REGSim tool documentation

Lakshmi E

Indian Institute of Technology Hyderabad

22-Jan-2020

Table of Contents

1. Installation of external python libraries:

1.1 Installing the Platypus package for non-dominated sorting genetic algorithm (NSGA-II):

- To install using pip, run the following command,

```
pip install platypus-opt
```

- To install the Platypus package using anaconda,

```
conda config --add channels conda-forge
conda install platypus-opt
```

- For more details about the Platypus package,

https://platypus.readthedocs.io/en/latest/getting-started.html#

1.2 Installing pyDOE module package for Latin Hypercube sampling (LHS) [4]:

- To install the package using pip command,

```
pip install --upgrade pyDOE
```

- To install using anaconda,

```
conda install -c conda-forge pydoe
```

- To download and install manually,

https://pythonhosted.org/pyDOE/index.html

Note: REGSim is under progressive development, and you can download the latest version at https://github.com/LaksE91/REGSim.git

2. Introduction

The tutorial gives an application of the Recharge Estimation and Groundwater Simulation (REGSim) tool to simulate the groundwater level using a simple conceptual model(Box-1). This toolbox helps to understand the groundwater behaviour at a regional scale to guide water management. The model works based on the water budget approach with inflow as recharge, lateral inflow, and outflow as pumping, lateral outflow, which influence the storage of the groundwater. We also included geographic information system(GIS) tools in REGSim to automate the process of lateral flow estimation based on the observed groundwater head.

The following section describes the process of the REGSim toolbox to run the framework in python platform. The first step is to estimate the lateral flow fluxes, which further used as input during the calibration period of the model (Section 3.1). The second step is about the simulation and optimization of the groundwater model. In the next level, validation of the model is performed based on the Pareto optimal solutions obtained during the calibration period (Section 3.2). The third part describes the uncertainty and sensitivity analysis used for the model (Section 3.3).

BOX-1:

The groundwater balance equation used in this framework is shown in equation (1) [2]

$$h_t = h_{t-1} + \frac{r * P_t}{S_y} - \frac{Q_{p_t}}{S_y * A} + \frac{Q_{in_t} - Q_{out_t}}{S_y * A} \qquad (1)$$

where, h is the depth to water level [m], r is the recharge factor [-], P is the rainfall [m], S_y is the specific yield [-], $Q_{in/out}$ is the lateral inflow/outflow [m^3/month], Q_p is the pumping rate [m^3], A is the aquifer area [m^2], and subscript t denotes the current month.

3. Implementation of REGSim tool with an example dataset

The REGSim tool aims to model the time series of regional groundwater levels using a lumped conceptual groundwater model. The working process and methods are illustrated in detail with an application to the aquifer system of the urban agglomerate Hyderabad, India. Here, REGSim tool is incorporated with an example dataset to simulate the groundwater level. The dataset for optimization and uncertainty analysis is supported by the comma-separated (**.csv**) file containing four inputs with monthly time steps includes rainfall, groundwater head, lateral inflow, and outflow.

The input file required for the tutorial is provided in `Data/` folder, and the expected results of the groundwater model are added in `Example_results/` folder.

The execution of the scripts is supported by the **IDLE/Spyder/command prompt.**

3.1 Estimation of lateral flow:

a. Code name: *Step-1a_Estimation_of_slope.py*

Description:

Evaluation of slope along the boundary facilitated using the ArcGIS tools, and the manuscript addresses detailed methodology. The input data and the specifications required for this module are given in Table 1. The **Create Points on Lines** tool for creating a point on the lines is downloaded from http://ianbroad.com/arcgis-toolbox-create-points-polylines-arcpy/.

Table 1: Data and its specification for the module.

Input data	File format	File name format	Remarks
Groundwater elevation	Raster (.tif)	'YEAR_GWL_MONTH.tif'	'2004_GWL_Jan.tif'
Study area boundary	Vector (.shp)	'bound_XXXX.shp', 'bndin_XXXX.shp', 'bndout_XXXX.shp',	XXXX – study area name Make it as three copies

Indian Institute of Technology Hyderabad

lakshhmi.17@gmail.com

- Set the current directory where the data and codes are stored in the folder. Given the user-defined buffer distance (meters) and the number of points, the average gradient along the study area boundary is calculated.

```
# work in the current directory
env.workspace=(input("give the current directory:"))
dirpath = os.getcwd()

#assign the buffer distance
buffer_dist = input('Buffer distance between the study area (meters):')
num_pts     = input('no. of points considered across the boundary:')
```

Functions:

`buffer(bound)`	User-defined function	To perform the buffer analysis.
`ext_pts(bound,boundin, boundout,bufin,bufout)`		To create the point feature across the boundary.
`pts_value(raster,list)`		To extract groundwater elevation in point feature.
`avg_sl(raster)`		To estimate the average slope across the boundary.

Arguments:

`bound, boundin, boundout`	Three sets of same study area boundary files.
`bufin, bufout`	Buffer inside and outside polygon generated from the buffer tool.
`raster`	Groundwater elevation raster.
`list`	Point vector for buffer inside and outside.

Output:

```
give the current directory:'F:\CE15RESCH11013_LAKSHMI\Code\GWM\Code_process_instruct
\Step_1_Lateralflowestimation'
Buffer distance between the study area (meters):1000
no. of points considered across the boundary:1000
Creating buffer inside and outside the boundary area...
Converting polygon to line feature class...
Created points to the feature class...
bound_hmda.shp
bndin_hmda.shp
bndou_hmda.shp
buffin1000.shp
bufout1000.shp
Extracting the elevation data from the raster to the point featureclass...
2004_GWL_Jan.tif
buffin1000.shp
bndin_hmda.shp
bufout1000.shp
bndou_hmda.shp
Estimating slope in each point of the boundary area...
['bndin_Jan_extrpts.dbf', 'bndou_Jan_extrpts.dbf']

        Table:          bound1000_Jan_extrpts1000_04.dbf
        Type:           dBase III Plus
        Codepage:       ascii (plain ol' ascii)
        Status:         DbfStatus.CLOSED
        Last updated:   2020-04-09
        Record count:   1000
        Field count:    8
        Record length:  100
        --Fields--
          0) mem_point_ N(10,0)
          1) mem_point1 F(13,11)
          2) bound_hmda N(9,0)
          3) bound_hm_1 N(10,0)
          4) bound_hm_2 N(6,0)
          5) bound_hm_3 F(13,11)
          6) rastervalu F(19,11)
          7) slope F(19,11)

 Saving the output file
```


Indian Institute of Technology Hyderabad

lakshhmi.17@gmail.com

b. Code name: *Step1b_Estimation_of_laterflow.py*

Description:

Lateral flow fluxes are estimated based on Darcy's law (Box-2). Input data required for the script is the **'.csv.'** file (*Note: Rearrange the file name month-wise, see the, e.g., figure, slope.csv*), which contains the file names of output (**'output.csv'**) from the previous step. Here, lateral flow divided into lateral inflow (flow enters into the study area boundary) and lateral outflow (flow leave the study area boundary).

```
slope - Notepad
File  Edit  Format  View  Help
bound1000_Jan_extrpts1000_04
bound1000_Feb_extrpts1000_04
bound1000_Mar_extrpts1000_04
bound1000_Apr_extrpts1000_04
bound1000_May_extrpts1000_04
bound1000_Jun_extrpts1000_04
bound1000_Jul_extrpts1000_04
bound1000_Aug_extrpts1000_04
bound1000_Nov_extrpts1000_04
bound1000_Dec_extrpts1000_04
```

Run the script and set the current directory where the data and codes are available. The average slope is generated as output with given user-defined input.

BOX-2:

The lateral flow can be estimated using Darcy's law as follows(2):

$$Q_{in/out_t} = T * i * L \qquad (2)$$

Where, $Q_{in/out}$ is the lateral flow (m^3 /month), i is the hydraulic gradient (m/m), T is the transmissivity (m^2/month), l is the length of the study area boundary (m).

Output:

```
give the current directory:'F:\CE15RESCH11013_LAKSHMI\Code\GWM\Code_process_inst
ruct\Step_1_Lateralflowestimation'

iterating using zip
Transmissivity of the aquifer:(unit m2/day)144
Polyline study area boundary shapefile:'bound_hmda_line.shp'

iterating using zip
[2718909.7653732379]  [2420892.2911623488]

 Lateral inflow and outflow are estimated
```

The sample dataset to execute the lateral flow scripts (section 3.1 a, b) includes groundwater elevation raster (January 2004), and boundary shapefiles. User can automate the python script with the given monthly groundwater elevation raster and boundary shapefiles.

3.2 Calibration and validation of the model:

a. Code name: *Step_2_Calibration_of_the_model.py*

Description:

Non dominated sorting genetic algorithm II (NSGA-II) [3], multi-objective optimization method used during the calibration of the model. The calibrated parameters such as specific yield, recharge factor, and maximum pumping rate and objective function as Root Mean Squared Error, Mean Absolute Error, and Nash-Sutcliffe model efficiency are considered during the optimization process. The data required to calibrate the model are monthly groundwater head [meters], precipitation [millimetres], and lateral flow [million cubic meter].

- We simulate the model under three recharge scenarios, such as constant recharge for all the months (Case 1), two recharge factors for monsoon and non-monsoon seasons (Case 2), and three recharge factors for winter, summer and monsoon seasons (Case 3).
- In the given an example, the total number of months considered is 60 and the calibrated period as 48. Run the model with required recharge conditions.
- Set the parameters to range based on the characteristic of the aquifer considered for the analysis. The number of decision variables varies based on the test case is considered. E.g., Case 1 has three decision variables, such as pumping rate, specific yield, and recharge factor, and three objective functions as default for all the cases. User can give their required iterations during the simulation.

```
total number of months considered:60
number of months considered for calibration:48
Test case: 1
maximum pumping range [min,max]:[50*10**6,100*10**6]
specific yield range [min,max]:[0.014,0.16]
recharge case-1 range [min,max]:[0.05,0.2]
area of the study area boundary in m2: 5536407425
no of decision variables: 3
no of objective functions: 3
no. of iterations:10000
```

- The model executed with the specified parameters, and the performance metrics are determined by fitting the observed and simulated groundwater head. Using the NSGA-II algorithm, the groundwater model is calibrated and computes the optimal pareto front. The best optimal solutions are selected based on user decisions. The optimal pareto solutions obtained during the simulation are stored as **'pareto.txt.'**
- The user can edit or add the objective functions in the script **'metrics.py'** to obtain the pareto optimal front.

```python
# Calculate root mean squared error
def rmse_metric(obs,sim):
    rmse     = np.sqrt((np.mean((obs - sim)**2)))
    return rmse

# Calculate mean absolute error
def mae_metric(obs,sim):
    mae = np.mean(np.abs((obs - sim)))
    return mae

# Calculate nash sutcliff efficiency
def nse_metric(obs,sim):
    nse      = 1 - sum((sim-obs)**2)/sum((obs-np.mean(obs))**2)
    return nse
```

Functions:

<table>
<tr><td><code>data_sep(input_data)</code></td><td rowspan="3">User-defined function</td><td>It divide the data into training and testing period based on the input data.</td></tr>
<tr><td><code>gw_model(input_para)</code></td><td>To solve the problem using the NSGA-II algorithm.</td></tr>
<tr><td><code>sim_mod(input_para,input_calib,area,rech_case)</code></td><td>It invokes the groundwater model for the optimization and returns the multi-objective functions.</td></tr>
<tr><td><code>Problem(V,M),</code>
<code>problem.types,</code>
<code>problem.function</code></td><td>Per-defined class of Platypus module</td><td>To define the functions, list of decision variables and returns the objective values.</td></tr>
<tr><td><code>paretoplot(df_opt,rech_case)</code></td><td>User-defined function</td><td>It is used for graphical representation of the pareto set.</td></tr>
</table>

Arguments:

`input_data`	Dataset of the model
`input_para`	Decision variables
`rech_case`	Recharge scenarios (t = 1,2,3)

`input_calib`	Dataset during the calibration period
`V, M`	V = the total number of decision variables for each case. M= the total number of objective functions considered. Here M= 3 problem.types = it assigns the decision variables problem.function = defines the function (here, `gw_model()`) that call the model with a list of decision variable and gives the list of the objective values.
`df_opt`	Dataframe contains pareto optimal solutions.

Output:

The function `paretoplot()` is invoked to perform the specific task, and the output is generated using the python code **'visualplot.py'**.

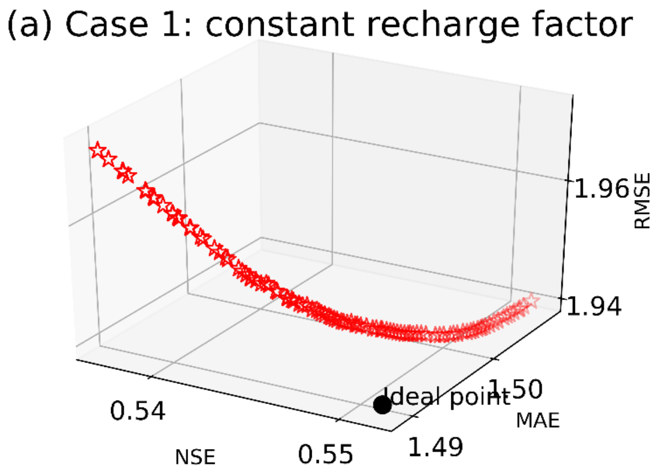
```
def paretoplot(df,test_case):

    fig = plt.figure()
    ax  = fig.add_subplot(111, projection='3d')

    # ideal point
    x = max(-df.NSE)
    y = min(df.MAE)
    z = min(df.RMSE)

    # plot the pareto-front of the objective functions
    ax.scatter(-df.NSE,df.MAE,df.RMSE,s=100,edgecolors='r',
               marker = '*',facecolors ='none')
    ax.scatter(x,y,z, s=100,edgecolors='k',marker = 'o',
               facecolors ='black')
    ax.text(x,y,z, 'Ideal point',size=14)
    ax.set_title("(a) Case-{}".format(test_case), fontsize=20)
    ax.set_xticklabels(-df.NSE,fontsize=15)
    ax.set_yticklabels(df.MAE,fontsize=15)
    ax.set_zticklabels(df.RMSE,fontsize=15)
    ax.zaxis.set_major_formatter(FormatStrFormatter('%.2f'))
    ax.xaxis.set_major_formatter(FormatStrFormatter('%.2f'))
    ax.yaxis.set_major_formatter(FormatStrFormatter('%.2f'))
    ax.set_xlabel("NSE", fontsize =12,labelpad=10)
    ax.set_zlabel("RMSE", fontsize =12,labelpad=5,rotation=90)
    ax.set_ylabel("MAE", fontsize =12,labelpad=10)
    ax.locator_params(tight=True, nbins=3)

    plt.tight_layout()
    plt.savefig(".\ Results\paretoplot_case{}.png".format(test_case),
                dpi=600,bbox_inches='tight')
    plt.show()
```



b. Code name: *<u>Step_3_Validation_of_the_model.py</u>*

Description:

The optimal solutions obtained from the pareto front is further used to validate the model for three recharge scenarios. For the given an example, the optimal value of calibrated parameters chosen for the Case 1 recharge scenario.

```
area of the study area boundary in m2: 5536407425
Test case: 1
Optimal specific yield:0.019
Optimal pumping rate: 9.92078e+07
Optimal recharge rate: 0.19
```


Indian Institute of Technology Hyderabad
lakshhmi.17@gmail.com

Functions:

`modelrun(optimal_set, data,area,rech_case)`	User-defined function	It calls the groundwater model for the simulation.
`valplot(obsv_head,gwhead,tcount,months,rech_case)`		It plots the simulated and observed head during the validation period.

Arguments:

`optimal_set`	List of optimal solutions for three cases.
`rech_case`	Recharge scenarios (t = 1,2,3).
`data`	Data frame consists of the total dataset used during the calibration and validation period.
`area`	Area in meter^2.
`obsv_head`	Monthly observed groundwater head.
`gwhead`	Simulated groundwater head.
`tcount`	The time duration of the model.
`months`	Variable to label the month/year in the plot.

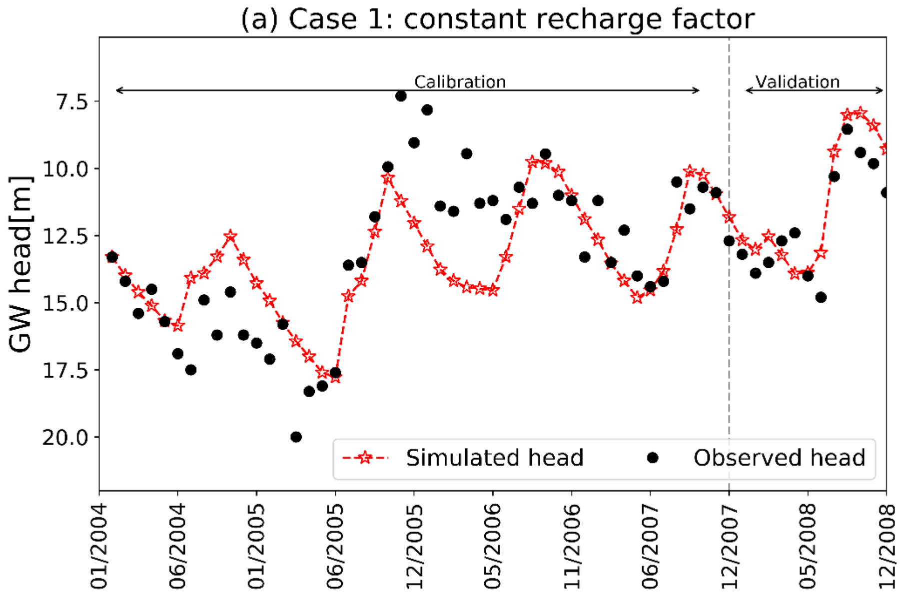
Output:

The given example plot is generated based on the `matplotlib` module used in the code. The user can modify the code x-axis range based on the time and month of the graph in **'Step 3 Validation_of_the_model.py'** and also the other specification such as annotations, text properties (***visualplot.py***) concerning the requirement.

```
obsv_head                   = input_data.H
months                      = ['01/2004','06/2004','01/2005','06/2005','12/2005',
                               '05/2006','11/2006','06/2007','12/2007','05/2008',
                               '12/2008']

tcount                      = np.arange(1,61,1)

val_plt                     = valplot(obsv_head,gwhead,tcount,months,rech_case)
```



3.3 Uncertainty and sensitivity analysis:

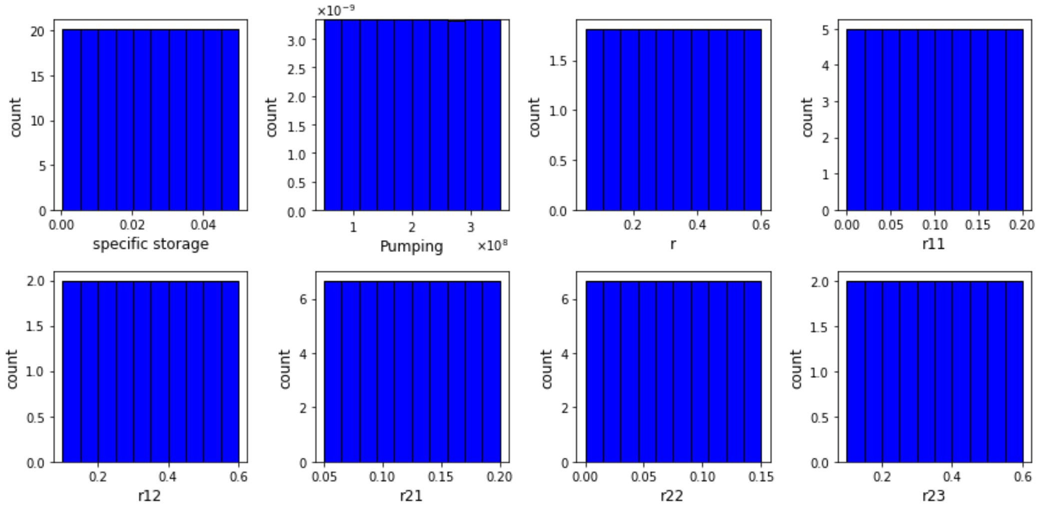
a. Code name: *Step_4_Uncertainity_check.py*

Description:

- Generalized likelihood uncertainty estimation (GLUE) proposed by [1] is employed to predict the uncertainty in the groundwater model (Please refer to the author's paper for the detailed methodology).To assess the uncertainty, the model assigns a plausible range of each parameter. Here, random parameter samples obtained using the Latin hypercube sampling method (LHS).
- Run the script and give the parameter range for all three cases to generate the random sample sets, as shown in the figure below. The histogram shows the LHS sampling for all the parameter sets.

```
total number of months considered:60
number of months considered for calibration:48
maximum pumping range [min,max]:[50*10**6,350*10**6]
specific yield range [min,max]:[0.0005,0.05]
recharge case-1 range [min,max]:[0.1,0.6]
recharge non-monsoon case-2 [min,max]:[0,0.2]
recharge monsoon case-2 [min,max]:[0.08,0.6]
recharge winter case-3 [min,max]:[0.05,0.2]
recharge summer case-3 [min,max]:[0,0.15]
recharge monsoon case-3 [min,max]:[0.08,0.6]
area of the study area boundary in m2: 5536407425
Test case: 1
no of sample:1000
```


lakshhmi.17@gmail.com



- Assign the confidence interval limit to predict the uncertainty interval. In the example, 90% confidence interval used, where 5% is the lower limit, and 95% is the upper limit. Also, assign the percentage of the acceptable threshold (behavioural set), say 5% here. Assume the maximum depth to water table is feasible in your study area. Here, we considered 25m as maximum depth to water table, to avoid negative values during the simulation process.

```
lower Confidence interval:0.05
upper Confidence interval:0.95
assign percentage of acceptable threshold:0.05
possible maximum depth to water table(m):25
Confidence_interval considered:90
```

Functions:

`rand(noi,nos,p,sy,r11,r12,r21,r22,r23)`		To call the LHS module (external library), to generate a uniform sample of the parameter set.
`uncertain(rech_case,input_data,input_calib,samp_set,area,lb,ub,cutoff,h_max,1)`	User-defined function	It invokes the GLUE method to estimate predictive uncertainty.
`myglueplot(CI_bounds,rech_case)`		To plot the prediction intervals obtained from GLUE to capture the observed head.

`obsv_inside(CI_bounds)`		Percentage of observation value captured within the uncertainty interval.

Arguments:

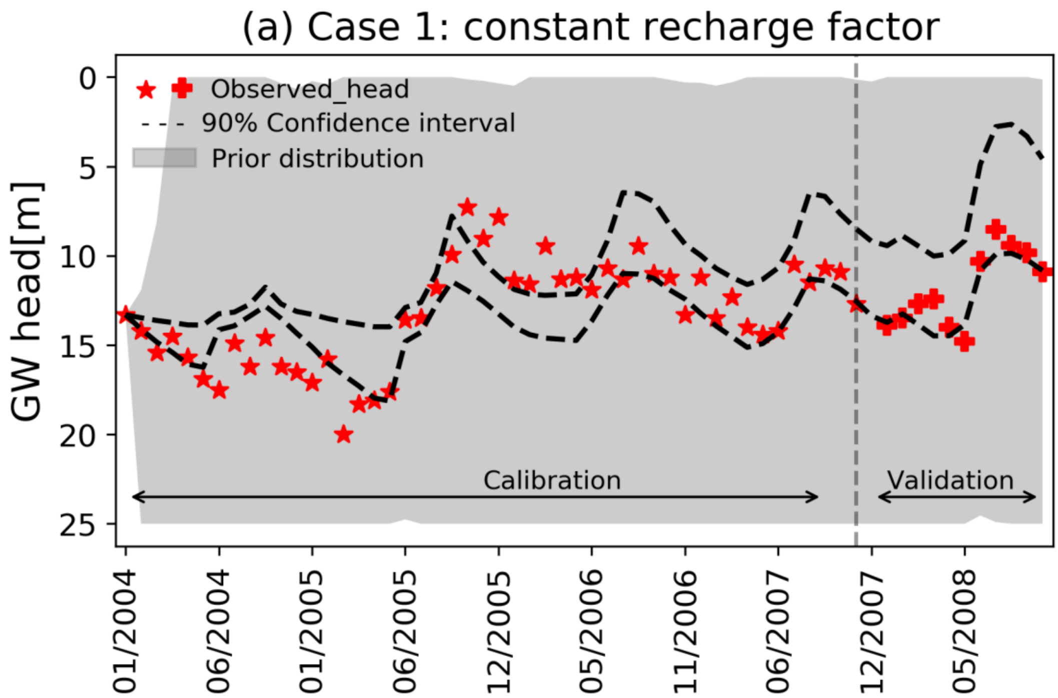
`noi`	Number of parameters used to generate random sample sets.
`nos`	Number of sample sets,
`P`	Maximum pumping range
`r1`	Recharge factor for case-1
`r11`	Recharge factor for non-monsoon, case-2
`r12`	Recharge factor for monsoon, case-2
`r21`	Recharge factor for winter, case-3
`r22`	Recharge factor for summer, case-3
`r23`	Recharge factor for monsoon, case-3
`rech_case`	Recharge scenarios (t = 1,2,3).
`input_data`	Total datasets used in the model
`input_calib`	Dataset for the calibration period
`samp_set`	Random sample parameter set
`area`	Area of the study area
`lb`	The lower limit of the confidence interval
`ub`	The upper limit of the confidence interval
`cut_off`	Acceptable threshold, behavioural set
`h_max`	Maximum depth to water table within the study area (meter)
`CI_bounds`	An input data frame of uncertainty prediction

Indian Institute of Technology
Hyderabad

lakshhmi.17@gmail.com

Output:

In the example, we run the model for the case-1 scenarios with a 90% prediction interval, as shown in the figure. The grey portion interval is the total range of the parameter set considered. In contrast, the black dotted line is the 90% confidence interval (User can modify the plotting code, ***glueplot.py*** based on their requirement).




```
percentage of observation within Confidence interval:50.0%
```

b. Code name: *Step_4b_Senstivity.py*

Description:

- Empirical cumulative distribution (ECDF) function is used to plot the distribution of the datasets to identify the sensitivity of the input parameters. The ranges of the input parameters are based on the wide range values considered during the uncertainty analysis using the LHS method.
- ECDF curve for each input parameter is plotted based on the user-defined input variables such as the recharge cases, confidence interval, and behavioural and non-behavioural. The behavioural set is the acceptable threshold of the performance metrics (say, top 5% of NSE), whereas the non-behavioural set is the remaining dataset of the performance metric (say, 1- 0.05 = 0.95).

```
total number of months considered: 60
number of months considered for calibration:48
Test case: 1
lower Confidence interval:0.05
upper Confidence interval:0.95
assign percentage of acceptable threshold:0.05
assign percentage of unacceptable threshold:0.95
possible maximum depth to water table(m):25
area of the study area boundary in m2: 5536407425
```

Functions:

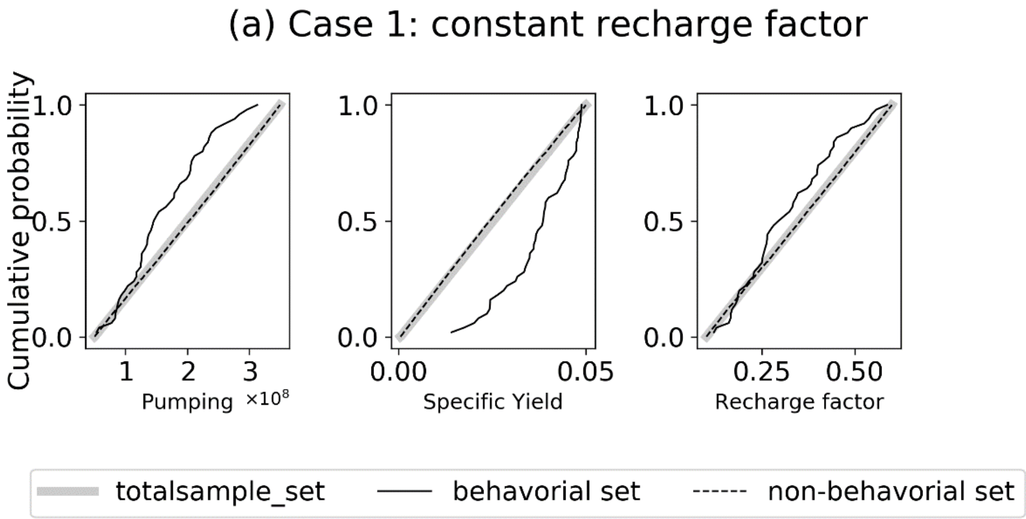
`Sim_glue(rech_case,data,calib,df_Psets,area,lb,ub,cutoff1,h_max,1)`	User-defined function	It invokes the GLUE method to return the acceptable parameter set to estimate the cumulative distribution function (CDF).
`eplt(evar_p,evar_q,rech_case)`		The function, plot the CDF of parameter sets.

Arguments:

`rech_case`	Recharge scenarios (t = 1,2,3).
`data`	Total datasets used in the model
`cal`	Dataset for the calibration period
`df_Psets`	Random sample parameter set
`area`	Area of the study area
`lb`	The lower limit of the confidence interval
`ub`	The upper limit of the confidence interval
`cut_off1`	Acceptable threshold, behavioural set
`cut_off2`	Unacceptable threshold, non-behavioural set
`h_max`	Maximum depth to water table within the study area (meter)
`evar_p`	Cumulative probability (0-1)
`evar_q`	A sample set of each input parameter

Indian Institute of Technology
Hyderabad
lakshhmi.17@gmail.com

Output:



4. Norms of the aquifer properties:

The aquifer properties, such as transmissivity, specific yield, and recharge, can be used for the groundwater assessment based on the report of the groundwater resource estimation committee (GEC). The following tables are the recommended values of the aquifer properties and utilized in the area where there is a lack of sufficient data and information available in the field (Source: http://cgwb.gov.in/documents/gec97.pdf).

4.1 Transmissivity:

Type of Aquifer	Transmissivity range (m^2/day)
POROUS ROCK FORMATIONS	
• Unconsolidated formations	250 to 4000
• Semi-consolidated formations	100 to 2300
HARD ROCK FORMATIONS	
• Igneous and metamorphic rocks excluding volcanic and carbonate rocks	10 to 500
• Volcanic rocks	25 to 100

4.2 Specific yield:

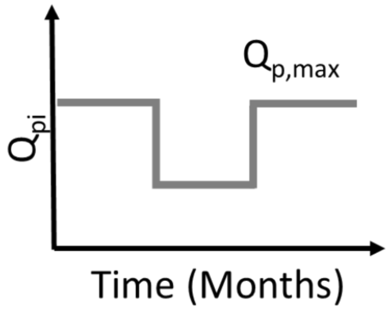
S.No	Formation	Recommended value (%)	Minimum value (%)	Maximum value(%)
1.	***Alluvial areas***			
	Sandy	16	12	20
	Silty	10	8	12
	Clayey	6	4	8

2.	***Hard rock areas***			
	Weathered granites, gneiss, schist with low clay content	3	2	4
	Weathered granites, gneiss, schist with significant clay content	1.5	1	2
	Weathered or vesicular, jointed basalt	2	1	3
	Laterite	2.5	2	3
	Sandstone	3	1	5
	Quartzite	1.5	1	2
	Limestone	2	1	3
	Karstified limestone	8	5	15
	Phyllites, shales	1.5	1	2
	Massive poorly fractured rock	0.3	0.2	0.5

4.3 Recharge due to rainfall

S.No	Formation	Recommended value (%)	Minimum value (%)	Maximum value(%)
1.	***Alluvial areas***			
	Indo-Gangetic and inland areas	22	20	25
	East coast	16	14	18
	West coast	10	8	12
2.	***Hard rock areas***			
	Weathered granites, gneiss, schist with low clay content	11	10	12
	Weathered granites, gneiss, schist with significant clay content	8	5	9
	Granulite facies like charnockite etc.	5	4	6
	Vesicular and jointed basalt	13	12	14
	Weathered basalt	7	6	8
	Laterite	7	6	8
	Semi-consolidated sandstone	12	10	14
	Consolidated sandstone, quartzite, limestone (expect cavernous limestone)	6	5	7
	Phyllites, shales	4	3	5
	Massive poorly fractured rock	1	1	3

4.4 Step function of pumping discharge:



$Q_{p_i} \rightarrow f(Q_{p,max})$ [m^3/s], where $Q_{p,max}$ is the maximum pumping rate and f is a step function . Pumping rate as a function of time(months). Pumping is assumed to be high during the non-monsoon season and lesser in monsoon season.

References:

[1] Beven, K., & Binley, A. (1992). The future of distributed models: model calibration and uncertainty prediction. *Hydrological processes*, *6*(3), 279-298, https://doi.org/10.1002/hyp.3360060305.

[2] Bredenkamp, D. B., Botha, L. J., Van Tonder, G. J., & Van Rensburg, H. J. (1995). Manual on quantitative estimation of groundwater recharge and aquifer storativity: based on practical hydro-logical methods. Water Research Commission.

[3] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., & Fast, A. (2002). Nsga-ii. *IEEE transactions on evolutionary computation*, *6*(2), 182-197, https://doi.org/10.1109/4235.996017 .

[4] Lee, A. D. (2018). pyDOE: Design of experiments for Python.

Indian Institute of Technology
Hyderabad

lakshhmi.17@gmail.com