Hydrologic Model Calibration Overview

Dr. Bryan Tolson
http://www.civil.uwaterloo.ca/btolson/

CIVE 781: Principles of Hydrologic Modelling
University of Waterloo
Jun 3-8, 2019
Friday Outline

a. Introduction and overview of model calibration

b. Automatic or Auto Calibration:
   - Single-objective optimization
   - Multiple objective optimization

c. Sensitivity Analysis
My Research Expertise

1. How do we build better models?
   - New methods for calibration (optimization), sensitivity analysis, model-building and quantifying model prediction uncertainty
   - My focus is not on physics/chemistry but on procedures for fitting model predictions to measured data

2. How do we optimize the design, management or operation of complex water resources systems?
   - This involves one or more numerical simulation models

Consistently, my research is aimed at solving computationally intensive problems
Model Simulation Stages

- In general, when building/applying a model, if historical system response data is available, model simulations are conducted in three distinct steps:
  1. CALIBRATION
  2. VALIDATION
  3. FORECAST/FUTURE/SCENARIO PREDICTION

- Note that forecasting as a very specific definition:
  - See for example Beven and Young (2013)
  - Hydrologic forecasting is based on forecasted climate inputs to the model
  - Climate change or land use change impact assessment IS NOT forecasting!

Calibration versus Validation Model Testing

1. CALIBRATION: is the process of assigning parameters values that generate simulation results that are consistent with observed historical system response behaviour.

2. VALIDATION: Stop model refinement, an independent check that the model also reasonably predicts past historical data that were *not used in model calibration* (also hindcasting).

3. FORECAST/FUTURE/SCENARIO PREDICTION: use the model to evaluate question(s) of interest.

Classic /must read paper on model validation is by Klemes (1986):
- proposes 4 levels of increasingly difficult model validation, the most difficult being validating the model on a *new time period* and *new basin*.
- Urges modellers to choose the validation approach that most resembles the extrapolation level necessary to assess question of interest in Step 3 above.

Model Building versus Calibration

• Model calibration includes much more than simply assigning model parameter values
• Model calibration really includes decisions like:
  • How to discretize the model
  • How to interpolate point rainfall onto model computational units (e.g. HRUs)
  • With modelling frameworks like RAVEN, how to select the processes you will include in your model
• **Before you do any modelling, determine very carefully the reasons or questions you intend to answer with the model you build**
• Without these, it is very difficult to make objective decisions about building and calibrating your model
Subjective Decisions in Calibration

- Assume model structure, rainfall interpolation, discretization decisions are made already
- An incomplete list of additional subjective decisions you must make includes:
  - Model spin-up or initialization period (and initial state variable values)
  - Temporal period and locations for calibration
  - Measured system response variables to calibrate to
  - Model parameters to adjust
  - Initial values (sometimes) and feasible ranges of parameters selected to adjust
  - How model relative prediction quality be evaluated (this can include defining objective functions and constraints)
  - If uncertainty will be considered in calibration
  - How to generate candidate parameter sets

MANUAL CALIBRATION  AUTO-CALIBRATION
Manual vs Auto-Calibration

• How to generate candidate parameter sets

MANUAL CALIBRATION

AUTO-CALIBRATION = automated sampling

• This process can be very subjective
• Less subjective if conducted by expert modeler but ...
• Result quality can be a strong function of modeler mood/patience/curiousity [which is not good]
• Modeller must still decide steps/procedure re parameter value generation and how long she/he will spend iterating and trial and error

• Choice of algorithm to use can be subjective
• Choice of algorithm parameters is subjective for many algorithms
• How long to wait for answer is another sometimes subjective choice

• Both manual and automatic calibration can be “successful”
• Chances of manual model calibration success decrease with level of model experience
Manual vs Auto-Calibration

Many researchers and practitioners believe manual calibration is superior to automatic calibration. I DISAGREE. Reasons include:

1. Most humans are simply not capable of systematically searching multidimensional parameter space. Human answer not as optimal as algorithm.
2. Human cognition limits our ability to simultaneously assess the joint relative quality of one solution versus another considering multiple measured hydrographs. More data → the more we need to aggregate performance measures, hence defining an objective function.
3. If you want to quantify uncertainty, manual calibration approach is effectively incompatible for this purpose (a sampling algorithm is needed).
4. Do you really want to rely on your manual solution without checking if optimization could give you a significantly better quality prediction?
5. Do you really think you will only need to perform the tedious trial and error manual calibration process once?
6. There is no reason a practitioner can’t combine their expertise/judgement with an auto-calibration algorithm to for example choose between multiple candidate parameter sets that were generated via automatic calibration.
Model Calibration (Parameter Estimation) as an Optimization Problem

- Hydrologic simulation models have many parameters that can’t be measured in the field
  - Traditional approach: manual or trial and error
  - Alternatively it is easily formulated as an optimization problem

Optimization formulation:

\[
\begin{align*}
\text{min } f(x) & \quad \text{← model error} \\
\text{subject to } & \\
\quad x_{\text{min}} & \leq x \leq x_{\text{max}} \\
\text{possibly other constraints}
\end{align*}
\]
Bryan’s Subjective Guide to Subjective Decision-Making for Model Calibration

- Model spin-up or initialization period (and initial state variable values)
  - *One year is often safe but double check. Initial states can be calibrated also.*
- Temporal period and locations for calibration
  - *Easiest to have validation period follow calibration period. Where to put most extreme hydrologic responses – calibration or validation? Depends …*
- Measured system response data to calibrate to
  - *Use all data! Never ignore data just because a bit more coding needed to do so.*
- Model parameters to adjust
  - *Utilize modeler expertise but be wary of assuming too many truly unknown parameters do not have to be calibrated.*
- Initial values (sometimes) and feasible ranges of parameters selected to adjust
Bryan’s Subjective Guide to Subjective Decision-Making for Model Calibration

• How model relative prediction quality be evaluated (this can include defining objective functions and constraints)
  • Define this consistently, relate to model purpose.

• If uncertainty will be considered in calibration
  • Calibrate deterministically first. Moving to uncertainty quantification from deterministic calibration framework is fairly natural.

• How to generate candidate parameter sets
  • This process should be repeatable to avoid highly variable results
  • Do not select an obscure optimization technique, many recent comparative calibration studies ID good choices
Motivational Auto-Calibration results

• Can auto-calibration help an expert modeller?
  – Dr. R. Soulis from U Waterloo
  – Spatially distributed Watflow hydrologic model
  – ~40 parameters

Introduction

- Can auto-calibration help an expert modeller?
  - Dr. R. Soulis from U Waterloo
  - Spatially distributed Watflow hydrologic model
  - ~40 parameters

Expert calibration
NSE = 0.68

Guided calibration
NSE = 0.86
Motivational Auto-Calibration results

- AGENCY provided parameter set with gradient-based algorithm:

- After calibration with more effective algorithm:
Friday’s Course Objectives

• By the conclusion of this course, students will:
  – Be aware of some recommended practices for making subjective calibration decisions
  – Understand the calibration/model testing/forecasting/scenario analysis stages in model development
  – Be able to perform and understand sensitivity analysis, automatic calibration (e.g., DDS, PADDS) and you will be directed to future readings for quantifying model prediction uncertainty
  – Be trained to utilize an open source parameter estimation package called Ostrich to perform automatic calibration (optimization) with a variety of single and multi-objective optimization algorithms

• Participants will have a powerful suite of model calibration tools at their disposal at the end of this course
Automatic Calibration
(Optimizing model parameters)

Dr. Bryan Tolson
http://www.civil.uwaterloo.ca/btolson/

CIVE 781: Principles of Hydrologic Modelling
University of Waterloo
Jun 3-8, 2019
Lecture Outline

1. Introduction to model calibration
2. Why calibration is a hard problem (requiring optimization)
3. My own single-objective autocalibration algorithm: DDS
4. Ostrich – DDS – Software Exercise C1
5. Ostrich – PADD – Software Exercise C2
Model Calibration (Parameter Estimation) as an Optimization Problem

- Hydrologic simulation models have many parameters that can’t be measured in the field
  - Traditional approach: manual or trial and error
  - Alternatively it is easily formulated as an optimization problem

Optimization formulation:

\[
\begin{align*}
\text{min } f(x) \\
\text{subject to } x_{\text{min}} \leq x \leq x_{\text{max}} \\
\text{possibly other constraints}
\end{align*}
\]
Figure: Observed and modeled soil moisture of an irrigation experiment in two depths. SCE was used to infer soil physics parameters. (Arndt Piayda @ UFZ Leipzig)
Introduction of Calibration
– Example: Soil moisture data –

Figure: Observed and modeled soil moisture of an irrigation experiment in two depths. SCE was used to infer soil physics parameters. (Arndt Piayda @ UFZ Leipzig)
Introduction of Calibration
– Example: Tree population and evolution –

**Figure**: Observed and modeled tree populations (biomass) of three different species over 500 years. (Edna Rödig, Sebastian Lehmann @ UFZ Leipzig)
Introduction of Calibration

– Example: Tree population and evolution –

Figure: Observed and modeled tree populations (biomass) of three different species over 500 years. (Edna Rödig, Sebastian Lehmann @ UFZ Leipzig)
Figure: Calibrating WECan under permafrost conditions in Tianshuihai, China (Matthias Cuntz, Ute Wollschläger © UFZ, Leipzig)
Introduction of Calibration
– Example: Soil temperatures & moisture –

Figure: Calibrating WECan under permafrost conditions in Tianshuihai, China (Matthias Cuntz, Ute Wollschläger © UFZ, Leipzig)
Introduction of Calibration
– Example: Streamflow modeling –
Introduction of Calibration

– Example: Streamflow modeling –

Dense data
Few parameters
Integral data
Data error unknown
Outliers
Multiple stations
Introduction of Calibration

Minimize discrepancy between modeled variables and observations
Introduction of Calibration

Minimize discrepancy between modeled variables and observations

sounds easy, but:
Introduction of Calibration

Minimize discrepancy between modeled variables and observations

sounds easy, but:

- model (version) dependent
- question dependent
- location dependent
- budget dependent
- dependent on data availability
- ...
Introduction of Calibration

Minimize discrepancy between modeled variables and observations

sounds easy, but:

- model (version) dependent
- question dependent
- location dependent
- budget dependent
- dependent on data availability
- ...

There’s no recipe!
It’s an art!
Be creative!
Introduction of Calibration
– Main components: Objective function –

discrepancy measure = objective function
Introduction of Calibration
– Main components: Objective function –

discrepancy measure = objective function
Introduction of Calibration

– Main components: Objective function –

discrepancy measure = objective function
Introduction of Calibration
– Main components: Objective function –

discrepancy measure = objective function
Introduction of Calibration
– Main components: Objective function –

discrepancy measure = objective function

including:

- distance metric
  e.g., absolute error, squared difference, …

- combining multiple objectives
  e.g., bias, mean, variance, NSE, percent bias, …

- data error/ uncertainty handling
  e.g., likelihood, error model, …

- weighting of time steps/ multiple spatial units
  e.g., equal weighting, area weighting, seasonal weighting, …
Model Calibration Difficulty

- Difficult non-linear optimization problem:
  - typical problems shown to have multiple local minima. Meaning of multiple local minima?
  - On top of that, many of our hydrologic modelling codes are beasts in terms of computation time and so calibration budget is severely limited (perhaps 100 model runs total)

- Thus, very difficult and usually impossible to know if globally optimal solution is achieved. Global optimum parameter set?

- Traditional optimization algorithms not often suitable
- Hydrologic modellers have turned to heuristic global optimization algorithms
- They search for global optimum but no guarantee it will be found
“Easy” versus “Difficult” Optimization Problems

- Linear problems are easy to solve
- Some non-linear problems also fairly easy
- But many are difficult like this:

\[ f(x) \]

\[ x \]

\[ J(S) \]
Key Optimization Concepts

- Most problems have multiple local optima, many of them which are definitely inferior to global optima
- Derivatives, almost always computed numerically for our models, can be inaccurate and very wrong due to fixed precision in model input and output files
- In real calibration problems, we approximate the global optimum but never know how far from it we really are
- Some algorithms converge very slowly when presented with too many decision variables (parameters in a calibration problem), others not so much
- Some algorithms suffer greatly if parameter ranges are excessively wide, others not so much
- Most algorithms require fine tuning of algorithm parameters (how is unclear!) if modeler is in need of a good calibration result on a realistic timeline

- Some really poor choices for optimizing high dimensional (more than 10 DVs) problems:
  - Worst: Uniform random sampling or gridded sampling
  - You can do better: Start at parameter 1, solve 1-D optimization problem, got to parameter 2, repeat until all parameters cycled through

- Gradient-based searches or non-linear regression formulations
- Simulated Annealing and Genetic Algorithms
- Shuffled Complex Evolution (Duan et al., 1992)

- SCE is still a popular and often good choice but a newer alternative I unbiasedly recommend is called Dynamically Dimensioned Search (DDS)

- DDS was designed to overcome the slow convergence of SCE/GA plus eliminate the need to select optimization algorithm parameters
- A hydrologist does not want to mess around determining an appropriate ‘mutation probability’ or ‘population size’
Dynamically Dimensioned Search

- A novel model calibration tool
- Stochastic global optimization algorithm – a direct search method
- Originally designed for automatic calibration of environmental simulation models:
  - Simple to implement & no parameter-tuning needed
  - Generate calibrated model in modeller’s time frame
  - Calibration objective is to find good or acceptable solutions, not the globally optimal solution

---


Dynamically dimensioned search algorithm for computationally efficient watershed model calibration

Bryan A. Tolson¹ and Christine A. Shoemaker²
DDS Algorithm Description

- Algorithm scales to user-specified computational limits
- DDS mimics the manual calibration process

**STEP 1.** Define DDS inputs for $D$ dimensional problem:
- neighborhood perturbation size parameter, $r$ (0.2 is default)
- maximum # of function evaluations, $m$

**STEP 2.** Evaluate objective function at initial solution

**STEP 3.** Randomly select a subset of the $D$ decision variables for perturbation from the current best solution.

- Size of subset decreases as maximum function evaluation limit approached

**STEP 4.** Perturb the decision variables selected in Step 3 from their current best solution

- normally distributed perturbations with adequate variance ensures global search
- perturbations beyond decision variable boundary are reflected

**STEP 5.** Evaluate new solution and update current best solution if necessary

**STEP 6.** Update function evaluation counter, $i=i+1$, and check stopping criterion:
- IF $i = m$ → STOP
- ELSE repeat STEP 3
Robust Performance of DDS

• My research group has applied DDS to numerous case studies:
  – 6 – 60 parameters calibrated, 100 to 100,000 model evaluations
  – Excellent performance relative to other algorithms

• Yen et al. (2014):
  – Compared 6 calibration algorithms and concluded DDS “surpassed all other methods in convergence speed and behavioral statistics”

• Arsenault et al. (2014):
  – Compared 10 global optimization algorithms in 40 case studies
  – Concluded DDS was among top 3 algorithms which “were either as good as or better than the other methods”

• DDS applied above with **same algorithm parameter value**


DDS Performance Comparison: Simultaneous Flow, TSS & Total Phosphorus Calibration in SWAT2000

• 26 or 30 parameters depending on size of model
• Objective function, $E_w$, defined as follows:

$$\max_x E_w(x) = 0.5\left(E_{NS}^Q - \max[0, |%B^Q| -10]/100\right) +$$

$$0.2\left(E_{NS}^{TSS} - \max[0, |%B^{TSS}| -30]/100\right) +$$

$$0.3\left(E_{NS}^{TP} - \max[0, |%B^{TP}| -30]/100\right)$$

- $E_{NS}$ is Nash-Sutcliffe coefficient
- %B is average bias of model predictions
• Above is similar to a weighted $E_{NS}$
DDS Performance Comparison: “Fine-Tuned” SCE Parameters

Average Best Objective Function Value (~ Average NS coefficient)

Swat Model Evaluations

- DDS
- SCE (p=5)
- SCE (p=5 to p=2)
- SCE p=2
Comparing Final DDS & SCE Solutions on Rastrigin10 Function

Probability of finding equal or better solution

Objective Function Value (Min)

- DDS
- SCE p=1
- SCE p=2
- SCE p=3
- SCE p=4
Common Questions and Recommendations when Using DDS

• How many parameters should I calibrate?
  ... *lot’s if you need to (but good to do a synthetic test)*

• Should I do a sensitivity analysis?
  ... *not necessarily but often a good idea. Perhaps avoidable as long as there are not many parameters included that the obj. function is almost completely insensitive to*

• Should I try to adjust the value of DDS parameter to something other than \( r = 0.2 \)?
  ... *NO. Just perform multiple calibration trials and use best solution. When solving real problem, use at least 2 or 3 calibration trials in case unlucky*
Common Questions and Recommendations when Using DDS

• How many model evaluations (obj. function evaluations) should I use?
  ...This should be based on when you want an answer and avg. time to evaluate each parameter set
  ... 1000 has worked for calibrating dozens of parameters
  Or try 100*D where D is number of DVs

Observed DDS algorithm behaviour as user computational limit changes (not always like this though)
*TIP* when using DDS (PADDS) or any Algorithm *TIP*

- How many model evaluations (objective function evaluations) should I use?
  - There is a better approach than previous slide though (see below)

- How do I know my algorithm will work? Or is this convergence setting OK?
  - truthfully you can’t know until you try it
  - but a good indication given by solving a synthetic calibration experiment (optimal result known) for equivalent real calibration problem under same algorithm configuration
    - REALLY important for low budget calibration experiments
      - *try a synthetic calibration experiment*

Building synthetic experiment:
Synthetic & Real Calibration Experiments

• Consider a typical real calibration experiment:
  – Objective function is NSE
  – Reasonable to good NSE is 0.7 or 0.8
  – Unreasonable is probably less than 0.5
  – Terrible result is negative
  – Quality of initial guess of parameters? NSE close to 0 often.

• In contrast, what does a perfect synthetic calibration result look like?
  – NSE achievable?
Examples of Synthetic Calibration Experiments

• Hypothetical synthetic calibration test result 1:
  – Calibrate 10 select model parameters
  – Objective function is NSE
  – NSE for initial solutions are ~0.85 and final result of algorithm is 0.98
  – Is the algorithm working OK? Y/N?
  – Anything else going on?

• Hypothetical synthetic calibration test result 2:
  – Same as #1 except initial NSE solution is 0.1 and final optimized result showed optimized NSE = 0.60?
  – Is the algorithm working? Y/N?

• Actual synthetic test results included in recently submitted paper:
  – Calibrate distributed model to streamflow weighted average NSE of 12 stations
  – 149* model parameters estimated using 2000 model evaluations
  – Weighted average NSE = 0.98
Common Questions and Recommendations when Using DDS

• How can I handle constraints in my calibration? [beyond parameter bound constraints]

A) Assuming constraints are on some function of model output:
   IF constraint violated
      \[ SSE = \text{inf.} \]
   ELSE
      calculate SSE normally
   ENDIF

Place above in objective function evaluation code.
Works fine for DDS so long as at least ~10% of parameter space is feasible

B) Convert constraints to a second objective measuring the amount they are not collectively satisfied and solve a multi-objective optimization problem with PADDs

C) If constraints are on parameters, try hard to make sure DDS always generates a feasible parameter set by:
   - formulating decision variables to ensure this
   - using Ostrich software tied parameters option
The notes on analyzing results apply to DDS and other optimization algorithm outputs.

- You have just clicked run … waited … and now you have a single calibration result.
- Next step is to repeat it at least one more time from new initial solution (and new random seed if applicable).
- Your work is NOT done as a modeller. You should not blindly trust results. So once you have two or more algorithm results consider the following …
Analyzing Calibration Algorithm

- Main components: Analysis of calibration results –

don’t trust blindly but analyze the calibration results by:

✓ comparing results of independent calibration runs (multi-start)

∞ helps to detect if algorithm ended up in local optimum

---

![Graph of Griewank (N=10) Function Evaluations](image)

Objective Function Value

Function Evaluations

0  200  400  600  800  1000

0.35  0.3  0.25  0.2  0.15  0.1  0.05  0  0.3  0.2  0.1  0.05  0

---

![Graph of Griewank (N=10) Function Evaluations](image)

Objective Function Value

Function Evaluations

0  200  400  600  800  1000  1200  1400

0.4  0.3  0.2  0.1  0
Main components: Analysis of calibration results

don’t trust blindly but analyze the calibration results by:

✓ comparing results of independent calibration runs (multi-start)
   \(\Downarrow\) helps to detect if algorithm ended up in local optimum

✓ testing different initial guesses (first parameter set)
   \(\Downarrow\) helps to detect if algorithm ended up in local optimum
- **Main components: Analysis of calibration results** –

  don’t trust blindly but analyze the calibration results by:

  ✓ comparing results of independent calibration runs (multi-start)
    ◇ helps to detect if algorithm ended up in local optimum

  ✓ testing different initial guesses (first parameter set)
    ◇ helps to detect if algorithm ended up in local optimum

  ✓ analyzing how parameter values develop over course of calibration
    ◇ helps to identify insensitive/ unidentifiable parameters
    ◇ you might be able to identify parameter dependencies/ interactions

  ✓ analyzing final parameter values
    ◇ if values are at parameter boundary, revise the parameter search domain

  ✓ checking (visually) match of optimal model run and observations
    ◇ helps to see if objective function is suitable
READY to see how to use DDS?

A few options:

1. Easy to code based on WRR paper pseudocode
2. Download my easy to use MATLAB distribution package from my website: http://www.civil.uwaterloo.ca/btolson/
3. Ostrich software which we will utilize and is here: http://www.eng.buffalo.edu/~ismatott/Ostrich/OstrichMain.html
OSTRICH
(Optimization Software Toolkit)

A model-independent, multi-algorithm, optimization software tool

L. Shawn Matott (Primary developer)

Ostrich development supporters:
Dr. Bryan Tolson
Dr. Julianne Mai
Dr. James R. Craig
Dr. Amin Jahanpour
How to couple these two programs?

For lumped hydrologic models:
- most efficient to build application (.exe) with both codes integrated
- For DDS, the algorithm code is simple so this is actually easy

For even modestly computationally intensive models:
- Negligible efficiency gains to build application (.exe) with both codes integrated
- Any efficiency gains come at the cost of managing continuous integration of two different codes/programs

My philosophy is to keep codes separate, only loosely couple them

Ostrich parameter estimation software is perfect for this
What is OSTRICH?

- **OSTRICH** is a model-independent calibration and optimization tool that implements a variety of algorithms and calculates an extensive suite of calibration statistics.

- **OSTRICH** can be configured (with NO programming) to operate with any modeling program that utilizes a text-based input/output file format.

- Versions of **OSTRICH** are available for both Windows, Linux, Mac. Code is open source.

- **OSTRICH** is suitable for application to a variety of environmental and/or water resources problems.
General Steps to Prepare for a New Ostrich-based Calibration

1. Independent of Ostrich determine **how to run model** → input parameters & simulated outputs
2. Prepare model batch files so Ostrich can invoke model repeatedly. Must check this manually!
3. Prepare the template *****.tpl** Ostrich input file(s) for parameter value transfer
4. Prepare the **Ostln.txt** Ostrich input file to define all optimization run inputs
5. Run Ostrich & view key output files
Ostrich – Model Communication: **Parameter Transfer.** Concept of File Pairs

Ostrich template file, Irondequoit.rvp.tpl

model input file, Irondequoit.rvp

parameters are replaced with unique parameter names
(File already created for you in C1)
Preparing Ostrich Input files: **Ostln.Txt**

**Optimization Method**
- See Ostrich Manual

**The Model**
- File Pairs
- Parameters to Be Calibrated
- Save select model output files for best solution

**Ostln.txt - Notepad**

- **ProgramType**: DDS
- **ObjectiveFunction**: GCOP
- **ModelExecutable**: Ost-RAVEN.bat
- **PreserveBestModel**: save_best.bat
- **#OstrichWarmStart**: yes

```
BeginFilePairs
  Irondequoit.rvp.tpl; Irondequoit.rvp
  #can be multiple (.rvh, .rvl)
EndFilePairs

#Parameter/DV Specification
BeginParams
  #parameter init. low high tx_in tx_ost tx_out
  par_x1  random  0.01  2.5 none none none
  par_x2  random  -15  10  none none none
  par_x3  random   10  700 none none none
  par_x4  random    0   7   none none none
  par_x5  random     1  30   none none none
  par_x6  random     0   1   none none none
EndParams
```

- *Use uniformly randomly sampled parameter value as initial solution (possibly ignored due to later algorithm input)*

- *log10 here means Ostrich internally converts to log scale and thus algorithm operates on log x rather than x, but input and output of parameter in real/original scale*

- *‘none’ means no transformation*
Preparing Ostrich Input files: **OstIn.Txt** continued

**Optimization Method Specific Settings. Refer to Ostrich Manual.**

Where to find model output(s) to build objective function

Transform response variable(s) into cost function to be minimized

Minimize NegNS + APM → Minimize NegNS

Other constraints. See Ostrich Manual

**BeginResponseVars**
```
#name  filename
NS  \model\Irondequot_Diagnostics.csv; OST_NULL 1 4 ',';
```

**BeginTiedRespVars**
```
NegNS 1 NS wsum -1.00
```

**BeginCOP**
```
CostFunction NegNS
PenaltyFunction APM
```

**BeginConstraints**
```
# not needed when no constraints, but PenaltyFunction statement above is required
# name  type  penalty  lwr  upr  resp.var
```

# Randomized control added
RandomSeed 123

#Algorithm should be last in this file:

```
BeginDDSAlg
PerturbationValue 0.20
MaxIterations 50
UseRandomParamValues
UseInitialParamValues
# above initializes DDS to parameter values IN the initial model input files
EndDDSAlg
```
C1 is Raven calibration example. Details:

- Irondequoit creek watershed (326 km2) in New York State, USA draining into Lake Ontario
- Raven using simple lumped model (GR4J)
- Calibration objective: maximize Nash-Sutcliffe efficiency at outlet
- Six model parameters to optimize
- Solve this with DDS

Walkthrough and in class tasks can be found on your Exercise_C1.pdf file. Get started!
Optional notes on Exercise C1

- Run through only if time left
Exercise C1: Make faster by model change

- Put these at end of *.rvi file
- :SuppressOutput
- :SilentMode
- ...run it again. Faster?

- This is a slick RAVEN feature but something all models should do
Exercise C1: **extract** option to pull in initial parameters from model input files

- Pretend your model directory contains a good initial guess that was derived outside of Ostrich
- You want algorithm to start with this solution (but you don’t want to type out values in Ostin.txt)

- Copy parameter files (*.rvp here) to Ostrich level directory
- Edit Ostin.txt, Params section → **extract**
- Edit Ostin.txt, DDSAlg section → UseInitialParamValues
Exercise C1: \textbf{OstrichWarmStart yes} option to build from previous Ostrich result.

- Pretend you want to polish a past solution derived from Ostrich
- Keep same directory structure and input files as in previous run (OK – maybe move Ostrich output files)

- \textit{Edit Ostin.txt, top file section} \Rightarrow \textbf{OstrichWarmStart yes}
- \textit{Run it (repeating DDS)}
- \textit{Increase MaxIterations to 200, repeat, open Ostrich manual to page 40 and read about Fletcher-Reeves algorithm. Polish next DDS result with this local, gradient search}
- \textit{BE AWARE: Bug will not let extract/Warmstart options to work when you do transformations of parameters (e.g. log10)}
Exercise C1: OstrichWarmStart yes option to build from previous Ostrich result. POLISH with Fletcher-Reeves Algorithm

- Only need to change ProgramType (watch spelling!) and Algorithm section

- Tired of waiting for convergence? Mine required 3-5 minutes or 700 evaluations to converge. Do this to exit gracefully:
  - Create file OstQuit.txt in Ostrich directory
END Optional notes on Exercise C1
Multiple Objective (MO) Optimization Algorithms for Model Calibration

- **Fundamental MO concept is non-dominance in objective space**
- Step 1: use MO optimizer to fund multiple non-dominated (Pareto optimal) calibration solutions:
  - Step 2: modeller becomes decision-maker. Pick one of the above solutions as the calibration solution (multi-criteria decision making)
- Multi-objective calibration approach is fundamental to diagnose model problems and assess relative quality of multiple competing simulation models
Fundamental multi-objective concept is non-dominance in objective function space.
Fundamental multi-objective concept is non-dominance in objective function space.
Fundamental multi-objective concept is **non-dominance** in objective function space.
Fundamental multi-objective concept is **non-dominance** in objective function space

- A nondominated solution
- Also called a Pareto-optimal solution
Pareto Front / Nondominated set of solutions
Classic Approach to Multi-Objective Optimization

• Solve a series of single objective optimization problems, each of which is one point on the Pareto front

• Such an approach is hard to control and typically inefficient giving rise to heuristic, population based multi-objective optimization algorithms
Pareto Archived DDS (PADDS) Algorithm for Multiobjective Optimization


- Also:
Visualization of PADDS Algorithm

- Example water distribution network design problem
PA-DDS Algorithm

- Initialize starting solutions
- Create ND-solution set
- Perturb selected ND solution
- Select a ND solution
- Select the New solution
- New solution is ND?
- Update ND solutions (if necessary)
- Continue?
- STOP

Initialize starting solutions
Create ND-solution set
Perturb selected ND solution
Select a ND solution
Select the New solution
New solution is ND?
Update ND solutions (if necessary)
Continue?
STOP
PA-DDS Algorithm – Selection

Selection functions to identify which current nondominated solution is used as basis for perturbation to create the next candidate solution.

PADDS has 4 different options for selection

- Random
- Crowding Distance CD: Asadzadeh & Tolson (2009) *Conf Gen & Evol Comp*
- Hyper-volume contribution HVC: Asadzadeh & Tolson (2013) *Eng Opt*
- Convex-hull contribution CHC: Asadzadeh, Tolson & Burn (2014) *WRR*
PA-DDS Algorithm – Selection

Consider random selection process:

1. Select random Pareto solution
2. Perturb associated parameter set (candidate solution) and evaluate objectives
3. Check if non-dominated
4. Update Pareto Front (if necessary)

Note that if candidate solution is nondominated, then it is automatically selected for perturbation in the next iteration.
Example PA-DDS Results

- PA-DDS approximates the tradeoff between multiple objectives (any #)
- Built from DDS, it is also fast and effective and no parameter tuning
- Relative to other MO optimizers used in model calibration, it performs quite well:
• Below highlights why we do MO optimization
Exercise C2 – PADDS with Ostrich

C2 is Raven multi-objective calibration example

- **Walkthrough and in class tasks can be found on your Exercise_C2.pdf file.**
- Go through a few more slides and then you can get started!
Some interesting background on Exercise C2

- Full extent of semi-distributed model (subbasins and HRUs) on Canadian Shield
Some interesting background on Exercise C2

- C2 Exercise is subregion (RAVEN allows turning off I/O, simulation for groups of HRUs with a single line in *.rvi file)
- 21 subbasins (9 of which are explicit lakes), 103 HRUs, 6 yr simulation
- Calibrate to Vermillion streamflow and to Lac La Croix inflows
- Maximize NSE of both using PADDS and optimizing 128 parameters
Some interesting background on Exercise C2

- A better depiction of lake complexity below
RECALL: General Steps to Prepare for a New Ostrich-based Calibration

1. Independent of Ostrich determine how to run model → input parameters & simulated outputs
2. Prepare model batch files so Ostrich can invoke model repeatedly. Must check this manually!
3. Prepare the template ***.tpl Ostrich input file(s) for parameter value transfer
4. Prepare the OstIn.txt Ostrich input file to define all optimization run inputs
5. Run Ostrich & view key output files
Exercise C2: Preparing Ostrich Input files: Ostln.Txt

NEW Optimization Method

Not applicable for PADDs
Exercise C2: Preparing Ostrich Input files: OstIn.Txt continued

- If 3 or more objectives and >10000 model runs Exact may be slow
- Where to find model output(s) to build objective function. NOW two
- Transform response variable(s) into cost functions to be minimized
- Minimize two objective functions now

NOW GO TO Exercise C2 worksheet
Optional notes on Exercise C2

- Run through only if time left
Exercise C2: **OstrichWarmStart yes** option to build from previous Ostrich result.

- Pretend you want to polish a past Pareto front set of solutions derived from Ostrich
- Keep same directory structure and input files as in previous run (OK – maybe copy Ostrich output files)

- *Edit Ostin.txt, top file section → OstrichWarmStart yes*
- *Run it (repeating PADDS)*

- **BE AWARE:** Bug will not let extract/Warmstart options to work when you do transformations of parameters (e.g. log10)
Exercise C2: **InitialParams** group to evaluate a user defined set of initial solutions

- Pretend you have three solutions you want PADDS to be initialized with
- Keep same directory structure and input files as in previous run (OK – maybe move Ostrich output files)

- **Edit Ostin.txt**, below **Params group** ➔ enter:
  ```plaintext
  BeginInitialParams
  # put 2 or 3 solutions here, 1 solution per row
  EndInitialParams
  ```

- Run it (repeating PADDS)
- **E.g. 3 user supplied solutions**, you will see Ostrich evaluate these 3 PLUS finish initializing with 2 random solutions
Exercise C2: ProgramType ModelEvaluation

- The ProgramType **ModelEvaluation** allows users to provide Ostrich with a list of parameter sets to evaluate.
- Particularly useful to get all model outputs of interest at the end of a multi-objective optimization run (in MO mode, Ostrich can’t save only non-dominated solutions).
- Option to use with this: **PreserveModelOutput** yes.
- List parameter sets in **InitParams** group.
- Try it: grab solutions from OstNonDonSolutions0.txt, remove first 3 columns.
- Saves a bit too much but more elegant scripts allowable (see Ostrich manual).
END Optional notes on Exercise C2
Some results for longer runs w PADDs

Sample results for MO Problem in T2

- 20 evaluations
- 2000 Serial PADDs
- 20000 Serial PADDs
Additional Constraints in Calibration

• Adjust baseline spatially variable parameters (e.g., LAI) with one global adjustment such that they all increase or decrease according to a multiplier, \( m \)
  
  \[ \text{Par1a} = m \times \text{Par1a}_\text{base} \quad \text{etc.} \]

  – Again, use linear TiedParameter option in Ostrich
  – Can exceed allowable bounds in this case though!

• Adjust baseline spatially variable parameters (e.g., LAI) with one global adjustment such that they all slide in between min and max spatially variable bounds according to some fraction of the range, \( f \)
  
  \[ \text{Par1a} = \text{Par1a}_\text{min} + f \times (\text{Par1a}_\text{max} - \text{Par1a}_\text{min}) \quad \text{etc.} \]

  – Again, use linear TiedParameter option in Ostrich
  – Bounds respected now
Additional Constraints in Calibration


- Formulates calibration problem such that a maximum number of hydrological signatures (e.g., flow duration curve) are replicated by the model within some tolerance

- ~12 signatures considered
Sensitivity Analysis
Dr. Bryan Tolson
http://www.civil.uwaterloo.ca/btolson/

CIVE 781: Principles of Hydrologic Modelling
University of Waterloo
Jun 3-8, 2018
Background

- Define $f(x_1, x_2, ..., x_n)$ as an environmental simulation model (or model chain) that predicts some key output of interest, $y$:
  \[ y = f(x_1, x_2, ..., x_n) \]
  - $y$ might be peak simulated daily flow
  - where the $x_i$’s are uncertain model inputs (model parameters, spatial data, forcing data like rainfall)

- **Fundamental:** begin any sensitivity or uncertainty study by precisely defining $y$

- Note that uncertainty in a random variable $y$ can be described by:
  - Variance of $y$, Range of $y$, ???
Propagation and Analysis of Uncertainty

The effects of uncertain inputs on output can be examined in 3 different ways:

1. **Sensitivity Analysis** – methods for computing the effect of changes in inputs
2. **Uncertainty Propagation** – methods for calculating the uncertainty in model outputs due to model input uncertainty
3. **Uncertainty Analysis** – methods for comparing the importance of input uncertainties in terms of their relative contributions to output uncertainty
   - note that sensitivity analysis methods can also be considered as uncertainty analysis methods but some methods are better than others

• **Consider conceptual model having 2 uncertain inputs:**

\[ y = f(x_1, x_2) \]

*From Morgan and Henrion (1990)*
Sensitivity Analysis Methods

• Sensitivity analysis simply boils down to answering a “What if ... “ question

• From Morgan and Henrion (1990)

• Why do we conduct a sensitivity analysis?
  – *Always* answer this question before you start doing a sensitivity analysis
  – FIRST reason is to use results to *try* to determine the relative importance of input uncertainties
  – A related SECOND reason, is that the most important input uncertainties can be given the most care and attention to describe appropriately (e.g., what 2 of 4 model inputs should I go out and sample?)
  – THIRD reason is to simply learn about the model you are using ... it becomes less of a ‘black-box’ the more you understand it

• I would caution that it is very easy to conduct a meaningless sensitivity analysis (many people do it *only* because it is an accepted best practice)

• I would also caution that the distinction between important and unimportant model inputs is almost always fuzzy & subjective
Sensitivity Analysis Discussion

• Consider Exercise C2 → 128 model parameters
  – Do we really want to calibrate this many parameters?
• Who has conducted a sensitivity analysis?
• Go through some scenarios of understanding why it was done.
Sensitivity Analysis Methods

• Consider most common methods for measuring sensitivity. See Morgan and Henrion (1990) for complete details

• **Local sensitivity measures**
  – Measure is anchored to a nominal scenario (nominal values of uncertain inputs)
  – Requires only 1-2 model runs per uncertain input
  – Results do not apply across entire variable ranges if the model response surface is nonlinear

• **Global sensitivity measures**
  – Measure aggregate sensitivity across entire variable ranges
  – Generally require Monte Carlo type analysis (i.e., >>100 model runs)

• In ideal world, we typically want global sensitivity measures

\[ y = f(x_1, x_2) \] is our hydrologic model
Local Sensitivity Analysis Methods

Definitions:

- $\mathbf{x} = (x_1, x_2)$ [also $\mathbf{X} = (x_1, x_2)$] is vector of inputs for a 2 input model
- $\mathbf{x}^0 = (x_1^0, x_2^0)$ is inputs at their nominal or base values
- $y^0 = f(x_1^0, x_2^0)$ is output as a function, $f$, of nominal input values
e.g. we write $y = f(x_1, x_2)$ instead of writing $y = 2x_2^2 + x_1x_2 + x_2^3$
- $U(x, y)$ denotes a measure of how sensitive output $y$ is to input $x$ (or uncertainty importance), there are different $U$’s

Most fundamental sensitivity measure:

$$U_s(x_1, y) = \frac{\partial f}{\partial x_1} \bigg|_{x^0} \quad U_s(x_2, y) = \frac{\partial f}{\partial x_2} \bigg|_{x^0}$$

- Note units of $U_s(x_1, y)$ & $U_s(x_2, y)$ !!!
- Not helpful alone

From Morgan and Henrion (1990)
Numerical Derivatives

- In hydrologic modelling, we typically always approximate derivatives of any function $f$ numerically based on techniques below
- based on Taylor Series expansion, $\Delta x$ is a small change in $x_i$
- Forward Difference approximation
  - *Quick but less accurate*

\[
f'(x_i) \approx \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x}
\]

\[
\frac{\partial f}{\partial x_i} \bigg|_{x^0} \approx \frac{f(x^0, x_i^0 + \Delta x) - f(x^0)}{\Delta x}
\]

- Central Difference Approximation
  - *Slower but more accurate*

\[
f'(x_i) \approx \frac{f(x_i + \Delta x) - f(x_i - \Delta x)}{2\Delta x}
\]

\[
\frac{\partial f}{\partial x_i} \bigg|_{x^0} \approx \frac{f(x^0, x_i^0 + \Delta x) - f(x^0, x_i^0 - \Delta x)}{2\Delta x}
\]

- **Determination of appropriate $\Delta x$ is case study specific, not too large or not too small!**
Unitless local sensitivity measure:

\[ U_E(x_1, y) = \left. \frac{\partial f}{\partial x_1} \right|_{x_0} \frac{x_1^0}{y^0} \quad U_E(x_2, y) = \left. \frac{\partial f}{\partial x_2} \right|_{x_0} \frac{x_2^0}{y^0} \]

Nominal Range Sensitivity Method (swing weight sensitivity):

\[ U_R(x_1, y) = f(x_1^+, x_2^0) - f(x_1^-, x_2^0) \]
\[ U_R(x_2, y) = f(x_1^0, x_2^+) - f(x_1^0, x_2^-) \]

- \( x_2^+ \) & \( x_2^- \) are max & min values
- **Less** local method of sensitivity

From Morgan and Henrion (1990)
Depicting Sensitivities and Uncertainty Importance

- Uncertainty importance = uncertainty analysis = which is most important source of uncertainty
- Two common approaches are Tornado and Pareto charts
- Example with 3 input variables \( y = f(L, Q, z) \) with pictures below from Loucks et al. (2005)
- Series order in both chart types is based on magnitude of impacts

Note that a Pareto chart with y-axis as % of total uncertainty could also depict uncertainty importance results
Global Sensitivity Analysis

• Local expert: Dr. Julianne Mai
• Have a look at some of her research papers for some ideas about Global methods