The 3-Dimensional Probabilistic Landslide Susceptibility (3DPLS) Model:

The 3-Dimensional Probabilistic Landslide Susceptibility Model (3DPLS) is a Python code developed for landslide susceptibility assessment. The 3DPLS model evaluates the landslide susceptibility on a local to a regional scale (i.e. single slope to 10 km2) and allows for the effects of variability of the model parameter on slope stability to be accounted for.
The 3DPLS model couples the hydrological and slope stability models. The hydrological model calculates the transient pore pressure changes due to rainfall infiltration using Iverson's linearized solution of the Richards equation assuming tension saturation. The slope stability model calculates the factor of safety (FS) by utilizing the extension of Bishop's simplified method of slope stability analysis to three dimensions. The 3DPLS model requires topographic data (e.g., DEM, slope, aspect, groundwater depth, depth to bedrock, geological zones), hydrological parameters (e.g., steady background infiltration rate, permeability coefficient, diffusivity), geotechnical parameters (e.g., soil unit weight, cohesion, friction angle), and rainfall data.
The model has a grid containing hundreds or thousands of cells depending on the problem size and refinement. The smallest unit of the grid is called a grid cell having its own model parameters. The developed model calculates the FS of an ellipsoidal sliding surface consisting of grid cells over a discretized problem domain while equivalent cell-based models, as the name states, perform the calculations per each cell individually. The model generates a large number of ellipsoidal sliding surfaces centered at each grid cell over the terrain and calculates the FS of all sliding surfaces. After the calculation, each cell is involved in several ellipsoidal sliding surfaces with different FS values. Among all FS values, the minimum FS representing the critical ellipsoidal sliding surface is assigned to each cell. Each simulation results in a FS map over the terrain. After a number simulation, the 3DPLS model provides the FS map of each simulation, the mean FS, μ(FS), and the probability of failure, Pf, of each cell.

The current version of the 3DPLS model can consider only 1 soil type over the study area. Besides, all the cells should have data (NoData value is not considered). The changes can be made in the main code, but the function code should be kept as it is.

For your questions and cooperation,

Emir Ahmet OGUZ
Norwegian University of Science and Technology
Department of Civil and Environmental Engineering
N-7491 Trondheim, Norway
e-mail: emir.a.oguz@ntnu.com /emirahmetoguz@gmail.com

Input parameters for the example problems:

There are 3 validation problems to validate the slope stability model of the 3DPLS. In addition to validation problems, 2 example problems (simplified case problem and case study problem given in Oguz et al. 2021) are provided in the “InputData” folder.

For validation problems and simplified case problems, some modifications are implemented in addition to the 3DPLS model. The code will do the modifications automatically when "ProblemName" is defined as one of {'Pr1', 'Pr2','Pr3S1Dry','Pr3S2Dry','Pr3S2Wet' 'SimpCase'}. These modifications are explained below for each example problem.

The following sections show the required changes in the main code for each example individually. When the changes are done, the code can be run. There is no need to make any change in the function code. The definitions of parameters exist in the main code.

The details of the problems are provided in Oguz et al. (2021).

Note: Line numbers are according to the main code and the provided code are set for the validation problem – 1.

*Validation Problem – 1:

Below parameters are to be changed for the validation problem – 1:

Line #	Parameter	Value
35	GIS_Data_Directory	Main_Directory + '\\InputData\\Validation\\Problem1'
118	riInp	np.array(([0.],[86400]))
136	CorrLenRangeX	np.array((0,))
137	CorrLenRangeY	np.array((0,))
143	InZone	np.array(([int(nrows/2),int(nrows/2)],[int(ncols/2),int(ncols/2)]))
146	MCnumber	1
149	AnalysisType	'Drained'
150	FSCalType	'Bishop3D'
174	UwsInp	1
175	MuC	0.1
176	MuPhi	0
182	CoVRange	np.array(('Low',))
183	CoVRangeC	np.array((0,))
184	CoVRangePhi	np.array((0.,))
197	Ella	1
199	Ellb	1
201	Ellc	1
203	EllAlpha	90
205	Ellz	0.5
215	SubDisNum	0
230	ProblemName	'Pr1'

Note: In addition to the 3DPLS model:

- Pore water pressures are assigned as zero to simulate dry condition when ProblemName is defined as ‘Pr1’.

Input files with different refinements (cell size) can be found for validation problem-1 in "InputData" folder.

***Validation Problem – 2:**

Below parameters are to be changed for the validation problem – 2:

Line #	Parameter	Value
35	GIS_Data_Directory	Main_Directory + '\\InputData\\Validation\\Problem2'
118	riInp	np.array(([0.],[86400]))
136	CorrLenRangeX	np.array((0,))
137	CorrLenRangeY	np.array((0,))
143	InZone	np.array(([95,95],[149,149]))
1	MCnumber	1
149	AnalysisType	'Drained'
150	FSCalType	'Bishop3D'
174	UwsInp	1
175	MuC	0.116
176	MuPhi	15
182	CoVRange	np.array(('Low',))
183	CoVRangeC	np.array((0,))
184	CoVRangePhi	np.array((0.,))
197	Ella	2.02
199	Ellb	2.02
201	Ellc	2.02
203	EllAlpha	90
205	Ellz	1.556
215	SubDisNum	0
229	ProblemName	'Pr2'

Note: In addition to the 3DPLS model:

- Pore water pressures are assigned as zero to simulate dry condition when ProblemName is defined as ‘Pr2’.
- In the 3DPLS model, the center of the ellipsoidal sliding surface is introduced with an offset perpendicular to the ground surface. Therefore, it was not possible to introduce the center of the sliding surface. With a small modification to the proposed model, the center could be defined.

<u>*Validation Problem – 3:</u>

Below parameters are to be changed for the validation problem – 3:

Line #	Parameter	Value
35	GIS_Data_Directory	Main_Directory + '\\InputData\\Validation\\Problem3\\Slide1' or Main_Directory + '\\InputData\\Validation\\Problem3\\Slide2'
118	riInp	np.array(([0.],[86400]))
136	CorrLenRangeX	np.array((0,))
137	CorrLenRangeY	np.array((0,))
143	InZone	np.array(([int(25.5/cellsize), int(25.5/cellsize)],[int(ncols/2-1),int(ncols/2-1)]))
146	MCnumber	1
149	AnalysisType	'Drained'
150	FSCalType	'Bishop3D'
174	UwsInp	18.84
175	MuC	28.7
176	MuPhi	20
182	CoVRange	np.array(('Low',))
183	CoVRangeC	np.array((0,))
184	CoVRangePhi	np.array((0.,))
197	Ella	24.38
199	Ellb	24.38
201	Ellc	24.38
203	EllAlpha	90
205	Ellz	16.35
215	SubDisNum	0
229	ProblemName	'Pr3S1Dry' or 'Pr3S2Dry' or 'Pr3S2Wet'

Note: In addition to the 3DPLS model:
For Slide 1 – Dry:

- Pore water pressures are assigned as zero to simulate dry condition when ProblemName is defined as ‘'Pr3S1Dry'.
- The inclination of ellipsoidal surface, EllBeta, is assigned directly as the arctan(0.5) instead of calculating for a rectangular area.

For Slide 2 – Dry:

- Pore water pressures are assigned as zero to simulate dry condition when ProblemName is defined as ‘'Pr3S2Dry'.
- The inclination of ellipsoidal surface, EllBeta, is assigned directly as the arctan(0.5) instead of calculating for a rectangular area.
- The parameters, {"ThetaAvr", "Theta", "AngleTangentXZE1", "A"} are modified when the cells are truncated. For truncated cells, new cohesion and friction angle values are assigned.

For Slide 2 – Wet:

- Negative pore water pressures are assigned as zero and suction is ignored when ProblemName is defined as ‘'Pr3S2Wet'.
- The inclination of ellipsoidal surface, EllBeta, is assigned directly as the arctan(0.5) instead of calculating for a rectangular area.
- The parameters, {"ThetaAvr", "Theta", "AngleTangentXZE1", "A"} are modified when the cells are truncated. For truncated cells, new cohesion and friction angle values are assigned.

*Simplified case problem:

Below parameters are to be changed for the simplified case problem:

Line #	Parameter	Value
35	GIS_Data_Directory	Main_Directory + '\\InputData\\SimplifiedCase'
118	riInp	np.array(([0.],[86400]))
136	CorrLenRangeX	np.array((0, 10, 20, 50, 100, 200, 500, 1000)
137	CorrLenRangeY	np.array((0, 10, 20, 50, 100, 200, 500, 1000)
143	InZone	np.array(([10,29],[10,29]))
146	MCnumber	1000
149	AnalysisType	'Drained' or 'Undrained'
150	FSCalType	'Bishop3D'
156	RanFieldMethod	'SCMD' or 'CMD'
158	SaveMat	'YES'
163	UwsInp	20
164	MuSu	40
168	CoVRange	np.array(('Low', 'Moderate', 'High'))
169	CoVRangeSu	np.array((0.1, 0.2, 0.3))
174	UwsInp	20
175	MuC	6
176	MuPhi	40
177	Ksat	0.000001
178	Diff0	0.000005
182	CoVRange	np.array(('Low', 'Moderate', 'High'))
183	CoVRangeC	np.array((0.1, 0.2, 0.3))
184	CoVRangePhi	np.array((0.05, 0.10, 0.15))
197	Ella	20
199	Ellb	20
201	Ellc	2
203	EllAlpha	90
205	Ellz	0
215	SubDisNum	0
229	ProblemName	'SimpCase'

Note: In addition to the 3DPLS model:

- During the sub-discretization, the DEM data is recalculated instead of using "np.kron" when ProblemName is defined as ‘'SimpCase'.

***Case study: Kvam Landslides:**

In Oguz et al. (2021): "Effects of soil heterogeneity on the susceptibility of shallow landslides", the code has been implemented on a case study: Kvam Landslides. The results can be reproduced using the provided code.
Below parameters are to be changed for the case study: Kvam Landslides:

Line #	Parameter	Value
35	GIS_Data_Directory	Main_Directory + '\\InputData\\KvamCaseStudy'
118	riInp	np.array(([7.144e-7],[86400]))
136	CorrLenRangeX	np.array((50,))
137	CorrLenRangeY	np.array((50,))
143	InZone	np.array(([10,107],[20,77]))
146	MCnumber	1000
149	AnalysisType	'Drained'
150	FSCalType	'Bishop3D'
156	RanFieldMethod	'SCMD'
158	SaveMat	'YES'
174	UwsInp	20
175	MuC	4
176	MuPhi	32
177	Ksat	0.000001
178	Diff0	0.000005
182	CoVRange	np.array(('Kvam',))
183	CoVRangeC	np.array((0.3,))
184	CoVRangePhi	np.array((0.2,))
197	Ella	100
199	Ellb	20
201	Ellc	2.5
203	EllAlpha	0
205	Ellz	0
211	TimeToAnalyse	86400 or 0
215	SubDisNum	100
229	ProblemName	' '

Note: The lines, {112,113,114,115} should be activated.