function measures the misfit between simulation outputs and observations and is

154 described in more detail in section 2.4. The posterior density distributions $p(C|Z)$ is estimated

155 from sampling parameter values to maximize the likelihood function $p(Z|C)$ or minimize the

156 cost function J. DA usually uses a sampling technique, such as Markov chain Monte Carlo

157 (MCMC) in this MIDA. The MCMC algorithm successively generates a new set of parameter

158 values from the prior parameter ranges and requires model run with these new parameter values.

159 Then the cost function is calculated to determine whether this new set of parameter values will

160 be accepted or not according to the Metropolis-Hastings criterion (see more description in

161 section 2.4). All accepted parameter values are used to generate posterior distributions where the

162   distinctive mode indicates the parameter uncertainty is well constrained. Meanwhile, we derive

163   maximum likelihood estimates (MLEs) of parameters from the posterior density distributions.

164   MIDA realizes model-independent Bayesian-based DA to estimate posterior density

165   distributions and MLEs of parameters via data exchanges between a given model and DA

166   algorithm.

167

168   **2.2 An overview of MIDA**

169   MIDA is a module that allows for automatic implementation of data assimilation without

170   intrusive modification or coding of the original model (https://doi.org/10.5281/zenodo.4762725,

171   last access: May 2021). Its workflow includes three steps: data preparation, execution of data

172   assimilation, and visualization (Fig. 1). Step 1 (data preparation) is to establish the standardized

173   data exchange between DA algorithm and the model. Step 2 (execution of data assimilation) is to

174   run DA as a black box independent of the model. Step 3 (visualization) is to diagnose parameter

175   uncertainty after DA. The modularity of the 3-step workflow is designed to enable MIDA for a

176   rapid DA application and adaption to a new model. In the following, we introduce the three-step

177   workflows of MIDA, its technical implementation and usage in detail.

178

179   **2.3 Step 1: Data preparation**

180   Step 1 is designed to initialize data exchange to transfer parameter values, model outputs,

181   observations and their variances between DA algorithm and the model to be used. Four types of

182   information are required either from interactive input or by modifying the 'namelist.txt' file (Fig.

183   1). The first type is about DA configuration, including the number of sampling series in DA and

184   the working path where the outputs of DA will be saved. The number of a sampling series is

185 essential in a DA task to define how many times parameter values are sampled to run the model.

186 The second type of information is about parameter ranges and their covariance. The third is the

187 model executable file. Finally, the fourth type is an output configuration file which contains the

188 file paths of model outputs, observations, and their variance. This file also instructs how to read

189 model outputs and compare each output with corresponding observation.

190   Traditional DA requires users to modify the code of model to incorporate the process of

191 data exchange between DA algorithm and the model. Therefore, the program of data exchange in

192 traditional DA is model-specific and users need to repeat such program when a new model

193 comes. In MIDA, the process of data exchange calls a model executable file which hides the

194 details of model code. When applied to a new model, MIDA only requires users to provide a

195 different model executable file in the 'namelist.txt' file and does not involve any additional

196 coding in either the model or MIDA. Thus, MIDA lowers the technical barrier for general

197 ecologists to conduct DA.

198   Traditional DA usually preset the number of parameters and the model outputs according

199 to a specific model before initializing the data exchange. This is because data exchange between

200 DA algorithm and model uses memory to transfer items such as parameter values. Instead,

201 MIDA organizes items in data exchange using different files. Items in data exchange are decided

202 by the data file loaded when MIDA is running. The number of parameter values, for example,

203 will be decided after the file of parameter range is read in MIDA. Through modifying files,

204 MIDA allows making efficient choices about the model-related items in data exchange. Thus,

205 MIDA is highly flexible and modular for DA with different models.

206   Traditional DA also preset observation types in the data exchange according to a specific

207 study before the data exchange. For example, if the traditional DA uses carbon flux observation,

208   it cannot switch to satellite remote sensing products without additional coding. MIDA uses the

209   concepts of object-orient programming (Mitchell and Apt, 2003) and dynamic initialization

210   (Cline et al., 1998) in computer science to provide a homogenous way to create various

211   observation types from a unified prototype class. A prototype class includes variables to store

212   observations and their variance and functions (e.g., read from observation files). The values in

213   variables are dynamically decided after the observation files are loaded when MIDA is running.

214   Different observation types derive from the prototype class with a high degree of reusability of

215   most functions. In such way, MIDA only requires users to provide different filenames of the

216   observations to be integrated in DA. Therefore, MIDA is highly flexible and modular for DA to

217   assimilate various observations.

218

219   **2.4 Step 2: Execution of data assimilation**

220   After the establishment of the standardized data exchange (step 1), step 2 is to run DA as a black

221   box for users without knowledge of DA itself. Notwithstanding the black-box goal, this section

222   provides a general description of DA below.

223   Data assimilation as a process integrates observations into a model to constrain

224   parameters and estimate parameter uncertainties. Data assimilation usually uses some types of

225   sampling algorithms, such as Markov chain Monte Carlo (MCMC), to generate posterior

226   parameter distribution under a Bayesian inference framework (Box and Tiao, 1992). As

227   mentioned in section 2.1, DA with MCMC algorithm estimates the posterior density distributions

228   through sampling to maximize likelihood function $p(Z|C)$ or minimize the misfit J between

229   simulation outputs and observations. This version of MIDA uses MCMC algorithm implemented

230   by the Metropolis-Hasting (MH) sampling method (Hastings, 1970; Metropolis et al., 1953). The

231   future version of MIDA could incorporate other data assimilation algorithms. Each iteration in

232   the Metropolis-Hasting sampling includes a proposing phase and a moving phase. The proposing

233   phase generates a new set of parameter values based on the starting point for the first iteration or

234   current accepted parameter values in the following iterations. If parameter covariance

235   (cov_{param}) is specified in step 1 on data preparation, this proposing phase will draw new

236   parameter values (C_{new}) within the prior ranges from a Gaussian distribution $N(C_{old}, cov_{param})$

237   where C_{old} is the predecessor set of parameter values. Without parameter covariance, new set of

238   parameter values will be generated from a uniform distribution within the prior ranges (Xu et al.,

239   2006).

240   The moving phase first calculates mismatches between observations and the model

241   simulation with the new set of parameter values as a cost function (J_{new} in Eq.3) (Xu et al.

242   2006):

243
$$J_{new} = \sum_{i=1}^{n} \frac{\sum_{t \in obs(z_i)} [Z_i(t) - X_i(t)]^2}{2\sigma_i^2} \tag{3}$$

244   Where n is the number of observations, $Z_i(t)$ is the ith observation at time t, $X_i(t)$ is the

245   corresponding simulation, σ_i^2 is the variance of the observation. The error is assumed to

246   independently follow a Gaussian distribution. This new set of parameter values will be accepted

247   if J_{new} is smaller than J_{old}, the cost function with the previous set of accepted parameter values,

248   or the value, $\exp\left(-\frac{J_{new}}{J_{old}}\right)$, is larger than a random number selected from a uniform distribution

249   from 0 to 1 according to the Metropolis criterion (Liang et al., 2018; Luo et al., 2011; Shi et al.,

250   2018; Xu et al., 2006). Once the new set of parameter values is accepted, J_{new} becomes J_{old}.

251   Those two phases of sampling will be iteratively executed until the number of sampling series set

252   in step 1 on preparation of DA is reached. Finally, the posterior density distributions can be

253   generated from all the accepted parameter values.

254   MIDA realizes the execution of data assimilation according to the procedure described

255   above. First, MIDA uses a 'call' function to execute model simulations to get values of $X_i(t)$.

256   Observations $Z_i(t)$ and their variance σ_i^2 are already provided via the standardized data

257   exchange as described in step 1. Then, MIDA calculates J_{new} according to Eq. 3 to decide the

258   acceptance of the current parameter values used in this simulation. If accepted, MIDA saves this

259   set of parameter values and associated J_{new} values in $C_{accepted}$ and $J_{accepted}$ arrays respectively

260   and triggers new proposing phrase based on this set of accepted parameter values. If not, MIDA

261   discards this set of parameter values and generates another new set of parameter values. MIDA

262   saves the new parameter values generated in the proposing phrase to "ParameterValue.txt", from

263   which the model reads before execution of the next model simulation. MIDA repeats the

264   proposing and moving phases until the number of sampling series is reached. At the end, MIDA

265   selects the best parameter values through maximum likelihood estimation and run model again

266   using this set of values to get optimized simulation outputs $X_i(t)$. Then MIDA saves the arrays

267   of accepted parameters, associated errors, maximum likelihood estimates (MLEs), and optimized

268   state variables $X_i(t)$ to four files, "parameter_accepted.txt", "J_accepted.txt", "MLE.txt", and

269   "OptimizedSimu.txt", respectively.

270   This execution of DA algorithm in MIDA enables users to conduct DA as a black box

271   and is independent of any particular model.

272

273   **2.5 Step 3: Visualization**

274   Step 3 is to visualize the results of DA in step 2. The end products of DA are accepted parameter

275   values, their associated J_{new} values, the maximum likelihood estimates, and optimized

276   simulation results as saved in the output files. MIDA enables visualization of parameter posterior

277   density distributions with a Python script. In the script, MIDA first read accepted parameter

278   values from "parameter_accepted.txt" file. Then, MIDA generates posterior probabilistic density

279   function (PPDF) for each parameter via 'kdeplot' function in the 'seaborn' package. The

280   maximum likelihood estimates of parameters correspond to the peaks of PPDF. The distinctive

281   mode of PPDF indicates how well the parameter uncertainty is constrained. Finally, MIDA

282   visualizes the PPDF for all parameters in a figure using the 'matplotlib' package.

283

284   **2.6 Implementation and architecture of MIDA**

285   MIDA is equipped with a graphical user interface (GUI) and users can easily execute it through

286   an interactive window. Users can also run MIDA as a script program without the GUI. MIDA is

287   written in Python (version 3.7). For the GUI-version, all relevant Python packages used in MIDA

288   are compiled together, thus users do not need to install them by themselves. For the non-GUI

289   version, users need to install Python 3.7 and relevant packages (i.e., numpy, pandas, shutil,

290   subprocess, matplotlib, math, os, and seaborn). MIDA is compatible with model source codes

291   written in multiple programming language (e.g., Fortran, C/C++, C#, MATLAB, R, or Python).

292   It is also independent of multiple operation systems (e.g., Windows, Linux, MacOS). In addition,

293   MIDA is also able to run on high-performance computing (HPC) platforms via task management

294   systems (e.g., Slurm).

295   The architecture of MIDA is class-based and each class is designed to describe an object

296   (e.g., parameter, observations, etc.) with variables and operations. Five classes are defined in

297   MIDA: parameter, observation, initialization, MCMC algorithm and the main program. The

298   main program is the start of MIDA execution. It calls functions from all other classes to finish

299   three-step workflow. As described in section 2.2, parameter and observation classes contain

300   variables to be transferred in data exchanges via file I/O operations. These operations are

301   implemented using the 'numpy' package. The initialization class is to read 'namelist.txt' in step

302   1 on data preparation and to assign values for the variables in all other classes. Then the class of

303   MCMC algorithm conducts DA as described in step 2. In this step, the simulation operation uses

304   a 'call' function in 'subprocess' package to call model executable. At the start of model

305   simulation, MIDA writes new parameter values to the 'ParameterValue.txt' file in the 'working

306   path' directory specified in step 1 on data preparation. Then the model executable read parameter

307   values from the 'ParameterValue.txt' file and run. After model simulation, DA algorithm can

308   read the model outputs by the output filenames indicated in the output configuration file. After

309   DA, step 3 executes an additional Python script to read accepted parameter values and plot the

310   posterior density distributions of parameters. The plotting operations uses 'matplotlib' and

311   'seaborn' packages. The implementation of GUI uses pyQt5 toolkit to support interactive usage

312   of MIDA. Users can also run MIDA in a non-interactive way with a 'main.py' script to trigger

313   the three-step workflows.

314

315   **2.7 User information of MIDA**

316   In order to use MIDA, users need to prepare data and a model. The data to be used in MIDA are

317   prior ranges and default values of parameters, parameter covariances, output configuration file,

318   observations and their variances. They are organized in different files. Before running MIDA,

319   users need to specify their filenames as suggested in step 1. When users want to use different

320   data sets in DA, they can simply change filenames with the new data sets via GUI or in the

321   'namelist.txt' file. Figure C1 is an example of the 'namelist.txt' file for a data assimilation study

322   with the DALEC model. The model to be used in MIDA should have those to-be-estimated

323   parameter values not fixed in model source code rather than changeable through

324   'ParameterValue.txt' file. MIDA writes new parameter values in each proposing phase during

325   DA to the 'ParameterValue.txt' file, from which the model reads the parameter values to run the

326   simulation.

327   To calculate the cost function, J, we have to have a one-to-one match between

328   observations and model outputs. For example, phenology models in one of the application cases

329   of MIDA below generate discrete dates of leaf onset, which is a one-to-one match to the

330   observations of spring leaf onset. In this case, observation $Z_i(t)$ and model output $X_i(t)$ to be

331   used in calculation of J is straightforward. In the application case for dynamic vegetation, the

332   data to be used are leaf area in six layers in a forest of 302 years old whereas the model simulates

333   leaf areas in eight layers from 0 to 800 years. To match observation, the model generates outputs

334   of leaf areas in six layers when simulated forest age reaches 302 years. This requires users to

335   prepare an output configuration file to instruct MIDA to read model outputs and re-organize their

336   outputs to match observation. The output configuration file starts with a single line listing an

337   observation filename and its corresponding output filenames. Content after the directories in the

338   output configuration file are instructions to map model outputs with the observation signified in

339   the first line. Each instruction is to match one or continuous elements in observation with

340   elements in outputs with the same length. A blank line means there are no further instructions.

341   Then a new matching between another observation and model outputs starts. An example of

342   output configure file is available in Appendix B.

343   Once MIDA finishes the execution of data assimilation, users may need basic knowledge

344   to assess the performance of DA. For example, the acceptance rate, which is given by MIDA, is

345   the fraction of proposed parameter values that is accepted. Ideally, the acceptance rate should be

346   about 20 ~ 50% (Xu et al., 2006). A very low acceptance rate indicates that many new proposed

347   parameter values (C_{new}) are rejected because C_{new} jumps too far away from the previously

348   accepted parameter values (Robert and Casella, 2013; Roberts et al., 1997). In this case, users are

349   suggested to reduce a jump scale in the proposing phase. On the other hand, a very high

350   acceptance rate is likely because C_{new} moves slowly from the previously accepted parameter

351   values. Users may increase the jump scale.

352   In addition, DA usually requires a convergence test to examine whether posterior

353   distributions from different sampling series converge or not. Convergence test requires running

354   DA parallelly or in multiple times with different initial parameter values. MIDA provides a

355   Gelman-Rubin (G-R) test (Gelman and Rubin, 1992) for this purpose. To use the G-R test, users

356   need to prepare a file containing initial parameters values in different sampling series and

357   indicate its filename in the 'namelist.txt' file as described in step 1. If the G-R statistics

358   approaches one, the sampling series in DA is converged. When sampling series is converged, all

359   accepted parameter values are used to generate the posterior distributions.

360   There are three types of posterior distributions: bell-shape, edge-hitting, and flat. The

361   bell-shaped posterior distributions indicate that these parameters are well constrained. Their peak

362   values are the maximum likelihood estimates of parameter values. The flat posterior distributions

363   suggest that the parameters are not constrained due to the lack of relevant information in data.

364   The edge-hitting posterior distributions result from complex reasons, such as improper prior

365   parameter range. Users may change the prior ranges to examine if those posterior distributions

366   can be improved or examine correlations among estimated parameters.

367

368   **3.   Applications of MIDA**

369   We applied MIDA to four groups of models, which are an ecosystem carbon cycle model, a

370   surrogate-based land surface model, nine phenology models, and a dynamic vegetation model,

371   respectively. These four cases demonstrate that MIDA is effective for stand-alone DA, flexible

372   to be applied to different models, and efficient for multiple model comparison.

373   **3.1 Case 1: Independent data assimilation with DALEC**

374   The first case study is to demonstrate that MIDA can be effective for independent data

375   assimilation with the data assimilation linked ecosystem carbon (DALEC) model (Lu et al.,

376   2017). DALEC has been used for data assimilation in several studies (Bloom et al., 2016; Lu et

377   al., 2017; Richardson et al., 2010; Safta et al., 2015; Williams et al., 2005). Previous studies all

378   incorporated data assimilation algorithms into DALEC, which requires invasive coding. This

379   case study is focused on reproducing the data assimilation results as in the study by Lu et al.

380   (2017) but with MIDA.

381   The version of DALEC used in this study is composed of six submodels (i.e.,

382   photosynthesis, phenology, autotrophic respiration, allocation, litterfall, and decomposition) to

383   simulate the carbon exchanges among five carbon pools (i.e., leaf, stem, root, soil organic matter

384   and litter) (Ricciuto et al., 2011). There are 21 parameters in DALEC, of which, 17 parameters

385   are derived from the six submodels and four parameters serve to initialize the carbon pools.

386   Table 2 summarizes the names, prior ranges and nominal values of these 21 parameters. The

387   observation is the Harvard Forest daily net ecosystem exchange (NEE) from year 1992 to 2006.

388   DALEC is coded in Fortran. In windows system, a gfortran compiler converts the model code to

389   an executable file (i.e., DALEC.exe).

390   Figure 2 is the GUI window of MIDA. We first set up a DA task as described in step 1

391   using the upper panel. In this application, the number of sampling series is set as 20,000. Once

392   users click the 'choose a directory' or 'choose a file' button, a new dialog window will pop up

393   and users are able to choose the directory or load files interactively. As describe in step 1 on

394   preparation of DA, the working path is where the outputs of DA and 'ParameterValue.txt' are

395   saved (e.g., C:/workingPath). After the output configuration file is loaded, the filenames of

396   model outputs, observations and their variance will be displayed in the window automatically.

397   This application only uses a 'NEE.txt' observation file. Similarly, after users load parameter

398   range file (e.g., a file named 'ParamRange.txt' contains three rows which are minimum,

399   maximum and default values of parameters), the content in this file is displayed as well. To

400   replace the current parameter range file loaded, users can simply upload another file. In this

401   application, the executive model file is 'DALEC.exe' with Fortran compiler in windows system.

402   Because we do not have parameter covariance information, this input is left blank. After 'save to

403   namelist file' is clicked, a 'namelist.txt' file containing all the inputs will be generated in the

404   working path.

405   After the DA task set up, we load the 'namelist.txt' file and click the 'run data

406   assimilation' button in the lower panel to trigger step 2 on execution of DA. A new dialog will

407   pop up to show the acceptance rate information and notify the termination of DA. Then we will

408   click the 'generate plots' button to visualize the posterior distributions of 21 parameters as

409   described in step 3.

410   Figure 3 shows that the simulation outputs using the optimized parameter values from

411   MIDA better fit with the observations than those using default parameter values. Figure 4 depicts

412   posterior distributions of the 21 parameters estimated from MIDA. More than half of the

413   parameters are constrained well with a unimodal shape. $X_{stem_{init}}$ and $X_{root_{init}}$ have a wide

414   occupation of the prior range, indicating that the observation data does not provide useful

415   information for them. The constrained posterior distributions in this study are similar to those

416   from the study in Lu et al. (2017). Note that MCMC estimates have a large variance and a low

417   convergence rate especially in high-dimensional problems, with a finite number of samples it is

418   not expected that two simulations would give exactly the same results.

419
420   **3.2 Case 2: Application of MIDA to a surrogate land surface model**

421   This case study is to examine the applicability of MIDA to a surrogate-based land surface model.

422   The original model is energy exascale earth system model: the land component (ELM) (Ricciuto

423   et al., 2018). As ELM is computationally expensive (one forward model simulation takes more

424   than one day), a sparse-grid (SG) surrogate system was developed to reduce the computational

425   time (Lu et al., 2018). The forcing data for the surrogate model is half-hourly meteorological

426   measurements at Missouri Ozark flux site from 2006 to 2014. The observations that were used

427   for optimization are annual sums of net ecosystem exchange (NEE), annual averages of total leaf

428   area index and latent heat fluxes from 2006 to 2010. The eight parameters selected (Table 3) are

429   the most important parameters for the variations in outputs (Ricciuto et al., 2018). The model is

430   written in Python. A 'pyinstaller' library packages the model code into an executable file. The

431   iteration number in MIDA is 20,000.

432   Figure 5 shows posterior distributions of calibrated parameters. c_{root}, SLA_{top},

433     $t_{leaffall}$, GDD_{onset} are constrained well with a unimodal distribution. However, the distribution

19

434   of the rest 4 parameters (i.e., N_{leaf}, CN_{root}, A_{r2l} and Res_m) cluster at near the edge. These

435   results match well with the study by Lu et al. (2018). As shown in Figure 6, the calibrated

436   parameters induce a performance improvement in simulating total leaf area index and NEE. For

437   latent heat, both the default and optimized simulation obtain good agreement with the

438   observation. These conclusions are also similar to those in Lu et al. (2018).

439   MIDA hides the detailed differences between models. For example, DALEC model in

440   case 1 is a process-based model to simulate ecosystem carbon cycle while surrogate-based ELM

441   in case 2 is an approximation of land surface model. They are also different in programming

442   language, simulation time, forcing data, etc. MIDA is able to deal with models with so many

443   different characteristics and hides these differences from users. Users only need to indicate the

444   filenames of the model to be used, its parameter range, the output configuration file, etc. in the

445   'namelist.txt' file. Thus, MIDA simplified the DA applications using different models.

446

447   **3.3 Case 3: Evaluation of multiple phenological models**

448   This study case uses nine phenological models (Yun et al., 2017) to demonstrate the applicability

449   of MIDA in model comparison. Five out of the nine models predict phenological events, such as

450   the day of leaf onset, using growing degree days, which are calculated as temperature

451   accumulation above a base temperature. The other four models consider two processes: chilling

452   effects of cold temperature on dormancy before budburst and forcing effects of warm

453   temperature on plant development. Each model uses different response functions to represent

454   chilling and forcing effects. The detailed model descriptions and associated parameter

455   information are in supplementary table.

456   Data are from the Spruce and Peatland Responses Under Climatic and Environmental

457   Change experiment (SPRUCE) (Hanson et al., 2017) located in northern Minnesota, USA. The

458   experiment consists of five-level whole-ecosystem warming (i.e., +0, +2.25, +4.5, +6.75, +9°C)

459   and two-level elevated CO_2 concentrations (i.e., +0, +500ppm). Dates of leaf onset were

460   observed with PhenoCam (Richardson et al., 2018) for tree species: *Picea mariana* and *Larix*

461   *laricina*. For the sake of demonstration of MIDA application, we only show DA results for *Larix*

462   *laricina* with +9°C warming treatment and +0 ppm CO_2 treatment from 2016 to 2018.

463   MIDA was used to compare performances of the nine models in reference to the same

464   observations of leaf onset dates after DA. We as users changed filenames of model executable

465   file (i.e., PhenoModels.exe), defined parameter ranges, and assigned the directory of working

466   path for each model. MIDA then estimated the optimized parameters and save the corresponding

467   best simulation outputs to the working path for each of the nine models. Figure 7 shows the best

468   simulation output of these nine models. The simulation output of the 6th, 7th, 8th, and 9th models

469   better fit the observation than the other models. It demonstrates that models that consider both

470   chilling and heating effects can achieve good simulations of the leaf onset dates.

471

472   **3.4 Case 4: Supporting data assimilation with a dynamic vegetation model**

473   This case study is to examine the efficiency of MIDA to integrate remote sensing data into a

474   dynamic vegetation model. The model used in this study is Biome Ecological strategy simulator

475   (BiomeE) (Weng et al., 2019). This model simulates vegetation demographic processes with

476   individual-based competition for light, soil water, and nutrients. Individual trees in BiomeE

477   model are represented by cohorts of trees with similar sizes. The light competition among

478   cohorts is based on their heights and crown areas according to the rule of perfect plasticity

479   approximation (PPA) model (Strigul et al., 2008). Each cohort has seven pools: leaves, roots,

480   sapwood, heartwood, seeds, nonstructural carbon and nitrogen. After carbon are assimilated into

481   plants via photosynthesis, the assimilated carbon enters to nonstructural carbon pool and is used

482   for plant growth (i.e., diameter, height, crown area) and reproduction according to empirical

483   allomeric equations (Weng et al., 2019). In this application, two parameters to be constrained

484   (Table 4) are annual productivity rate and annual mortality rate of trees.

485   Observations to be used in DA are leaf area indexes in six vertical heights (i.e., 0-5m, 6-

486   10m, 11-15m, 16-20m, 21-25m, and 26-30m) at Willow Creek study site, Wisconsin, USA. The

487   forest at the site is an upland deciduous broadleaf forest of around 302 years old. The

488   observations were from Global Ecosystem Dynamics Investigation (GEDI) acquired by a Light

489   Detection and Ranging (Lidar) laser system, which is deployed on the International Space

490   Station (ISS) by NASA in 2018 (Dubayah et al., 2020). The observations were first averaged

491   from three footprints and then leaf area indexes in the six canopy layers were standardized to be

492   summed up as one.

493   To use MIDA, we reorganized the simulation outputs to match observations as suggested

494   in section 2.6. The BiomeE model simulates leaf areas in eight layers (i.e., 0-5m, 6-10m, 11-

495   15m, 16-20m, 21-25m, 26-30m, 31-35m, and 36-40m) from 0 to 800 years. An output

496   configuration file was provided to post-process model outputs of leaf area indexes in six layers to

497   match observations at the forest age of 302 years. These simulated leaf area indexes in the six

498   canopy layers were also standardized to match standardized observations of leaf area indexes.

499   The observations and post-processed simulation outputs were saved to 'LAI.txt' and

500   'simu_LAI.txt' files, respectively. The two files are used in MIDA for data assimilation to

501   generate posterior distributions of estimated two parameters as showed in figure 8. The

502 optimized parameter values through maximum likelihood estimation are different from their

503 default values. Figure 9 compares the simulation outputs with optimized parameters estimated by

504 MIDA to those with default parameter values. After DA with GEDI data in MIDA, the

505 simulation accuracy of leaf area index is substantially improved especially in middle (16~20m)

506 and highest (26~30m) layers.

507

508 **4. Discussion**

509 This study introduced MIDA as a model-independent tool to facilitate the applications of data

510 assimilation in ecology and biogeochemistry. The potential user community is ecologists with

511 limited knowledge of model programming and technical implementation of DA algorithms.

512 Several model-independent DA tools have already been developed, such as DART (Anderson et

513 al., 2009), openDA (Ridler et al., 2014), PDAF (Nerger and Hiller, 2013) and PEST (Doherty,

514 2004), mainly for applications in research areas of hydrology, atmosphere, and remote sensing.

515 These DA tools either use gradient descent method, such as Levenburg-Marqurdt algorithm in

516 PEST, or Kalman Filter methods, such as EnKF in DART, openDA, and PDAF. The Levenburg-

517 Marqurdt algorithm is a local search method, which is hard to find global optimization solution

518 for highly nonlinear models. EnKF updates state variables and parameter values each time when

519 observations are sequentially assimilated, resulting discrete values of estimated parameters.

520 Jumps in estimated parameter values by EnKF make it very difficult to obey mass conservation

521 in biogeochemical models (Gao et al., 2011). In this study, we used the MCMC method in MIDA

522 to generate parameter values and their posterior distributions. MCMC is a widely used method in

523 many DA studies with biogeochemical models but has been applied to individual models with

524 invasive coding (Bloom et al., 2016; Hararuk et al., 2015; Liang et al., 2018; Luo and Schuur,

23

525 2020; Ricciuto et al., 2011). Compared to the other model-independent DA tools mentioned

526 above, MIDA is the first tool that uses the MCMC method for DA.

527   Biogeochemical models are incorporating more detailed processes related to carbon and

528 nitrogen cycles (Lawrence et al. 2020). Complex biogeochemical models yield predictions with

529 great uncertainty (Frienlingstein et al. 2009 and 2014). Data assimilation has been increasingly

530 used to estimate parameter values against observations and reduce uncertainty in model

531 prediction (Luo et al. 2016, Luo and Schuur 2020). However, current applications of DA are

532 almost all model dependent. It requires ecologists to write code to integrate DA algorithm with

533 models. The coding practice is a big technical challenge for ecologists with limited program

534 ability. The distinct advantage of MIDA is to enable ecologists to conduct model independent

535 DA. MIDA streamlines workflow of the three-step procedure for DA to enable users to conduct

536 DA without extensive coding. Users mainly need to provide numerical and character values for

537 data exchanges to transfer data (i.e., parameter values, simulation outputs, observations) between

538 the model and MIDA by a file named 'namelist.txt' or by interactive inputs via a GUI window

539 (Fig. 2).

540   We tested MIDA in four cases for its applicability to ecological models. The first case is

541 applied to DALEC model, which has been used in several data assimilation studies (Bloom et al.,

542 2016; Lu et al., 2017; Safta et al., 2015; Williams et al., 2005). The previous DA studies all used

543 invasive coding to incorporate DA algorithm into models. As demonstrated in this study, MIDA

544 was applied to DALEC without invasive coding but by providing the directory to save DA

545 results and filenames of DALEC model executable, parameter prior range, and output

546 configuration file through the 'namelist.txt' file or interactive inputs in the first preparation step

547 of the workflow. Then, MIDA run DA as a black box with DALEC before visualizing the DA

548 results. Next, we tested the applicability of MIDA a surrogate-based ELM model and a dynamic

549 vegetation model BiomeE. To switch the test case from DALEC to the surrogate-based ELM

550 model and the BiomeE model, we changed the filenames of model executable, parameter prior

551 range, and output configuration file in the 'namelist.txt' file for MIDA. This flexibility of MIDA

552 in switching models for DA makes it much easier for model comparisons. We tested this

553 capability of MIDA with nine phenological models to compare alternative model structures.

554 Similarly, MIDA enables efficient switches of observations to be assimilated into models. Users

555 only need to change filenames of observations in the output configuration file. This feature of

556 MIDA makes it easier to utilize abundant traits databases such as TRY (Kattge et al., 2020),

557 FRED (Iversen et al., 2017), etc. Moreover, this feature of MIDA also helps evaluating the

558 relative information content of different observations for constraining model parameters and

559 prediction (Weng and Luo, 2011). Consequently, MIDA can facilitate selection of the most

560 informative observations and then better guide data collections in filed experiments. Ultimately,

561 MIDA can aid ecological forecasting and help reduce uncertainty in model predictions (Huang et

562 al., 2018; Jiang et al., 2018).

563 Although MIDA helps users to get rid of model detail, users may still need basic

564 knowledge about the model outputs to prepare the output configuration file which is to match

565 model outputs to observations one-by-one (see Section 2.6). This effort of preparing the

566 correspondence between model outputs and observations for MIDA is not that difficult because

567 users are reading or writing a text file and most model developers will provide reference to help

568 understanding observations or model output files.

569 Generally, MIDA requires longer time to run DA than the embedded DA algorithm,

570 because MIDA calls model simulation as an external executable rather than a function

571 embedded. Thus, we recommend MIDA for beginners of DA users with models that are less

572 complex. Besides, the current version of MIDA only incorporates Metropolis-Hasting sampling

573 approach. More MCMC methods (e.g., Hamiltonian Monte Carlo) may be incorporated into

574 MIDA in the future.

575

576 **5. Conclusions**

577 We developed MIDA to facilitate data assimilation for biogeochemical models. Traditional DA

578 studies require ecologists to program codes to integrate DA algorithms into model source codes.

579 The easy-to-use MIDA module enables ecologists to conduct model-independent DA without

580 extensive coding thus advancing the application of DA for ecological modeling and forecasting.

581 We demonstrated the capability of MIDA in four cases with a total of 12 ecological models.

582 These cases showed that MIDA is easy to perform for a variety of models and can efficiently

583 produce accurate parameter posterior distributions. Moreover, MIDA supports flexible usage of

584 different models and different observations in the DA analysis and allows a quick switch from

585 one model to another. This capability enables MIDA to serve as an efficient tool for model

586 intercomparison projects and enhancing ecological forecasting.

587

588 **Appendix A:** Nine phenological models

589 1. Growing degree (GD)

590 The growing degree (GD) model is one of the most widespread phenological model to simulate

591 the date of leaf onset ($\widehat{D}$). In this study, the time scale is limited to daily based on observation

592 records. The kernel of GD is to calculate the growing degree days (GDD, $\sum_{d=D_s}^{\widehat{D}-1} \Delta d$) which is the

593 heat accumulation above a base temperature (T_b). For simplicity, the daily temperature (T_d) can

594   be approximated by the average of daily maximum and minimum temperatures. The heat

595   accumulation starts at day D_s, which is empirically estimated, and ends when GDD reaches a

596   forcing requirement threshold (R_d). Two parameters to be constrained are base temperature (T_b)

597   and the forcing requirement (R_d). Their default values and prior range are listed in Table A1.

598
$$\Delta d = \begin{cases} T_d - T_b \ if \ T_d > T_b \\ \quad 0 \quad \ otherwise \end{cases} \tag{A1}$$

599
$$\sum_{d=D_s}^{\widehat{D}-1} \Delta d < R_d \le \sum_{d=D_s}^{\widehat{D}} \Delta d \tag{A2}$$

600   2.   Sigmoid function (SF)

601   Compared to the linear response function of GDD in GD model, the sigmoid function (SF)

602   model provides a non-linear function to better represent the non-linearity of the growth response

603   to heat accumulation. Three parameters to be constrained in DA are base temperature (T_b), the

604   forcing requirement (R_d) and temperature sensitivity (S_t). Their default values and prior range

605   are listed in Table A1.

606
$$\Delta d = \frac{1}{1+e^{S_t(T_d-T_b)}} \tag{A3}$$

607
$$\sum_{d=D_s}^{\widehat{D}-1} \Delta d < R_d \le \sum_{d=D_s}^{\widehat{D}} \Delta d \tag{A4}$$

608   3.   Beta function (BF)

609   In reality, the plant growth rate, as described with Δd, gradually increases up to a specific

610   temperature, then rapidly declines to a supra-optimal level. Such response can be well described

611   by a beta function with uni-modality and non-symmetrical shape. Three parameters are involved

612   in DA: minimum temperature (T_n), optimal temperature (T_o) and forcing requirement (R_d). The

613   other parameter values are fixed with empirical values. For example, maximum growth rate (R_x)

614   is set to one and maximum temperature (T_x) is assumed to be 45.

615
$$r_d = R_x \left(\frac{T_x-T_d}{T_x-T_o}\right)\left(\frac{T_d-T_n}{T_o-T_n}\right)^{\frac{T_o-T_n}{T_x-T_o}} \tag{A5}$$

616
$$\Delta d = \begin{cases} r_d \ if \ r_d > 0 \\ 0 \ otherwise \end{cases} \tag{A6}$$

617
$$\sum_{d=D_s}^{\widehat{D}-1} \Delta d < R_d \leq \sum_{d=D_s}^{\widehat{D}} \Delta d \tag{A7}$$

618   4. Days transferred to standard temperature (DTS)

619   According to Arrhenius las, the relationship between growth rate and daily temperature T_d can

620   be interpolated by the equation 8 (Ono and Konno, 1999). With a factor weighted by standard

621   temperature, the equation for DTS (Eq. A9) can better represent growth rate dependent on

622   temperatures. Three parameters considered in DA are: temperature sensitivity rate (E_a), standard

623   temperature (T_s) and forcing requirement (R_d).

624
$$k = e^{\frac{-E_a}{R \cdot T_d}} \tag{A8}$$

625
$$\Delta d = e^{\frac{E_a(T_d - T_s)}{R \cdot T_d \cdot T_s}} \tag{A9}$$

626
$$\sum_{d=D_s}^{\widehat{D}-1} \Delta d < R_d \leq \sum_{d=D_s}^{\widehat{D}} \Delta d \tag{A10}$$

627   5. Thermal period fixed model (TP)

628   The difference between GD and TP models are heat accumulation occurs in a fixed time period

629   (D_n). The day of leaf onset is the last day ($\widehat{D_s} + D_n$) when the accumulated heat reaches the

630   forcing requirement. The start day ($\widehat{D_s}$) of heat accumulation begins in day one and moves one

631   day forward each time to estimate Eq. (A12). Three parameters are involved in DA: the base

632   temperature (T_b), the period length (D_n) and the forcing requirement (R_d).

633
$$\Delta d = \begin{cases} T_d - T_b \ if \ T_d > T_b \\ 0 \quad otherwise \end{cases} \tag{A11}$$

634
$$R_d \leq \sum_{d=\widehat{D_s}}^{\widehat{D_s}+D_n} \Delta d \tag{A12}$$

635   6. Chilling and forcing (CF)

636 Compared to GD, there is another distinctive chilling period for dormancy. CF model

637 sequentially calculates two accumulations in opposite directions: chilling accumulation and anti-

638 chilling accumulation. The start day of chilling accumulation (D_s) is implicitly set as 273.0

639 which is October 1st. The end day of chilling accumulation (D_0) is the beginning of anti-chilling

640 accumulation. Three parameters are considered in DA: the chilling requirement (R_d^C) and the

641 forcing requirement (R_d^F), the temperature threshold (T_c).

$$\Delta d = \begin{cases} T_d - T_c & if \ T_d \geq 0 \\ -T_c & otherwise \end{cases} \tag{A13}$$

$$\Delta_d^C = \begin{cases} \Delta d & if \ \Delta d < 0 \\ 0 & otherwise \end{cases} \tag{A14}$$

$$\Delta_d^F = \begin{cases} \Delta d & if \ \Delta d > 0 \\ 0 & otherwise \end{cases} \tag{A15}$$

$$\sum_{d=D_s}^{D_0-1} \Delta_d^C > R_d^C \geq \sum_{d=D_s}^{D_0} \Delta_d^C \tag{A16}$$

$$\sum_{d=D_0}^{\hat{D}-1} \Delta_d^F < R_d^F \leq \sum_{d=D_0}^{\hat{D}} \Delta_d^F \tag{A17}$$

647 7. Sequential model (SM)

648 The difference between CF and SM models is that SM used a beta function (Eq. A18) for the

649 calculation of chilling accumulation and adopted a sigmoid function (Eq. A20) for anti-chilling

650 accumulation. The detailed descriptions of these two functions can be referred to the

651 introductions of BF model and CF model. The maximum temperature is empirically set as

652 13.7695. Six parameters are constrained in DA: minimum temperature (T_n), optimal temperature

653 (T_o), temperature sensitivity (S_t), forcing base temperature (T_b), chilling requirement (R_d^C), and

654 forcing requirement (R_d^F).

$$r_d = (\frac{T_x - T_d}{T_x - T_o})(\frac{T_d - T_n}{T_o - T_n})^{\frac{T_o - T_n}{T_x - T_o}} \tag{A18}$$

$$\Delta_d^C = \begin{cases} r_d & if \ r_d < 0 \\ 0 & otherwise \end{cases} \tag{A19}$$

657
$$\Delta_d^F = \frac{1}{1+e^{S_t(T_d-T_b)}}$$
(A20)

658
$$\sum_{d=D_s}^{D_0-1}\Delta_d^C > R_d^C \geq \sum_{d=D_s}^{D_0}\Delta_d^C$$
(A21)

659
$$\sum_{d=D_0}^{\hat{D}-1}\Delta_d^F < R_d^F \leq \sum_{d=D_0}^{\hat{D}}\Delta_d^F$$
(A22)

660   8. Parallel model (PM)

661   Critical difference between PM and above two-step models is that the chilling and anti-chilling

662   accumulations happen simultaneously (Fu et al., 2012). In the earlier dates during chilling

663   period, only small fraction (K_d) of forcing (Eq. A25) will be accumulated. The maximum

664   temperature is empirically set as 15.3. Seven parameters will be considered in DA: minimum

665   temperature (T_n), optimal temperature (T_o), temperature sensitivity (S_t), forcing base temperature

666   (T_b), chilling requirement (R_d^C), forcing requirement (R_d^F), and a forcing weight coefficient (K_m).

667
$$r_d = \left(\frac{T_x-T_d}{T_x-T_o}\right)\left(\frac{T_d-T_n}{T_o-T_n}\right)^{\frac{T_o-T_n}{T_x-T_o}}$$
(A23)

668
$$\Delta_d^C = \begin{cases} r_d & if\ r_d < 0 \\ 0 & otherwise \end{cases}$$
(A24)

669
$$K_d = \begin{cases} K_m + (1-K_m)\frac{\sum_{i=D_s}^{d}\Delta_i^C}{R_d^C} & if\ \sum_{d=D_s}^{D_0-1}\Delta_d^C > R_d^C \\ 1 & otherwise \end{cases}$$
(A25)

670
$$\Delta_d^F = \frac{K_d}{1+e^{S_t(T_d-T_b)}}$$
(A26)

671
$$\sum_{d=D_s}^{D_0-1}\Delta_d^C > R_d^C \geq \sum_{d=D_s}^{D_0}\Delta_d^C$$
(A27)

672
$$\sum_{d=D_0}^{\hat{D}-1}\Delta_d^F < R_d^F \leq \sum_{d=D_0}^{\hat{D}}\Delta_d^F$$
(A28)

673   9. Alternating model (AM)

674   AM fixes the start date of chilling period (D_s^C) as November 1st and the start date of anti-chilling

675   period (D_s^F) as January 1st. The difference between AM and the other models above is that the

676   forcing requirement is not a parameter value but is decided by the length of chilling days (Fu et

677   al., 2012). Five parameters to be constrained in DA are: chilling temperature (T_c), forcing base

678   temperature (T_b) and three coefficients (a, b, c) in calculation of forcing requirement.

679
$$\Delta_d^C = \begin{cases} 1 \ if \ T_d \leq T_c \\ 0 \ otherwise \end{cases}$$
(A29)

680
$$\Delta_d^F = \begin{cases} T_d - T_b \ if \ T_d > T_b \\ 0 \quad otherwise \end{cases}$$
(A30)

681
$$R_d^C = \sum_{i=D_S^C}^{d} \Delta_i^C$$
(A31)

682
$$R_d^F = a + b \cdot e^{-c \cdot R_d^C}$$
(A32)

683
$$\sum_{d=D_S^F}^{\hat{D}-1} \Delta_d^F < R_d^F \leq \sum_{d=D_S^F}^{\hat{D}} \Delta_d^F$$
(A33)

684

685　Table A1: A summary of parameters to be calibrated in nine phenological models. Their default

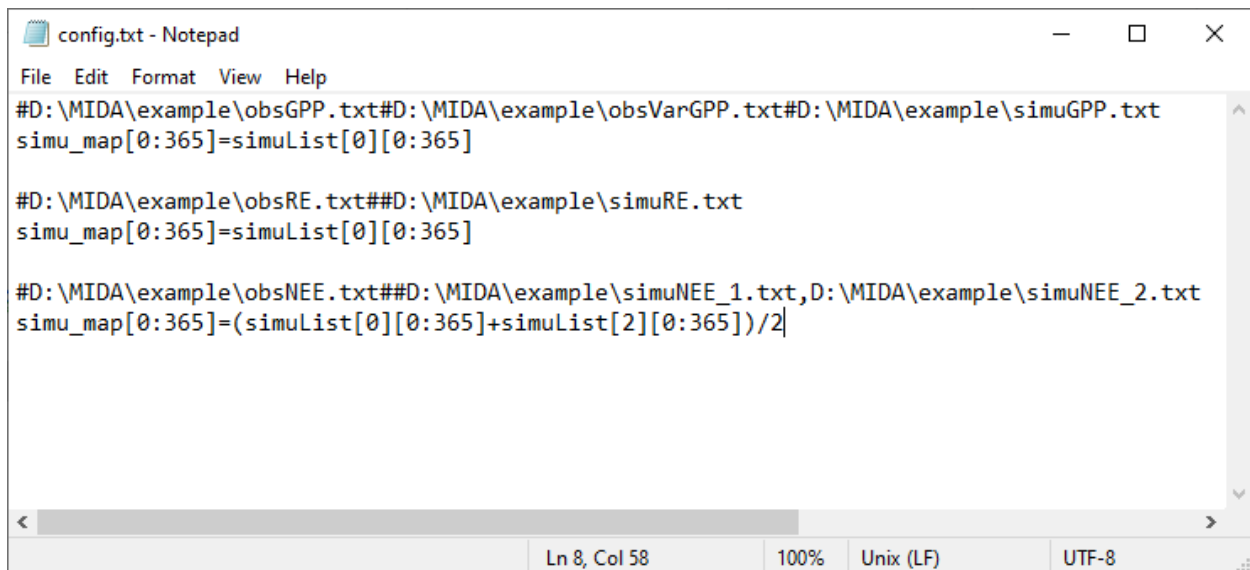686　parameter value and prior parameter range are shown.

Model	Parameter	Description	Unit	Default	Range
GD	T_b	Base temperature	°C	10	[-5, 25]
	R_d	Forcing requirement	°Cd	35	[0, 200]
SF	T_b	Base temperature	°C	-1.5	[-10, 25]
	R_d	Forcing requirement	°C	50	[0, 500]
BF	T_o	Optimal temperature	°C	15	[10, 35]
	T_n	Minimum temperature	°C	0	[-10, 5]
	R_d	Forcing requirement	°Cd	11	[0, 50]
DTS	E_a	Temperature sensitivity rate	-	250	[1, 1500]
	T_s	Standard temperature	°C	10	[-30, 40]
	R_d	Forcing requirement	°Cd	50	[1, 200]
TP	T_b	Base temperature	°C	12.5	[0, 30]
	D_n	Period length	d	25	[0, 50]
	R_d	Forcing requirement	°Cd	20	[0, 150]
CF	R_d^C	Chilling requirement	°Cd	-124	[-300, 0]
	R_d^F	Forcing requirement	°Cd	120	[0, 300]
	T_c	Chilling base temperature	°C	5	[0, 30]
SM	T_n	Minimum temperature	°C	-20	[-80, 0]
	T_o	Optimal temperature	°C	0	[-26, 10]
	S_t	Temperature sensitivity	-	-1.8	[-5, 0]
	T_b	Forcing base temperature	°C	5	[-5, 35]
	R_d^C	Chilling requirement	°Cd	20	[0, 80]
	R_d^F	Forcing requirement	°Cd	20	[0, 80]
PM	T_n	Minimum temperature	°C	-20	[-80, 0]
	T_o	Optimal temperature	°C	0	[-26, 10]
	S_t	Temperature sensitivity	-	-0.6	[-1, 0]
	T_b	Forcing base temperature	°C	5	[-5, 35]
	R_d^C	Chilling requirement	°Cd	11.35	[0, 80]
	R_d^F	Forcing requirement	°Cd	44.01	[0, 80]
	K_m	Forcing weight coefficient	-	0.2	[0, 1]
AM	T_c	Chilling base temperature	°C	4.6	[-10, 10]
	T_b	Forcing base temperature	°C	5	[-5, 35]
	a	Coefficient for forcing adjustment	-	11.51	[0.01, 15]
	b	Coefficient for forcing adjustment	-	88	[0, 200]
	c	Coefficient for forcing adjustment	-	-0.01	$[-1, -10^{-4}]$

687

688

689 **Appendix B:** An example of output configuration file

690 Output configuration file (e.g., config.txt) is to indicate the directories of observations and

691 simulation output files as well as how they map to each other. Figure B1 is an example of the

692 output configuration file. There are three blocks of functions to map simulation outputs to

693 observed GPP, RE, and NEE. The blocks of mapping functions are separated by a blank line.

694 Each mapping block starts with the directories of one observation, its observation variance and

695 model outputs, which are separated by a hash key. If there is no observation variance available,

696 users can ignore this directory. If multiple simulation outputs are used to correspond to one

697 observation, the directories of simulation outputs are separated by a comma. The rest of the

698 mapping block describes how to map simulation outputs to observations. The simu_map variable

699 is simulation output after mapping. The simuList variable saves the simulation outputs specified

700 in the first line. Taking the third mapping block in Fig. B1 as an example, simuList[0] saves

701 contents in simuNEE_1.txt and simuList[0][0:365] saves the first 365 elements in this file.


```
config.txt - Notepad                                            —    □    ×

File   Edit   Format   View   Help
#D:\MIDA\example\obsGPP.txt#D:\MIDA\example\obsVarGPP.txt#D:\MIDA\example\simuGPP.txt
simu_map[0:365]=simuList[0][0:365]

#D:\MIDA\example\obsRE.txt##D:\MIDA\example\simuRE.txt
simu_map[0:365]=simuList[0][0:365]

#D:\MIDA\example\obsNEE.txt##D:\MIDA\example\simuNEE_1.txt,D:\MIDA\example\simuNEE_2.txt
simu_map[0:365]=(simuList[0][0:365]+simuList[2][0:365])/2



                                   Ln 8, Col 58       100%   Unix (LF)      UTF-8
```

702

703 Figure B1: An example of output configuration file
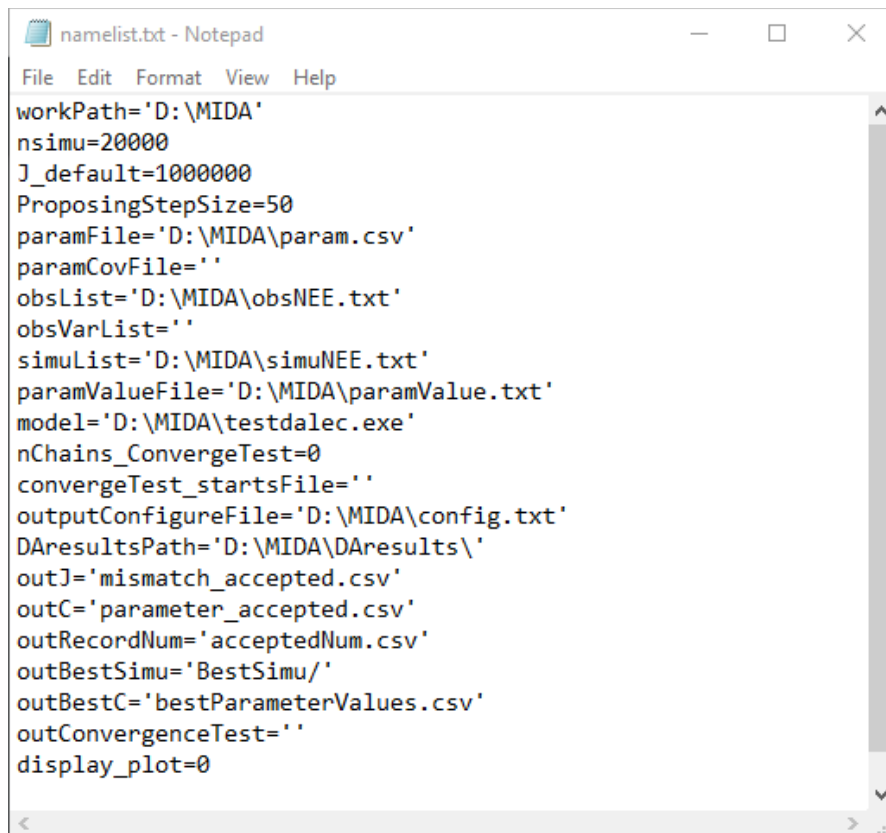
704   **Appendix C:** An example of the namelist.txt file

705   The Fig. C1 shows an example of the namelist.txt for the first study case with the DALEC

706   model. Users need to prepare the namelist.txt before execution of data assimilation (DA) either

707   manually or via GUI. Below describes the content in the namelist.txt. Detailed explanation or

708   tutorials are available in the Zenodo repositories at the end of the appendixes.

709   'workpath' is the directory where the MIDA executable are saved. 'nsimu' is the number

710   of iterations in execution of data assimilation. 'J_default' is the default mismatch (i.e., cost

711   function) to be compared in the first moving phase of data assimilation. 'ProposingStepSize'

712   controls the jump scale in the proposing phase of data assimilation. Users can increase or

713   decrease this value to adjust the acceptance rate to be in a range from 0.2 to 0.5. 'paramFile' is

714   the directory of a csv file saving parameter-related information such as parameter range.

715   'obsList' saves the directories of observations. Multiple observations are separated by semicolon.

716   Similarly, 'obsVarList' saves the directories of observation variance in the same order as that of

717   'obsList'. 'simuList' saves the directories of simulation outputs corresponding to the

718   observations. With GUI, MIDA reads directories in the output configuration file (e.g., config.txt)

719   which users provide and assign values for 'obsList','obsVarList', and 'simuList' in the

720   namelist.txt automatically. In this case, if the directories of observations change, users only need

721   to modify the output configuration file and generate the namelist.txt again with GUI-based

722   MIDA.

723   'paramValue' is the directory of a txt file where MIDA writes out new set of parameter

724   values for model execution in each iteration of data assimilation. Its default value is

725   'ParameterValue.txt' under the workpath specified in the first line of the namelist.txt. 'model'

726   saves the directory of model executable. 'nChains_convergeTest' indicates whether to conduct

727   German-Rubin (G-R) convergence test or not. If G-R test is used, its values is the number of

728   multiple MCMC chains. If not, its value is zero. 'convergeTest_startsFile' is the directory of a

729   csv file saving default parameter values as the start points in multiple MCMC chains.

730   'outConvergenceTest' saves the results of G-R test. If 'nChains_ConvergeTest' is zero, both

731   values of 'convergeTest_startsFile' and 'outConvergenceTest' are empty. 'DAresultsPath' is the

732   directory saving the results of DA whose directories are also listed in the following six lines:

733   'outJ' for the accepted mismatches; 'outC' for the accepted parameter values; 'outRecordNum'

734   for the number of accepted parameter values; 'outBestSimu' for the best simulation outputs with

735   the optimal parameter values; 'outBestC' for the optimal parameter values. For MIDA without

736   GUI, 'display_plot' indicates whether or not to visualize the posterior distributions after DA.


```
namelist.txt - Notepad                              —    □    ×
File  Edit  Format  View  Help
workPath='D:\MIDA'
nsimu=20000
J_default=1000000
ProposingStepSize=50
paramFile='D:\MIDA\param.csv'
paramCovFile=''
obsList='D:\MIDA\obsNEE.txt'
obsVarList=''
simuList='D:\MIDA\simuNEE.txt'
paramValueFile='D:\MIDA\paramValue.txt'
model='D:\MIDA\testdalec.exe'
nChains_ConvergeTest=0
convergeTest_startsFile=''
outputConfigureFile='D:\MIDA\config.txt'
DAresultsPath='D:\MIDA\DAresults\'
outJ='mismatch_accepted.csv'
outC='parameter_accepted.csv'
outRecordNum='acceptedNum.csv'
outBestSimu='BestSimu/'
outBestC='bestParameterValues.csv'
outConvergenceTest=''
display_plot=0
```

737

738   **Figure C1**. An example of the 'namelist.txt' file. In order to use MIDA, users need to prepare

739   data and a model and specify their file names and directories in the 'namelist.txt' file.

740　*Code and data availability.* The code of MIDA is available at the Zenodo repository

741　https://doi.org/10.5281/zenodo.4762725 (last access: May 2021). Data used in this study are

742　available at https://doi.org/10.5281/zenodo.4762779. A comparison of the time cost using the

743　embedded DA algorithm and MIDA is available at the Zenodo repository

744　https://doi.org/10.5281/zenodo.4891319.

745

746　*Video supplement.* Tutorial videos of how to use MIDA is available at

747　https://doi.org/10.5281/zenodo.4762777

748

749　*Author contributions.* XH, IS, and YL designed the study. XH built the workflow of MIDA and

750　tested its capability in four cases. DL, DMR, and PJH provided data and model for the first and

751　second test cases. XL prepared models and ADR provided observations for the third case. EW

752　and SN helped to prepare data and model for the fourth case. XH, LJ, EH and YL analyzed the

753　results. All authors contributed to the preparation of the manuscript.

754

755　*Competing interests.* The authors declare that they have no conflict of interest.

756

760

761　**References**

762 Allen, J. I., Eknes, M. and Evensen, G.: An Ensemble Kalman Filter with a complex marine

763 ecosystem model: hindcasting phytoplankton in the Cretan Sea, Ann. Geophys., 21(1), 399–411,

764 doi:10.5194/angeo-21-399-2003, 2003.

765 Anderson, J., Hoar, T., Raeder, K., Liu, H., Collins, N., Torn, R. and Avellano, A.: The data

766 assimilation research testbed a community facility, Bull. Am. Meteorol. Soc., 90(9), 1283–1296,

767 doi:10.1175/2009BAMS2618.1, 2009.

768 Bloom, A. A., Exbrayat, J. F., Van Der Velde, I. R., Feng, L. and Williams, M.: The decadal state of

769 the terrestrial carbon cycle: Global retrievals of terrestrial carbon allocation, pools, and residence

770 times, Proc. Natl. Acad. Sci., 113(5), 1285–1290, doi:10.1073/pnas.1515160113, 2016.

771 Bonan, G.: Climate Change and Terrestrial Ecosystem Modeling, Cambridge University Press.,

772 2019.

773 Box, G. E. P. and Tiao, G. C.: Bayesian Inference in Statistical Analysis, John Wiley & Sons, Inc.,

774 Hoboken, NJ, USA., 1992.

775 Ciais, P., Chris, S., Govindasamy, B., Bopp, L., Brovkin, V., Canadell, J., Chhabra, A., Defries, R.,

776 Galloway, J. and Heimann, M.: Carbon and other biogeochemical cycles, Clim. Chang. 2013

777 Phys. Sci. Basis, 465–570, 2013.

778 Cline, M. P., Lomow, G. and Girou, M.: C++ FAQs, Pearson Education., 1998.

779 Doherty, J.: PEST: Model independent parameter estimation. Fifth edition of user manual,

780 Watermark Numer. Comput., doi:10.1016/B978-0-08-098288-5.00031-2, 2004.

781 Evensen, G.: The Ensemble Kalman Filter: Theoretical formulation and practical implementation,

782 Ocean Dyn., 53(4), 343–367, doi:10.1007/s10236-003-0036-9, 2003.

783 Fer, I., Kelly, R., Moorcroft, P. R., Richardson, A. D., Cowdery, E. M. and Dietze, M. C.: Linking

784 big models to big data: Efficient ecosystem model calibration through Bayesian model

785 emulation, Biogeosciences, 15(19), 5801–5830, doi:10.5194/bg-15-5801-2018, 2018.

786 Fox, A., Williams, M., Richardson, A. D., Cameron, D., Gove, J. H., Quaife, T., Ricciuto, D.,

787   Reichstein, M., Tomelleri, E., Trudinger, C. M. and Van Wijk, M. T.: The REFLEX project:

788   Comparing different algorithms and implementations for the inversion of a terrestrial ecosystem

789   model against eddy covariance data, Agric. For. Meteorol., 149(10), 1597–1615,

790   doi:10.1016/j.agrformet.2009.05.002, 2009.

791 Fox, A. M., Hoar, T. J., Anderson, J. L., Arellano, A. F., Smith, W. K., Litvak, M. E., MacBean, N.,

792   Schimel, D. S. and Moore, D. J. P.: Evaluation of a Data Assimilation System for Land Surface

793   Models Using CLM4.5, J. Adv. Model. Earth Syst., 10(10), 2471–2494,

794   doi:10.1029/2018MS001362, 2018.

795 Friedlingstein, P., Cox, P., Betts, R., Bopp, L., von Bloh, W., Brovkin, V., Cadule, P., Doney, S.,

796   Eby, M., Fung, I., Bala, G., John, J., Jones, C., Joos, F., Kato, T., Kawamiya, M., Knorr, W.,

797   Lindsay, K., Matthews, H. D., Raddatz, T., Rayner, P., Reick, C., Roeckner, E., Schnitzler, K.-

798   G., Schnur, R., Strassmann, K., Weaver, A. J., Yoshikawa, C. and Zeng, N.: Climate–Carbon

799   Cycle Feedback Analysis: Results from the C4MIP Model Intercomparison, J. Clim., 19(14),

800   3337–3353, doi:10.1175/JCLI3800.1, 2006.

801 Fu, Y. H., Campioli, M., Van Oijen, M., Deckmyn, G. and Janssens, I. A.: Bayesian comparison of

802   six different temperature-based budburst models for four temperate tree species, Ecol. Modell.,

803   230, 92–100, doi:10.1016/j.ecolmodel.2012.01.010, 2012.

804 Gao, C., Wang, H., Weng, E., Lakshmivarahan, S., Zhang, Y. and Luo, Y.: Assimilation of multiple

805   data sets with the ensemble Kalman filter to improve forecasts of forest carbon dynamics, Ecol.

806   Appl., 21(5), 1461–1473, doi:10.1890/09-1234.1, 2011.

807 Gelman, A. and Rubin, D. B.: Inference from Iterative Simulation Using Multiple Sequences, Stat.

808   Sci., 7(4), 457–472, doi:10.1214/SS/1177011136, 1992.

809 Hanson, P. J., Riggs, J. S., Nettles, W. R., Phillips, J. R., Krassovski, M. B., Hook, L. A., Gu, L.,

810   Richardson, A. D., Aubrecht, D. M., Ricciuto, D. M., Warren, J. M. and Barbier, C.: Attaining

811   whole-ecosystem warming using air and deep-soil heating methods with an elevated CO_2

812   atmosphere, Biogeosciences, 14(4), 861–883, doi:10.5194/bg-14-861-2017, 2017.

813 Hararuk, O., Xia, J. and Luo, Y.: Evaluation and improvement of a global land model against soil

814   carbon data using a Bayesian Markov chain Monte Carlo method, J. Geophys. Res.

815   Biogeosciences, 119(3), 403–417, doi:10.1002/2013JG002535, 2014.

816 Hararuk, O., Smith, M. J. and Luo, Y.: Microbial models with data-driven parameters predict

817   stronger soil carbon responses to climate change, Glob. Chang. Biol., 21(6), 2439–2453,

818   doi:10.1111/gcb.12827, 2015.

819 Hastings, W. K.: Monte carlo sampling methods using Markov chains and their applications,

820   Biometrika, 57(1), 97–109, doi:10.1093/biomet/57.1.97, 1970.

821 Hou, E., Lu, X., Jiang, L., Wen, D. and Luo, Y.: Quantifying Soil Phosphorus Dynamics: A Data

822   Assimilation Approach, J. Geophys. Res. Biogeosciences, 124(7), 2159–2173,

823   doi:10.1029/2018JG004903, 2019.

824 Ise, T. and Moorcroft, P. R.: The global-scale temperature and moisture dependencies of soil

825   organic carbon decomposition: An analysis using a mechanistic decomposition model,

826   Biogeochemistry, 80(3), 217–231, doi:10.1007/s10533-006-9019-5, 2006.

827 Iversen, C. M., McCormack, M. L., Powell, A. S., Blackwood, C. B., Freschet, G. T., Kattge, J.,

828   Roumet, C., Stover, D. B., Soudzilovskaia, N. A., Valverde-Barrantes, O. J., van Bodegom, P.

829   M. and Violle, C.: A global Fine-Root Ecology Database to address below-ground challenges in

830   plant ecology, New Phytol., 215(1), 15–26, doi:10.1111/nph.14486, 2017.

831 Jiang, J., Huang, Y., Ma, S., Stacy, M., Shi, Z., Ricciuto, D. M., Hanson, P. J. and Luo, Y.:

832 Forecasting Responses of a Northern Peatland Carbon Cycle to Elevated CO2 and a Gradient of

833 Experimental Warming, J. Geophys. Res. Biogeosciences, 123(3), 1057–1071,

834 doi:10.1002/2017JG004040, 2018.

835 Kattge, J., Bönisch, G., Díaz, S., Lavorel, S., Prentice, I. C., Leadley, P., Tautenhahn, S., Werner,

836 G. D. A., Aakala, T., Abedi, M., Acosta, A. T. R., Adamidis, G. C., Adamson, K., Aiba, M.,

837 Albert, C. H., Alcántara, J. M., Alcázar C, C., Aleixo, I., Ali, H., Amiaud, B., Ammer, C.,

838 Amoroso, M. M., Anand, M., Anderson, C., Anten, N., Antos, J., Apgaua, D. M. G., Ashman, T.

839 L., Asmara, D. H., Asner, G. P., Aspinwall, M., Atkin, O., Aubin, I., Baastrup-Spohr, L.,

840 Bahalkeh, K., Bahn, M., Baker, T., Baker, W. J., Bakker, J. P., Baldocchi, D., Baltzer, J.,

841 Banerjee, A., Baranger, A., Barlow, J., Barneche, D. R., Baruch, Z., Bastianelli, D., Battles, J.,

842 Bauerle, W., Bauters, M., Bazzato, E., Beckmann, M., Beeckman, H., Beierkuhnlein, C., Bekker,

843 R., Belfry, G., Belluau, M., Beloiu, M., Benavides, R., Benomar, L., Berdugo-Lattke, M. L.,

844 Berenguer, E., Bergamin, R., Bergmann, J., Bergmann Carlucci, M., Berner, L., Bernhardt-

845 Römermann, M., Bigler, C., Bjorkman, A. D., Blackman, C., Blanco, C., Blonder, B.,

846 Blumenthal, D., Bocanegra-González, K. T., Boeckx, P., Bohlman, S., Böhning-Gaese, K.,

847 Boisvert-Marsh, L., Bond, W., Bond-Lamberty, B., Boom, A., Boonman, C. C. F., Bordin, K.,

848 Boughton, E. H., Boukili, V., Bowman, D. M. J. S., Bravo, S., Brendel, M. R., Broadley, M. R.,

849 Brown, K. A., Bruelheide, H., Brumnich, F., Bruun, H. H., Bruy, D., Buchanan, S. W., Bucher,

850 S. F., Buchmann, N., Buitenwerf, R., Bunker, D. E., et al.: TRY plant trait database – enhanced

851 coverage and open access, Glob. Chang. Biol., 26(1), 119–188, doi:10.1111/gcb.14904, 2020.

852 De Kauwe, M. G., Medlyn, B. E., Walker, A. P., Zaehle, S., Asao, S., Guenet, B., Harper, A. B.,

853 Hickler, T., Jain, A. K., Luo, Y., Lu, X., Luus, K., Parton, W. J., Shu, S., Wang, Y. P., Werner,

854 C., Xia, J., Pendall, E., Morgan, J. A., Ryan, E. M., Carrillo, Y., Dijkstra, F. A., Zelikova, T. J.

855 and Norby, R. J.: Challenging terrestrial biosphere models with data from the long-term

856 multifactor Prairie Heating and CO2 Enrichment experiment, Glob. Chang. Biol., 23(9), 3623–

857 3645, doi:10.1111/gcb.13643, 2017.

858 Keenan, T. F., Davidson, E., Moffat, A. M., Munger, W. and Richardson, A. D.: Using model-data

859 fusion to interpret past trends, and quantify uncertainties in future projections, of terrestrial

860 ecosystem carbon cycling, Glob. Chang. Biol., 18(8), 2555–2569, doi:10.1111/j.1365-

861 2486.2012.02684.x, 2012.

862 Keenan, T. F., Davidson, E. A., Munger, J. W. and Richardson, A. D.: Rate my data: Quantifying

863 the value of ecological data for the development of models of the terrestrial carbon cycle, Ecol.

864 Appl., 23(1), 273–286, doi:10.1890/12-0747.1, 2013.

865 Lawrence, D. M., Fisher, R. A., Koven, C. D., Oleson, K. W., Swenson, S. C., Bonan, G., Collier,

866 N., Ghimire, B., van Kampenhout, L., Kennedy, D., Kluzek, E., Lawrence, P. J., Li, F., Li, H.,

867 Lombardozzi, D., Riley, W. J., Sacks, W. J., Shi, M., Vertenstein, M., Wieder, W. R., Xu, C.,

868 Ali, A. A., Badger, A. M., Bisht, G., van den Broeke, M., Brunke, M. A., Burns, S. P., Buzan, J.,

869 Clark, M., Craig, A., Dahlin, K., Drewniak, B., Fisher, J. B., Flanner, M., Fox, A. M., Gentine,

870 P., Hoffman, F., Keppel-Aleks, G., Knox, R., Kumar, S., Lenaerts, J., Leung, L. R., Lipscomb,

871 W. H., Lu, Y., Pandey, A., Pelletier, J. D., Perket, J., Randerson, J. T., Ricciuto, D. M.,

872 Sanderson, B. M., Slater, A., Subin, Z. M., Tang, J., Thomas, R. Q., Val Martin, M. and Zeng,

873 X.: The Community Land Model Version 5: Description of New Features, Benchmarking, and

874 Impact of Forcing Uncertainty, J. Adv. Model. Earth Syst., 11(12), 4245–4287,

875 doi:10.1029/2018MS001583, 2019.

876 LeBauer, D. S., Wang, D., Richter, K. T., Davidson, C. C. and Dietze, M. C.: Facilitating feedbacks

877 between field measurements and ecosystem models, Ecol. Monogr., 83(2), 133–154,

878   doi:10.1890/12-0137.1, 2013.

879 Levenberg, K.: A method for the solution of certain non-linear problems in least squares, Q. Appl.

880   Math., 2(2), 164–168, 1944.

881 Li, Q., Lu, X., Wang, Y., Huang, X., Cox, P. M. and Luo, Y.: Leaf area index identified as a major

882   source of variability in modeled CO2 fertilization, Biogeosciences, 15(22), 6909–6925,

883   doi:10.5194/bg-15-6909-2018, 2018.

884 Liang, J., Zhou, Z., Huo, C., Shi, Z., Cole, J. R., Huang, L., Konstantinidis, K. T., Li, X., Liu, B.,

885   Luo, Z., Penton, C. R., Schuur, E. A. G., Tiedje, J. M., Wang, Y. P., Wu, L., Xia, J., Zhou, J. and

886   Luo, Y.: More replenishment than priming loss of soil organic carbon with additional carbon

887   input, Nat. Commun., 9(1), 1–9, doi:10.1038/s41467-018-05667-7, 2018a.

888 Liang, J., Zhou, Z., Huo, C., Shi, Z., Cole, J. R., Huang, L., Konstantinidis, K. T., Li, X., Liu, B.,

889   Luo, Z., Penton, C. R., Schuur, E. A. G., Tiedje, J. M., Wang, Y., Wu, L. and Xia, J.: organic

890   carbon with additional carbon input, Nat. Commun., 1–9, doi:10.1038/s41467-018-05667-7,

891   2018b.

892 Lu, D., Ricciuto, D., Walker, A., Safta, C. and Munger, W.: Bayesian calibration of terrestrial

893   ecosystem models: A study of advanced Markov chain Monte Carlo methods, Biogeosciences,

894   14(18), 4295–4314, doi:10.5194/bg-14-4295-2017, 2017.

895 Lu, D., Ricciuto, D., Stoyanov, M. and Gu, L.: Calibration of the E3SM Land Model Using

896   Surrogate-Based Global Optimization, J. Adv. Model. Earth Syst., 10(6), 1337–1356,

897   doi:10.1002/2017MS001134, 2018.

898 Luo, Y. and Schuur, E. A. G.: Model parameterization to represent processes at unresolved scales

899   and changing properties of evolving systems, Glob. Chang. Biol., 26(3), 1109–1117,

900   doi:10.1111/gcb.14939, 2020.

901 Luo, Y., Wu, L., Andrews, J. A., White, L., Matamala, R., Schäfer, K. V. R. and Schlesinger, W.

902   H.: ELEVATED CO2 DIFFERENTIATES ECOSYSTEM CARBON PROCESSES:

903   DECONVOLUTION ANALYSIS OF DUKE FOREST FACE DATA, Ecol. Monogr., 71(3),

904   357–376, doi:10.1890/0012-9615(2001)071[0357:ECDECP]2.0.CO;2, 2001.

905 Luo, Y., Ogle, K., Tucker, C., Fei, S., Gao, C., LaDeau, S., Clark, J. S. and Schimel, D. S.:

906   Ecological forecasting and data assimilation in a data-rich era, Ecol. Appl., 21(5), 1429–1442,

907   doi:10.1890/09-1275.1, 2011.

908 Ma, S., Jiang, J., Huang, Y., Shi, Z., Wilson, R. M., Ricciuto, D., Sebestyen, S. D., Hanson, P. J.

909   and Luo, Y.: Data-Constrained Projections of Methane Fluxes in a Northern Minnesota Peatland

910   in Response to Elevated CO2and Warming, J. Geophys. Res. Biogeosciences, 122(11), 2841–

911   2861, doi:10.1002/2017JG003932, 2017.

912 Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E.: Equation of

913   state calculations by fast computing machines, J. Chem. Phys., 21(6), 1087–1092,

914   doi:10.1063/1.1699114, 1953.

915 Mitchell, J. C. and Apt, K.: Concepts in programming languages, Cambridge University Press.,

916   2003.

917 Nerger, L. and Hiller, W.: Software for ensemble-based data assimilation systems-Implementation

918   strategies and scalability, Comput. Geosci., 55, 110–118, doi:10.1016/j.cageo.2012.03.026,

919   2013.

920 Van Oijen, M., Cameron, D. R., Butterbach-Bahl, K., Farahbakhshazad, N., Jansson, P. E., Kiese,

921   R., Rahn, K. H., Werner, C. and Yeluripati, J. B.: A Bayesian framework for model calibration,

922   comparison and analysis: Application to four models for the biogeochemistry of a Norway

923   spruce forest, Agric. For. Meteorol., 151(12), 1609–1621, doi:10.1016/j.agrformet.2011.06.017,

924   2011.

925 Ono, S. and Konno, T.: Estimation of flowering date and temperature characteristics of fruit trees by

926   DTS method, Japan Agric. Res. Q., 33(2), 105–108, 1999.

927 Raeder, K., Anderson, J. L., Collins, N., Hoar, T. J., Kay, J. E., Lauritzen, P. H. and Pincus, R.:

928   DART/CAM: An ensemble data assimilation system for CESM atmospheric models, J. Clim.,

929   25(18), 6304–6317, doi:10.1175/JCLI-D-11-00395.1, 2012.

930 Raupach, M. R., Rayner, P. J., Barrett, D. J., Defries, R. S., Heimann, M., Ojima, D. S., Quegan, S.

931   and Schmullius, C. C.: Model-data synthesis in terrestrial carbon observation: Methods, data

932   requirements and data uncertainty specifications, Glob. Chang. Biol., 11(3), 378–397,

933   doi:10.1111/j.1365-2486.2005.00917.x, 2005.

934 Rayner, P. J., Scholze, M., Knorr, W., Kaminski, T., Giering, R. and Widmann, H.: Two decades of

935   terrestrial carbon fluxes from a carbon cycle data assimilation system (CCDAS), Global

936   Biogeochem. Cycles, 19(2), n/a-n/a, doi:10.1029/2004GB002254, 2005.

937 Ricciuto, D., Sargsyan, K. and Thornton, P.: The Impact of Parametric Uncertainties on

938   Biogeochemistry in the E3SM Land Model, J. Adv. Model. Earth Syst., 10(2), 297–319,

939   doi:10.1002/2017MS000962, 2018.

940 Ricciuto, D. M., King, A. W., Dragoni, D. and Post, W. M.: Parameter and prediction uncertainty in

941   an optimized terrestrial carbon cycle model: Effects of constraining variables and data record

942   length, J. Geophys. Res., 116(G1), G01033, doi:10.1029/2010JG001400, 2011.

943 Richardson, A. D., Williams, M., Hollinger, D. Y., Moore, D. J. P., Dail, D. B., Davidson, E. A.,

944   Scott, N. A., Evans, R. S., Hughes, H., Lee, J. T., Rodrigues, C. and Savage, K.: Estimating

945   parameters of a forest ecosystem C model with measurements of stocks and fluxes as joint

946   constraints, Oecologia, 164(1), 25–40, doi:10.1007/s00442-010-1628-y, 2010.

947 Richardson, A. D., Hufkens, K., Milliman, T., Aubrecht, D. M., Chen, M., Gray, J. M., Johnston,

948   M. R., Keenan, T. F., Klosterman, S. T., Kosmala, M., Melaas, E. K., Friedl, M. A. and Frolking,

949   S.: Tracking vegetation phenology across diverse North American biomes using PhenoCam

950   imagery, Sci. Data, 5, 1–24, doi:10.1038/sdata.2018.28, 2018.

951 Ridler, M. E., Van Velzen, N., Hummel, S., Sandholt, I., Falk, A. K., Heemink, A. and Madsen, H.:

952   Data assimilation framework: Linking an open data assimilation library (OpenDA) to a widely

953   adopted model interface (OpenMI), Environ. Model. Softw., 57, 76–89,

954   doi:10.1016/j.envsoft.2014.02.008, 2014.

955 Robert, C. and Casella, G.: Monte Carlo statistical methods, Springer Science & Business Media.,

956   2013.

957 Roberts, G. O., Gelman, A. and Gilks, W. R.: Weak convergence and optimal scaling of random

958   walk Metropolis algorithms, Ann. Appl. Probab., 7(1), 110–120,

959   doi:10.1214/AOAP/1034625254, 1997.

960 Safta, C., Ricciuto, D. M., Sargsyan, K., Debusschere, B., Najm, H. N., Williams, M. and Thornton,

961   P. E.: Global sensitivity analysis, probabilistic calibration, and predictive assessment for the data

962   assimilation linked ecosystem carbon model, Geosci. Model Dev., 8(7), 1899–1918,

963   doi:10.5194/gmd-8-1899-2015, 2015.

964 Scholze, M., Kaminski, T., Rayner, P., Knorr, W. and Giering, R.: Propagating uncertainty through

965   prognostic carbon cycle data assimilation system simulations, J. Geophys. Res., 112(D17),

966   D17305, doi:10.1029/2007JD008642, 2007.

967 Shi, Z., Crowell, S., Luo, Y. and Moore, B.: Model structures amplify uncertainty in predicted soil

968   carbon responses to climate change, Nat. Commun., 9(1), 1–11, doi:10.1038/s41467-018-04526-

969   9, 2018.

970 Smith, M. J., Purves, D. W., Vanderwel, M. C., Lyutsarev, V. and Emmott, S.: The climate

971   dependence of the terrestrial carbon cycle, including parameter and structural uncertainties,

972   Biogeosciences, 10(1), 583–606, doi:10.5194/bg-10-583-2013, 2013.

973 Strigul, N., Pristinski, D., Purves, D., Dushoff, J. and Pacala, S.: Scaling from trees to forests:

974   Tractable macroscopic equations for forest dynamics, Ecol. Monogr., 78(4), 523–545,

975   doi:10.1890/08-0082.1, 2008.

976 Tao, F., Zhou, Z., Huang, Y., Li, Q., Lu, X., Ma, S., Huang, X., Liang, Y., Hugelius, G., Jiang, L.,

977   Doughty, R., Ren, Z. and Luo, Y.: Deep Learning Optimizes Data-Driven Representation of Soil

978   Organic Carbon in Earth System Model Over the Conterminous United States, Front. Big Data,

979   3(June), 1–15, doi:10.3389/fdata.2020.00017, 2020.

980 Trudinger, C. M., Raupach, M. R., Rayner, P. J., Kattge, J., Liu, Q., Park, B., Reichstein, M.,

981   Renzullo, L., Richardson, A. D., Roxburgh, S. H., Styles, J., Wang, Y. P., Briggs, P., Barrett, D.

982   and Nikolova, S.: OptIC project: An intercomparison of optimization techniques for parameter

983   estimation in terrestrial biogeochemical models, J. Geophys. Res. Biogeosciences, 112(2), 1–17,

984   doi:10.1029/2006JG000367, 2007.

985 Wang, Y. P., Trudinger, C. M. and Enting, I. G.: A review of applications of model-data fusion to

986   studies of terrestrial carbon fluxes at different scales, Agric. For. Meteorol., 149(11), 1829–1842,

987   doi:10.1016/j.agrformet.2009.07.009, 2009.

988 Weng, E. and Luo, Y.: Relative information contributions of model vs. data to short- and long-term

989   forecasts of forest carbon dynamics, Ecol. Appl., 21(5), 1490–1505, doi:10.1890/09-1394.1,

990   2011.

991 Weng, E., Dybzinski, R., Farrior, C. E. and Pacala, S. W.: Competition alters predicted forest

992   carbon cycle responses to nitrogen availability and elevated CO_2: simulations using an explicitly

993   competitive, game-theoretic vegetation demographic model, Biogeosciences Discuss., 1–35,

994   doi:10.5194/bg-2019-55, 2019.

995 Williams, M., Schwarz, P. A., Law, B. E., Irvine, J. and Kurpius, M. R.: An improved analysis of

996   forest carbon dynamics using data assimilation, Glob. Chang. Biol., 11(1), 89–105,

997   doi:10.1111/j.1365-2486.2004.00891.x, 2005.

998 Williams, M., Richardson, A. D., Reichstein, M., Stoy, P. C., Peylin, P., Verbeeck, H., Carvalhais,

999   N., Jung, M., Hollinger, D. Y., Kattge, J., Leuning, R., Luo, Y., Tomelleri, E., Trudinger, C. M.

1000   and Wang, Y. P.: Improving land surface models with FLUXNET data, Biogeosciences, 6(7),

1001   1341–1359, doi:10.5194/bg-6-1341-2009, 2009.

1002 Xu, T., White, L., Hui, D. and Luo, Y.: Probabilistic inversion of a terrestrial ecosystem model:

1003   Analysis of uncertainty in parameter estimation and model prediction, Global Biogeochem.

1004   Cycles, 20(2), 1–15, doi:10.1029/2005GB002468, 2006.

1005 Yun, K., Hsiao, J., Jung, M. P., Choi, I. T., Glenn, D. M., Shim, K. M. and Kim, S. H.: Can a multi-

1006   model ensemble improve phenology predictions for climate change studies?, Ecol. Modell., 362,

1007   54–64, doi:10.1016/j.ecolmodel.2017.08.003, 2017.

1008 Zobitz, J. M., Desai, A. R., Moore, D. J. P. and Chadwick, M. A.: A primer for data assimilation

1009   with ecological models using Markov Chain Monte Carlo (MCMC), Oecologia, 167(3), 599–

1010   611, doi:10.1007/s00442-011-2107-9, 2011.

1011

Table1: Comparison among MIDA and available DA tools

DA tool	Agnostic	DA algorithms	Global optima	Posterior distribution	Visualization
CCDAS	No	Automatic differentiation from Transformation of Algorithms in Fortran (TAF)	No	No	No
CARDAMOM	No	Markov Chain Monte Carlo	Yes	Yes	No
EcoPAD	No	Markov Chain Monte Carlo	Yes	Yes	Yes
OpenDA	No	EnKF, Ensemble Square-Root Filter, Particle Filter	Yes	Yes	No
DART	Yes	EnKF	Yes	Yes	No
PDAF	Yes	EnKF	Yes	Yes	No
PEST	Yes	Levenberg-Marquardt method	Rely on initial parameter values	No	No
MIDA	Yes	Markov Chain Monte Carlo	Yes	Yes	Yes

Table 2: A summary of 21 parameters to be calibrated in DALEC model. The default parameter value and prior parameter range are shown.

Parameter	Description	Unit	Default	Range
GDD_{min}	Growing degree day threshold for leaf out	$^{o}C\ d$	100	[10, 250]
GDD_{max}	Growing degree day threshold for maximum LAI	$^{o}C\ d$	200	[50, 500]
LAI_{max}	Seasonal maximum leaf area index	-	4	[2, 7]
$T_{leaffall}$	Temperature for leaf fall	^{o}C	5	[0, 10]
K_{leaf}	Rate of leaf fall	d^{-1}	0.1	[0.03 0.95]
NUE	N use efficiency	-	7	[1, 20]
Res_{growth}	Growth respiration fraction	-	0.2	[0.05, 0.5]
Res_m	Base rate for maintenance respiration	$\times 10^{-4}\ \mu mol\ m^{-2}d^{-1}$	1	[0.1, 100]
$Q_{10_{mr}}$	Maintenance respiration T-sensitivity	-	2	[1, 4]
A_{stem}	Allocation to plant stem pool	-	0.7	[0.1, 0.95]
τ_{root}	Root turnover time	$\times 10^{-4}\ d^{-1}$	5.48	[1.1, 27.4]
τ_{stem}	Stem turnover time	$\times 10^{-5}\ d^{-1}$	5.48	[1.1, 27.4]
$Q_{10_{hr}}$	Heterotrophic respiration T-sensitivity	-	2	[1, 4]
τ_{litter}	Base turnover for litter	$\times 10^{-3}\ umol\ m^{-2}d^{-1}$	1.37	[0.548, 5.48]
τ_{som}	Base turnover for soil organic matter	$\times 10^{-4}\ umol\ m^{-2}d^{-1}$	9.13	[0.274, 2.74]
K_{decomp}	Decomposition rate	$\times 10^{-3}\ d^{-1}$	1	[0.1, 10]
LMA	Leaf mass per area	$gC\ m^{-2}$	80	[20, 150]
$X_{stem_{init}}$	Initial value for stem C pool	$\times 10^3\ gC$	5	[1, 15]
$X_{root_{init}}$	Initial value for root C pool	gC	500	[100, 3000]
$X_{litter_{init}}$	Initial value for litter C pool	gC	600	[50, 1000]
$X_{som_{init}}$	Initial value for soil organic C pool	$\times 10^3\ gC$	7	[1, 25]

Table 3: A summary of eight parameters to be calibrated in surrogate-based ELM model. The default parameter value and prior parameter range are shown.

Parameter	Description	Unit	Default	Range
c_{root}	Rooting depth distribution parameter	m^{-1}	2.0	$[0.5, 4]$
SLA_{top}	Specific leaf area at canopy top	$m^2 gC^{-1}$	0.03	$[0.01, 0.05]$
N_{leaf}	Fraction of leaf N in RuBisCO	-	0.1007	$[0.1, 0.4]$
CN_{root}	Fine root C:N ratio	-	42	$[25, 60]$
A_{r2l}	Allocation ratio of fine root to leaf	-	1.0	$[0.3, 1.5]$
Res_m	Base rate for maintenance respiration	$\times 10^{-6} \mu mol\, m^{-2} s^{-1}$	2.525	$[1.5, 4]$
$t_{leaffall}$	Critical day length for senescence	$\times 10^4$ s	3.93	$[3.5, 4.5]$
GDD_{onset}	Accumulated growing degree days for leaf out	$^o C\, d$	800	$[600, 1000]$

Table 4: A summary of two parameters to be calibrated in the BiomE model. The default parameter value and prior parameter range are shown.

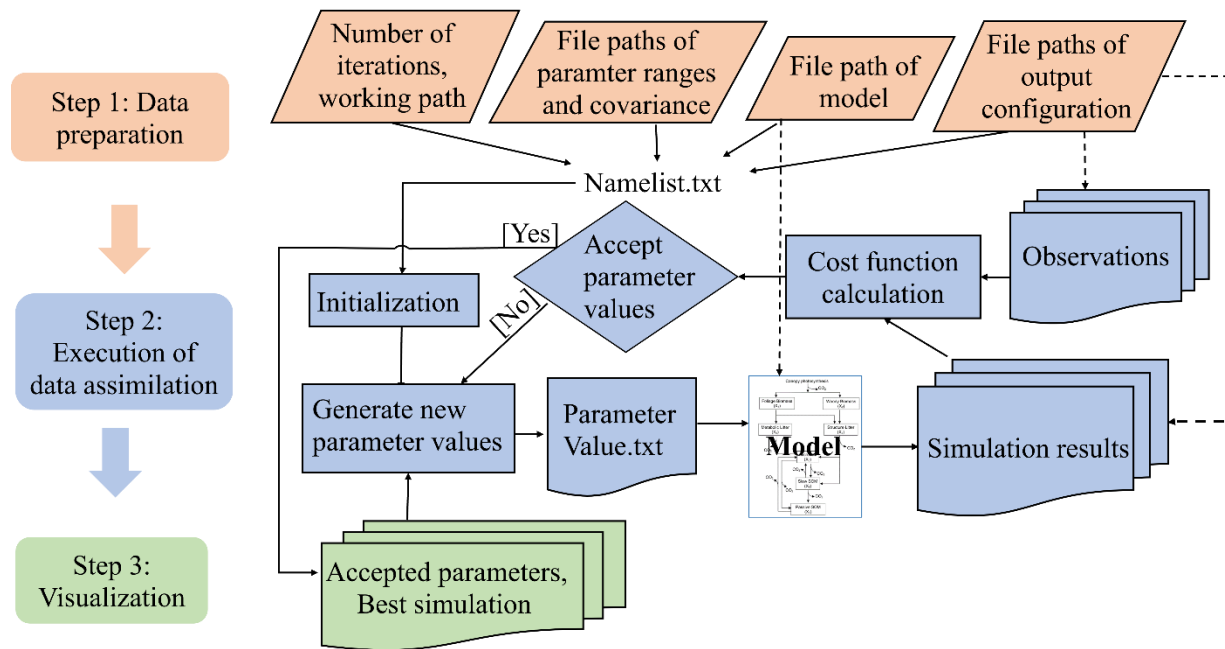| Parameter	Description	Unit	Default	Range
V_{annual}	Annual productivity per unit leaf area	$kgC\ y^{-1}m^2$	0.4	$[0.2, 2]$
M_{canopy}	Annual mortality rate in canopy layer	y^{-1}	0.02	$[0.01, 0.08]$

Figure captions

Figure 1: The three-step workflow of Model Independent Data Assimilation (MIDA) module. The workflow includes data preparation, execution of data assimilation (DA), and visualization. The data preparation step is to provide all the formatted essential data for DA via user input. The execution step is to calibrate parameter values towards a constrained posterior distribution with the fusion of observations. The visualization step is to diagnose the effects of DA. Rhombus in orange represents user-input data. Rectangle represents procedures and document/multidocument shape is for data files in computers. Dashed lines indicate locations of data. Solids lines indicate data flow pathways. With the three-step workflow, DA is agnostic to specific models and users will be released from technical burdens.

Figure 2: the GUI-MIDA window includes two panels. The upper panel is to set up a data assimilation task. Inputs can be loaded and applied to the step 1 on data preparation for DA. The lower panel is to run DA as described in step 2 and visualize the posterior distributions of parameters in step 3.

Figure 3: Comparison between the simulated daily net ecosystem exchange (NEE) by DALEC and the observed NEE at Harvard Forest from 1992 to 2006. Red circles represent modeled NEE with the optimized parameter values and green circles represent simulated NEE with the original parameter values. Simulations of DALEC are substantially improved after data assimilation in comparison with those before data assimilation.

Figure 4: Comparison between posterior distributions (red line) and default values (gray dash line) of the 21 parameters in DALEC. The peak in posterior distribution is the constrained parameter value from maximum likelihood estimation. This distinctive mode and its divergence from the default value indicates the effects of DA. Most parameters are well constrained, and some are far different from the original values.

Figure 5: Comparison between posterior distributions (red line) and default values (gray dash line) of the eight parameters in surrogate-based ELM. The peak in posterior distribution is the constrained parameter value from maximum likelihood estimation. This distinctive mode and its divergence from the default value indicates the effects of DA. Most parameters are well constrained, and some are far different from the original values.

Figure 6: Comparison between the simulated NEE, total leaf area index, latent heat flux by surrogate-based ELM and the observed ones at Missouri Ozark flux site from 2006 to 2014. The

blue lines indicate the observations, and their 95% confidence interval is in the dashed area. The green and red lines indicate the simulations with default parameter values and optimized values respectively. Simulations are generally improved after DA for all these three variables.

Figure 7: Comparison between the simulated growth date by 9 phenology models after DA and the observed growth date for *Larix laricina* with +9°C treatment at SPRUCE site from 2016 to 2018. Colored number indicates different models and shape represents different year. Overall, model 6,7,8,9 achieve better performance after DA.

Figure 8: Comparison between posterior distributions (red line) and default values (gray dash line) of the two parameters in BiomeE. The peak in posterior distribution is the constrained parameter value from maximum likelihood estimation. This distinctive mode and its divergence from the default value indicates the effects of DA. All parameters are well constrained and different from their original values.

Figure 9: Comparison between the simulated leaf area index (LAI) by BiomeE and the observed NEE at Willow Creek. Circles represent modeled NEE with the optimized parameter values and triangles represent simulated NEE with the original parameter values. Simulations of LAI are substantially improved after data assimilation in comparison with those before data assimilation.
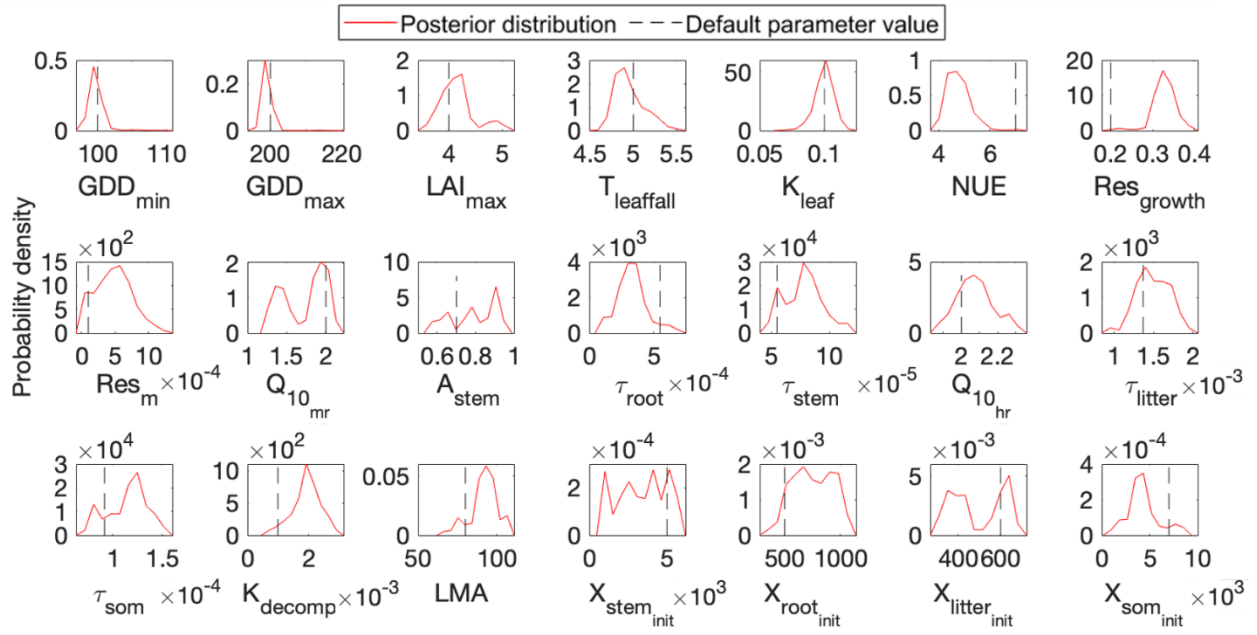
Figure 1: The three-step workflow of Model Independent Data Assimilation (MIDA) module. The workflow includes data preparation, execution of data assimilation (DA), and visualization. The data preparation step is to provide all the formatted essential data for DA via user input. The execution step is to calibrate parameter values towards a constrained posterior distribution with the fusion of observations. The visualization step is to diagnose the effects of DA. Rhombus in orange represents user-input data. Rectangle represents procedures and document/multidocument shape is for data files in computers. Dashed lines indicate locations of data. Solids lines indicate data flow pathways. With the three-step workflow, DA is agnostic to specific models and users will be released from technical burdens.

Figure 2: the GUI-MIDA window includes two panels. The upper panel is to set up a data assimilation task. Inputs can be loaded and applied to the step 1 on data preparation for DA. The lower panel is to run DA as described in step 2 and visualize the posterior distributions of parameters in step 3.

Figure 3: Comparison between the simulated daily net ecosystem exchange (NEE) by DALEC and the observed NEE at Harvard Forest from 1992 to 2006. Red circles represent modeled NEE with the optimized parameter values and green circles represent simulated NEE with the original parameter values. Simulations of DALEC are substantially improved after data assimilation in comparison with those before data assimilation.

Figure 4: Comparison between posterior distributions (red line) and default values (gray dash line) of the 21 parameters in DALEC. The peak in posterior distribution is the constrained parameter value from maximum likelihood estimation. This distinctive mode and its divergence from the default value indicates the effects of DA. Most parameters are well constrained, and some are far different from the original values.
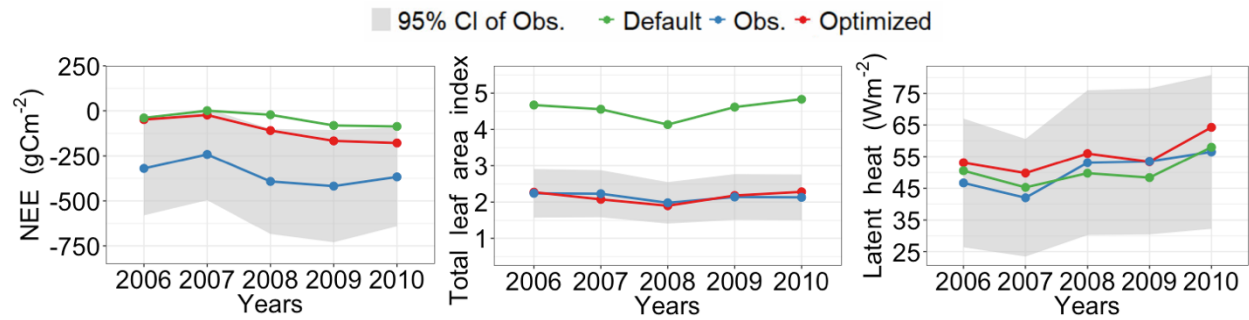
Figure 5: Comparison between posterior distributions (red line) and default values (gray dash line) of the eight parameters in surrogate-based ELM. The peak in posterior distribution is the constrained parameter value from maximum likelihood estimation. This distinctive mode and its divergence from the default value indicates the effects of DA. Most parameters are well constrained, and some are far different from the original values.

Figure 6: Comparison between the simulated NEE, total leaf area index, latent heat flux by surrogate-based ELM and the observed ones at Missouri Ozark flux site from 2006 to 2014. The blue lines indicate the observations, and their 95% confidence interval is in the dashed area. The green and red lines indicate the simulations with default parameter values and optimized values respectively. Simulations are generally improved after DA for all these three variables.
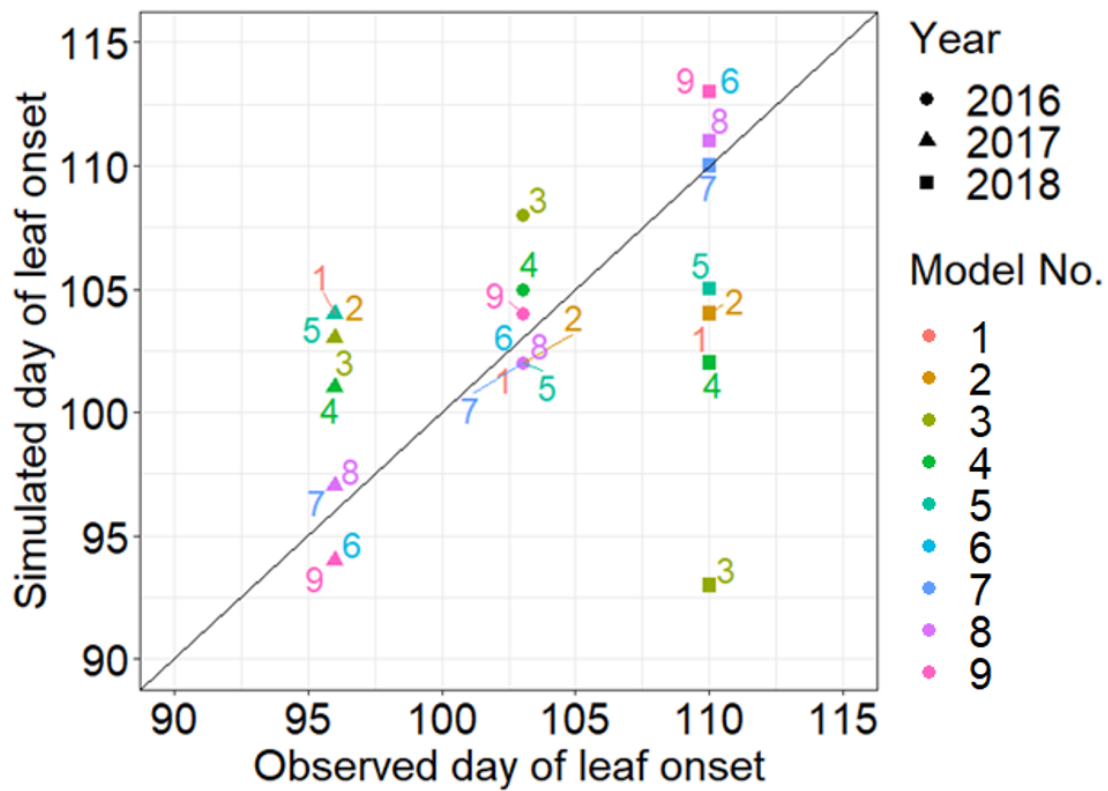
Figure 7: Comparison between the simulated growth date by 9 phenology models after DA and the observed growth date for *Larix laricina* with +9°C treatment at SPRUCE site from 2016 to 2018. Colored number indicates different models and shape represents different year. Overall, model 6,7,8,9 achieve better performance after DA.
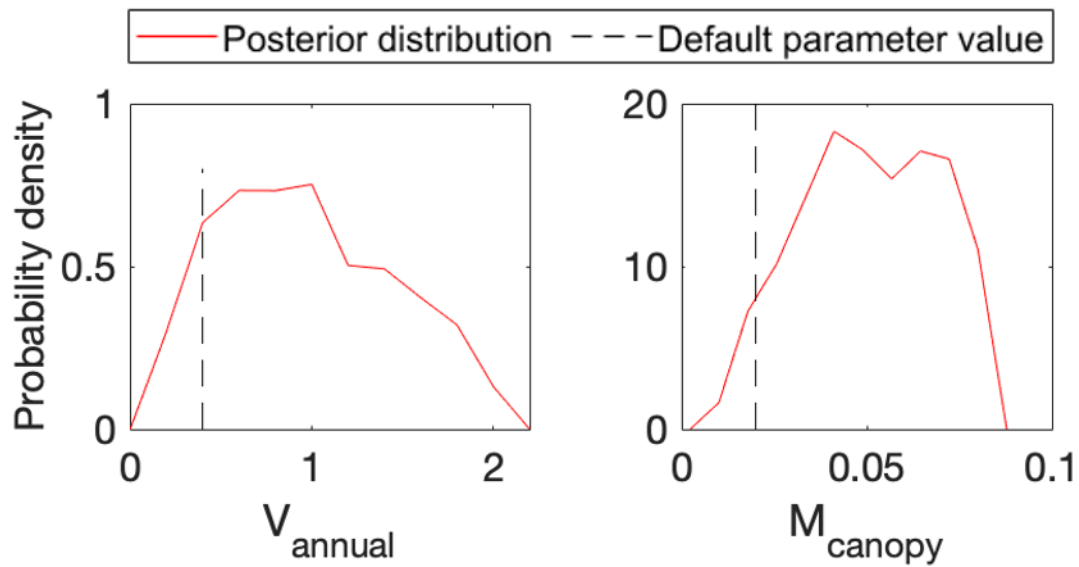
Figure 8: Comparison between posterior distributions (red line) and default values (gray dash line) of the two parameters in BiomeE. The peak in posterior distribution is the constrained parameter value from maximum likelihood estimation. This distinctive mode and its divergence from the default value indicates the effects of DA. All parameters are well constrained and different from their original values.
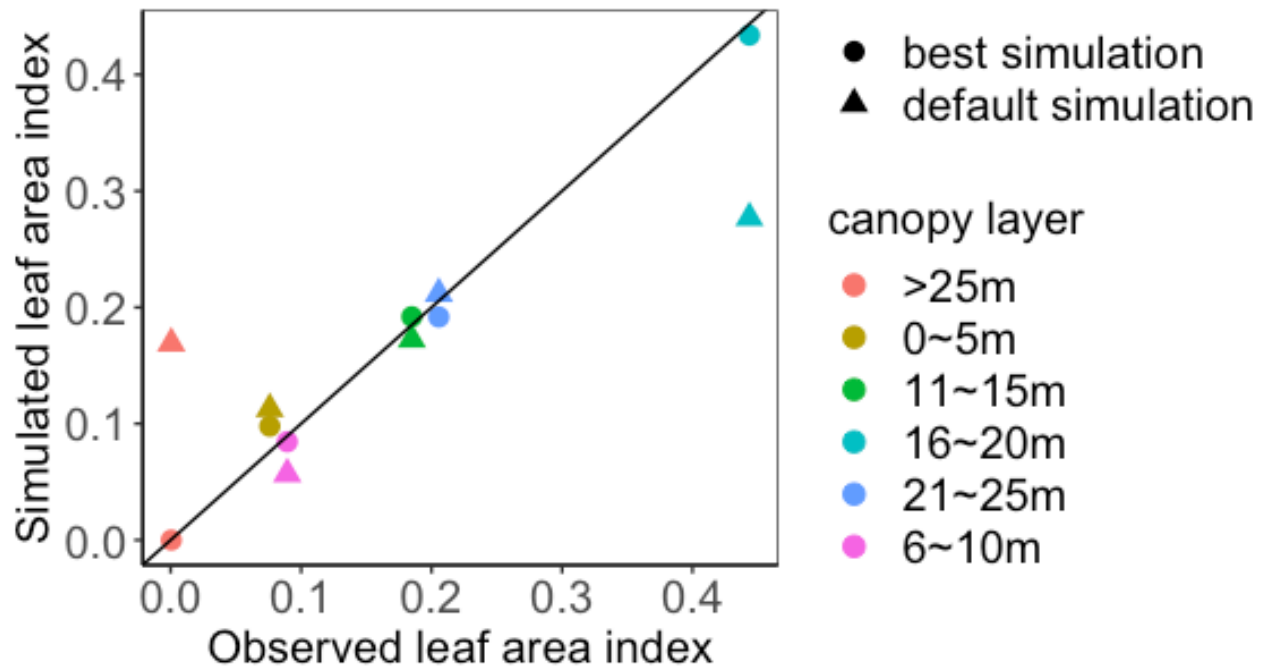
Figure 9: Comparison between the simulated leaf area index (LAI) by BiomeE and the observed NEE at Willow Creek. Circles represent modeled NEE with the optimized parameter values and triangles represent simulated NEE with the original parameter values. Simulations of LAI are substantially improved after data assimilation in comparison with those before data assimilation.