



IC Modelling and Analysis in Geotechnical Engineering

Universal Model Interface for PLAXIS (UMIP)

Contents:

The Universal Model Interface for PLAXIS (UMIP) is a platform designed to facilitate the implementation of user-defined soil models (UDSM) into PLAXIS. It includes an interface with the information passed by PLAXIS' kernel to the UDSM and general-purpose algorithms based on the Modified Euler sub-stepping approach with automatic error control. Additionally, it provides a repository for subroutines and functions that perform standard operations shared among different models. It is continuously expanding to deal with the wide range of models under development within the IC MAGE group.

Disclaimer:

IC MAGE soil models undergo extensive testing and validation but can never be guaranteed to be free of errors. IC MAGE soil models are employed at the users own risk and the authors of these models cannot be held responsible or liable for any errors arising from their application.

Document history:

Date	Revision	
2021-04-16	1.2	Initial publication
2021-05-11	1.3	Non-local integration
2021-05-14	1.4	Special option
2021-05-29	2.0	Time-dependent behaviour
2021-07-23	2.1	Stiffness matrix scaling control
2021-09-17	2.2	Additional non-local options
2022-01-11	2.3	Additional energy options
2022-04-28	3.0	Random fields for 2D
2022-06-05	3.1	Cross-correlated random fields for 2D
2022-09-30	3.2	Stability updates & introduction of two yield surface plasticity (ialg = 2)
2023-01-24	3.3	Factors of safety & improved support for settings file
2023-06-01	3.4	Integration of hypoplastic models (ialg = 3), support for HCA models
2023-08-08	3.5	Support for K_0 profiles and direct stiffness matrix specification
2024-10-04	3.6	Extended FoS support, initiated documentation project

How to cite:

Taborda, DMG, Kontoe, S, Tsiampousi, A (2024) "IC MAGE UMIP – universal model interface for PLAXIS (Version 3.6)". Zenodo.

doi: [10.5281/zenodo.13890543](https://doi.org/10.5281/zenodo.13890543)

IC MAGE UMIP

Contents of this document

Structure..... 3

UMIP – UDSM Interface Notes 3

 Initialisation variable 3

 Model characteristic flags 4

 Model integration flags 4

Non-Local Implementation..... 5

Safety Analysis 6

Integration algorithms..... 6

Cyclic reversal detection algorithm 6

Modifying model behaviour using ‘special option’ 7

Energy Module 7

Random Fields 10

References 12

STRUCTURE

The UMIP currently employs a Modified-Euler sub-stepping scheme with automatic error control based on the work outlined in Sloan (1987), Potts & Zdravkovic (1999) and Sloan et al. (2001). It uses the Pegasus method originally proposed by Dowell & Jarratt (1972) following the approach in Hong et al. (2012). The yield surface drift correction is based on Potts & Gens (1985), Potts & Zdravkovic (1999) and Sloan et al. (2001). The sequence of operations follows that described in detail by Azeiteiro (2021).

Energy-based quantities according to the algorithm proposed by Taborda et al. (2016) are also calculated, enabling the estimation of energy dissipation during shearing, as well as an evaluation of the corresponding damping ratio.

The UMIP is further divided into eight distinct components that aggregate different aspects of the code:

<i>Data modules</i>	Contain data required for running the UMIP
<i>Interface</i>	Link between PLAXIS and the UMIP
<i>Integration</i>	Link between the UMIP and the constitutive laws
<i>Tools</i>	Mathematical and computational tools for constitutive modelling
<i>Error</i>	Centralises error and warning codes
<i>Output</i>	Produces output of the integration of the model
<i>Energy</i>	Calculates energy-based quantities if requested
<i>Licensing</i>	Verifies the licensing rights of the current copy

The contents of each component in terms of functions are listed in the following section.

UMIP – UDSM INTERFACE NOTES**Initialisation variable**

Value of InitVar: hard(1)	Description
0	Full initialisation Provided by Plaxis kernel at start of first phase of model, or when “reset hardening parameters” is selected UMIP calls hardin if idtask = 1
1	Initialised state
2	Use of special option Calls hardres at every start of idtask = 2
3	Correction of stress state Calls hardin at every start of idtask = 2
4	HCA model flag Calls single instance of hardening parameter initialisation via linear_creep subroutine. Hard(1) reverts to 3 after HCA hardening parameter initialisation.

Model characteristic flags

Flag	Description
<i>istension</i>	Tension/Compression flag 0: compressive formulation 1: tensile formulation
<i>isaniso</i>	Flag for elastic anisotropy 0: isotropic elastic matrix 1: cross-anisotropic elasticity
<i>adir</i>	Flag defining direction of anisotropy 1: XX 2: YY 3: ZZ
<i>icyclic</i>	Reversal detection for cyclic models 0: No reversal detection via elauload/plauload 1: Reversal detection formulated in strain-space (elauload.f90) 2: Reversal detection formulated in stress space (plauload.f90)
<i>isafe</i>	Flag for safety analysis 0: not available 1: available
<i>ialg</i>	Integration algorithm
<i>inonlocal</i>	Nonlocal algorithm
<i>icreep</i>	Creep (HCA)
<i>irecord</i>	Recording (HCA)

Model integration flags

Flag	Description
<i>inemin</i>	Minimum number of elastic integration steps
<i>inemax</i>	Maximum number of elastic integration steps
<i>ihrdec</i>	Flag for integration control based on elastic hardening parameters
<i>ihrdpc</i>	Flag for integration control based on plastic hardening parameters
<i>nyield</i>	Maximum number of iterations of the Pegasus method
<i>inpmin</i>	Minimum number of plastic integration steps
<i>inpmax</i>	Maximum number of plastic integration steps
<i>ndrmax</i>	Maximum number of drift correction steps
<i>xnetol</i>	Elastic integration tolerance
<i>xnptol</i>	Plastic integration tolerance
<i>smltol</i>	Small tolerance
<i>ytol</i>	Yield surface tolerance

<i>tiptol</i>	Apex tolerance for correction of stress state
<i>cortol</i>	Tolerance for rounding corners of the yield surface
<i>fymax</i>	Maximum value of the yield function
<i>pnltol</i>	Plastic unloading tolerance
<i>ikunl</i>	Flag for imposing unloading stiffness
<i>ikfac</i>	Factor used to scale the stiffness for generating the stiffness matrix

NON-LOCAL IMPLEMENTATION

In non-local models, the determination of the plastic strains follows a different approach. The consistency condition states that:

$$dF = 0 \Rightarrow \frac{\partial F^T}{\partial \sigma} \cdot d\sigma + \frac{\partial F}{\partial e_p^{nl}} de_p^{nl} = 0 \quad (1)$$

where dE_d^{nl} is the increment in non-local deviatoric strain. Moreover, the change in stress can be determined using:

$$d\sigma = [D] \cdot d\varepsilon^{el} = [D] \cdot (d\varepsilon - d\varepsilon^p) = [D] \cdot d\varepsilon - [D] \cdot d\varepsilon^p = [D] \cdot d\varepsilon - [D] \cdot \Lambda \cdot \frac{\partial P}{\partial \sigma} \quad (2)$$

Substituting in the previous equation:

$$\frac{\partial F^T}{\partial \sigma} \cdot [D] \cdot d\varepsilon - \frac{\partial F^T}{\partial \sigma} \cdot [D] \cdot \Lambda \cdot \frac{\partial P}{\partial \sigma} + \frac{\partial F}{\partial e_p^{nl}} de_p^{nl} = 0 \quad (3)$$

Rearranging:

$$\Lambda = \frac{\frac{\partial F^T}{\partial \sigma} \cdot [D] \cdot d\varepsilon + \frac{\partial F}{\partial e_p^{nl}} de_p^{nl}}{\frac{\partial F^T}{\partial \sigma} [D] \frac{\partial P}{\partial \sigma}} \quad (4)$$

There are different implementations available for the non-local models which are typically available through a selection parameter (*Nonlocal_{control}*):

<i>inonlocal</i>	Calculation of Λ	Update of hardening parameters
0	Local version	Plastic strains
1	Non-local version (Eq. 4)	Non-local strains during elastic and plastic stages
2	Non-local version (Eq. 4)	Non-local strains during plastic stages with softening only
3	Non-local version (Eq. 4)	Non-local strains during plastic stages

Care must be taken when using this assumption whereby non-local total strains \approx non-local plastic strains in situations where undrained loading is being modelled using a constitutive model where hardening is determined by volumetric strains. This is particularly problematic as non-local total strains will be approximately zero while plastic strains will be potentially vastly different, affecting the behaviour of the

constitutive model. This may also be an issue in highly restricted problems where zones of the soil mass are possibly experiencing negligible volume changes.

SAFETY ANALYSIS

The use of factors of safety in Safety Analysis is made possible by altering the consistency condition, adding a component related to the FoS :

$$dF = 0 \Rightarrow \frac{\partial F}{\partial \sigma} d\sigma + \frac{\partial F}{\partial k} dk + \frac{\partial F}{\partial FoS} dFoS = 0 \quad (5)$$

Since

$$d\sigma = [D] \cdot d\varepsilon^{el} = [D] \cdot (d\varepsilon - d\varepsilon^p) = [D] \cdot d\varepsilon - [D] \cdot d\varepsilon^p = [D] \cdot d\varepsilon - [D] \cdot \Lambda \cdot \frac{\partial P}{\partial \sigma} \quad (6)$$

Substituting:

$$\begin{aligned} \frac{\partial F}{\partial \sigma} \cdot [D] \cdot d\varepsilon - \frac{\partial F}{\partial \sigma} \cdot [D] \cdot \Lambda \cdot \frac{\partial P}{\partial \sigma} + \frac{\partial F}{\partial k} dk + \frac{\partial F}{\partial FoS} dFoS &= 0 \\ \Rightarrow \Lambda &= \frac{\frac{\partial F}{\partial \sigma} \cdot [D] \cdot d\varepsilon + \frac{\partial F}{\partial FoS} \cdot dFoS}{\frac{\partial F}{\partial \sigma} \cdot [D] \cdot \frac{\partial P}{\partial \sigma} - \frac{1}{\Lambda} \cdot \frac{\partial F}{\partial k} \cdot dk} \end{aligned} \quad (7)$$

Safety analyses are therefore only possible if the model has the differential $\partial F / \partial FoS$ defined.

INTEGRATION ALGORITHMS

Three options available:

1. Modified-Euler for elastoplastic models with substepping for **single yield surface**
`udsm_ialg = 1`
`meeppred.f90`
2. For **two yield surfaces**
`udsm_ialg = 2`
`meeppred02.f90`
3. Rate-type formulation (hypo-plastic type model with no yield surface)
`udsm_ialg = 3`
`meratepred.f90`

CYCLIC REVERSAL DETECTION ALGORITHM

Two options available:

1. Cyclic reversal detection in the elastic range following Taborda (2011)
`udsm_icyclic = 1` using the `elauload.f90` subroutine
2. Plastic cyclic reversal detection using Dafalias and Manzari (2004)
`udsm_icyclic = 2` using the `plauload.f90` subroutine

MODIFYING MODEL BEHAVIOUR USING ‘SPECIAL OPTION’

The special option in the definition of the stage under ‘staged construction’ can be used to modify the behaviour of the model during the analysis. The initial implementation of this process allowed each model to set the implementation of this option. However, in general, a conflict between implementations existed: the same code, which is applied to all active materials in that stage, triggered different operations for different models, potentially generating a conflict. To avoid this, a general framework is set below, to which IC MAGE models should adhere if they are to be used in conjunction with other IC MAGE models during an analysis.

A ‘special option’ is assumed to be an integer value where each of the digits is individually interpreted as a code triggering a specific operation. Below, the general description of the roles of each of the digits is provided, with the specific set of operations being described under the “Model usage” section of the manual entry for each model.

The expected format is:

$$I_9 I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1$$

where each of the digits is expected to trigger the following alterations:

I_1	Factor (1 to 9) by which the elastic stiffness matrix is scaled up. This potentially slows down convergence but reduces chances of divergence.
I_2	Set to 1 to reset all energy quantities
I_3	Set to 1 to reset hardening parameters in preparation for a High Cycle Accumulation control loop
I_4	Set to 1 to reset the plastic strain tensor
I_5	Set to 1 to reset the void ratio
I_6	Set to 1 to reset the elastic stiffness of the material, set to 2 to reset elastic stiffness while incrementing reversal counters
I_7	Set to 1 to reset any other hardening parameters related to the elastic response of the material
I_8	Set to 1 to reset the position of the yield surface
I_9	Set to 1 to reset any other hardening parameters related to the plastic response of the material

Note that not every option will be available (or implemented) in every model. The manual entry for each model will specifically indicate this. Moreover, the use of a negative number will trigger the UMIP to interpret this as a request to use the maximum stiffness matrix, independently of the actual value of I_1 , provided it is not 0.

ENERGY MODULE

When the energy module is activated, the UMIP adds 33 extra hardening parameters to the model in order to allow for the calculation of energy-related quantities using the algorithm described in Taborda et al. (2016). These hardening parameters are placed always at the end of the list of hardening parameters for the model so they are numbered as $N_{hard} + i$, where N_{hard} is the number of hardening parameters of the model being used and i varies between 1 and 33. The table below lists the parameters and their location:

HP1	$\varepsilon_{xx,ener,lag}$	Lag point strain state (x-component)
HP2	$\varepsilon_{yy,ener,lag}$	Lag point strain state (y-component)
HP3	$\varepsilon_{zz,ener,lag}$	Lag point strain state (z-component)
HP4	$\gamma_{xy,ener,lag}$	Lag point strain state (xy-component)
HP5	$\gamma_{yz,ener,lag}$	Lag point strain state (yz-component)
HP6	$\gamma_{xz,ener,lag}$	Lag point strain state (xz-component)
HP7	$\sigma_{xx,ener,lag}$	Lag point stress state (x-component)
HP8	$\sigma_{yy,ener,lag}$	Lag point stress state (y-component)
HP9	$\sigma_{zz,ener,lag}$	Lag point stress state (z-component)
HP10	$\tau_{xy,ener,lag}$	Lag point stress state (xy-component)
HP11	$\tau_{yz,ener,lag}$	Lag point stress state (yz-component)
HP12	$\tau_{xz,ener,lag}$	Lag point stress state (xz-component)
HP13	$\varepsilon_{xx,ener,rev}$	Reversal point strain state (x-component)
HP14	$\varepsilon_{yy,ener,rev}$	Reversal point strain state (y-component)
HP15	$\varepsilon_{zz,ener,rev}$	Reversal point strain state (z-component)
HP16	$\gamma_{xy,ener,rev}$	Reversal point strain state (xy-component)
HP17	$\gamma_{yz,ener,rev}$	Reversal point strain state (yz-component)
HP18	$\gamma_{xz,ener,rev}$	Reversal point strain state (xz-component)
HP19	$\sigma_{xx,ener,rev}$	Reversal point stress state (x-component)
HP20	$\sigma_{yy,ener,rev}$	Reversal point stress state (y-component)
HP21	$\sigma_{zz,ener,rev}$	Reversal point stress state (z-component)
HP22	$\tau_{xy,ener,rev}$	Reversal point stress state (xy-component)
HP23	$\tau_{yz,ener,rev}$	Reversal point stress state (yz-component)
HP24	$\tau_{xz,ener,rev}$	Reversal point stress state (xz-component)
HP25	$E_{def,lag}$	Deformation level at lag point
HP26	$E_{def,i}$	Current deformation level
HP27	$E_{acc,lag}$	Accumulated energy at lag point
HP28	$E_{acc,i}$	Current accumulated energy
HP29	$E_{elastic,i}$	Current elastic energy
HP30	$E_{dissipated,i}$	Current dissipated energy
HP31	ξ_i	Current damping ratio

HP32	$G_{sec,i}$	Current secant shear modulus
HP33	$K_{sec,i}$	Current secant bulk modulus

The algorithm can be called by specifying the parameter $Energy_{calc}$ in the model input. There are currently 18 possible options for this parameter, depending on the component and the interpretation of reversals:

Component ↓	Algorithm →	Original (Taborda et al., 2016)	Reversals based on E_g (Eq. 9)
xx		1	11
yy		2	12
zz		3	13
xy		4	14
yz		5	15
xz		6	16
Deviatoric, E_d		7	17
Volumetric, ε_{vol}		8	18
Global, E_g		9	19

This allows for the determination of energy-related quantities based on a single component (options 1 to 6 and 11 to 16), deviatoric behaviour (options 7 and 17), volumetric behaviour (options 8 and 18) or just the global size of the strain tensor (options 9 and 19). There are two implemented algorithms: the original one which detects reversals and integrates the behaviour in between reversals, providing estimates of secant stiffness and damping ratio which reflect the nature of the cycles (options 1 to 9) and a ‘standard’ algorithm which detects reversals using always the size of the entire strain tensor (E_g in Equation 9), meaning that energy measurements from different components are not influenced by the reversal identification procedure.

For options 7 and 17, the following tensors are used:

$$\{e_{jk}\} = \{\varepsilon_{jk}\} - I_3 \cdot \frac{\varepsilon_{vol}}{3} = \begin{bmatrix} \varepsilon_{xx} - \frac{\varepsilon_{vol}}{3} & \frac{\gamma_{xy}}{2} & \frac{\gamma_{xz}}{2} \\ \frac{\gamma_{xy}}{2} & \varepsilon_{yy} - \frac{\varepsilon_{vol}}{3} & \frac{\gamma_{yz}}{2} \\ \frac{\gamma_{xz}}{2} & \frac{\gamma_{yz}}{2} & \varepsilon_{zz} - \frac{\varepsilon_{vol}}{3} \end{bmatrix} \quad (8)$$

$$\{s_{jk}\} = \{\sigma_{jk}\} - I_3 \cdot p = \begin{bmatrix} \sigma_{xx} - p & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_{yy} - p & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_{zz} - p \end{bmatrix} \quad (9)$$

whereas the deformation level is given by E_d .

For options 8 and 18, the volumetric strain and mean effective stress is used in the quantification of soil behaviour, with the deformation level being given by ε_{vol} .

For options 9 and 19, the full stress and strain tensors are used:

$$\{\varepsilon_{jk}\} = \begin{bmatrix} \varepsilon_{xx} & \frac{\gamma_{xy}}{2} & \frac{\gamma_{xz}}{2} \\ \frac{\gamma_{xy}}{2} & \varepsilon_{yy} & \frac{\gamma_{yz}}{2} \\ \frac{\gamma_{xz}}{2} & \frac{\gamma_{yz}}{2} & \varepsilon_{zz} \end{bmatrix} \quad (10)$$

$$\{\sigma_{jk}\} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_{zz} \end{bmatrix} \quad (11)$$

with the deformation level being calculated using:

$$E_{def} = \sqrt{\{\varepsilon_{jk}\} : \{\varepsilon_{jk}\}} = \sqrt{(\varepsilon_{xx})^2 + (\varepsilon_{yy})^2 + (\varepsilon_{zz})^2 + \frac{(\gamma_{xy})^2}{2} + \frac{(\gamma_{xz})^2}{2} + \frac{(\gamma_{yz})^2}{2}} \quad (12)$$

RANDOM FIELDS

The UMIP supports the generation of Random Fields in 2D analyses if the constitutive model has been formulated to do so. At the moment, the UMIP uses the Cholesky decomposition of the 2D covariance matrix to generate the field in a regular rectangular grid. If the grid has dimensions $n_y \times n_x$ (n_x values along direction x and n_y values along direction y), the covariance matrix has dimensions $n_x n_y \times n_x n_y$ and is assembled using the following algorithm:

for each line i of the grid (out of n_y lines) and for each line j of the grid (out of n_y lines)

distance between the lines: $dy = y_j - y_i$

select value k of line i (out of n_x values)

select value l of line j (out of n_x values)

calculate horizontal and vertical distances between the two points

distance within the line: $dx = x_l - x_k$

total distance: $d = \sqrt{(dx/\theta_x)^2 + (dy/\theta_y)^2}$

calculate the Markov function: $c = \exp(-2 \cdot d)$

store the value in covariance matrix: $C((i-1) \cdot n_x + k, (j-1) \cdot n_x + j)$

The Cholesky decomposition is then applied to generate the lower triangular L matrix using:

$$L_{jj} = \sqrt{C_{jj} - \sum_{k=1}^{j-1} L_{jk}^2}, \quad j = 1, n_x n_y$$

$$L_{ij} = \frac{1}{L_{jj}} \left(C_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} \right), \quad \text{for } i > j \text{ and } j = 1, n_x n_y$$

A vector V of dimension $n_x n_y$ containing random values according to a normal distribution with mean 0 and standard deviation 1 is then generated and multiplied by L , creating the random field vector T which is then reshaped into a matrix of size $n_y \times n_x$ which maps onto the grid created:

$$T = L^T \times V$$

The dimension of L is $n_x n_y \times n_x n_y$ and V is $n_x n_y \times 1$, meaning that the dimension of T is $n_x n_y \times 1$ (i.e. it is a vector). Once T is reshaped into M , a $n_y \times n_x$ matrix, the mean (μ) and standard deviation (σ) are then corrected:

$$M_{ij} = M_{ij} \cdot \sigma + \mu$$

This correction is different for log-normal distributions. In that case, matrix M is calculated using:

$$M_{ij} = M_{ij} \cdot \sigma^* + \mu^*$$

where

$$\sigma^* = \sqrt{\ln\left(\left(\frac{\sigma}{\mu}\right)^2 + 1\right)}$$

$$\mu^* = \ln(\mu) - 0.5 \cdot (\sigma^*)^2$$

The random sample of normally-distributed values (V) with mean 0 and variance 1 is generated using:

$$v = \sqrt{-2.0 \cdot \ln(v_1)} \cdot \cos(2\pi \cdot v_2)$$

where v_1 and v_2 are two randomly-generate values uniformly distributed between 0 and 1.

After the fields are generated, the log files provide the following data:

- Mean value for the calculated grid:

$$\mu = \frac{1}{n_x \cdot n_y} \cdot \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} M_{ij}$$

- A standard deviation for the calculated grid:

$$\sigma = \sqrt{\frac{1}{n_x \cdot n_y} \cdot \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} (M_{ij} - \mu)^2}$$

- For variables for which a normal distribution has been requested, the parameters above are the normal distribution parameters.
- For variables for which a log-normal distribution has been requested, the following distribution parameters are computed based on the values of μ and σ above:

$$\mu_{log-normal} = \exp\left(\mu + \frac{\sigma^2}{2}\right)$$

$$\sigma_{log-normal} = [\exp(\sigma^2) - 1] \cdot (2\mu + \sigma^2)$$

- The values of μ and σ , normalised by the input value, to assess how close the resulting statistics are to the desired ones.

The generation of cross-correlated fields is also possible for UMIP v3.1 and above. If requested by the model, the procedure above is performed with the following changes:

- the vector containing the random values with mean 0 and standard deviation 1 (V) is replaced by a matrix \bar{V} with dimensions $n_x n_y \times n_f$ where n_f is the number of cross-correlated fields to be generated.
- before multiplying by the decomposed covariance matrix, L , matrix \bar{V} is multiplied by the decomposed cross-correlation matrix, R :

$$T = L^T \times (\bar{V} \times R)$$

Matrix R is obtained by applying the Cholesky decomposition to the cross-correlation matrix, which has dimensions $n_f \times n_f$.

For three fields, the cross-correlation matrix has the form

$$\rho = \begin{bmatrix} 1.0 & \rho_{12} & \rho_{13} \\ \rho_{12} & 1.0 & \rho_{23} \\ \rho_{13} & \rho_{23} & 1.0 \end{bmatrix}$$

where ρ_{ij} is the coefficient denoting the cross-correlation between variables i and j for which values between -1 and 1 are possible. Values of -1 and 1 represent perfect correlation between the two variables (albeit with a different relationship), where as a value of 0 means the two variables are not correlated at all.

The value at a given Gauss Point is then obtained through inverse distance weighting with a p -value of 1. According to this method, for each Gauss Point of coordinates $P(x_c, y_c)$, the interpolated quantity is given as:

$$u(x_c, y_c) = \begin{cases} \frac{\sum_{i=1}^N w_i(P) \cdot u_i}{\sum_{i=1}^N w_i(P)}, & d_i(P) \neq 0 \text{ for all } i \\ u_i, & d_i(P) = 0 \text{ for some } i \end{cases}$$

where u_i is the value of the quantity at the grid point (x_i, y_i) and the weight $w_i(P)$ is given as:

$$w_i(P) = \frac{1}{[d_i(P)]^p} \text{ and } d_i(P) = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$$

REFERENCES

- Azeiteiro, R.J.N. (2021) Numerical simulation of liquefaction-related phenomena. PhD Thesis, University of Coimbra, Portugal.
- Dowell, M. and Jarratt, P. (1972). The ‘Pegasus’ method for computing the root of an equation. BIT Numerical Mathematics, 12 (4), 503–508.
- Hong, P.Y., Pereira, J.M., Cui, Y.J. and Tang, A.M. (2012). Explicit integration of a thermo-mechanical model for clays. Computers and Geotechnics, 46, 13–25.
- Potts, D.M. and Gens, A. (1985). A critical assessment of methods of correcting for drift from the yield surface in elasto-plastic finite element analysis. International Journal for Numerical and Analytical Methods in Geomechanics, 9 (2), 149–159.
- Potts, D.M. and Zdravkovic, L. (1999). Finite element analysis in geotechnical engineering. Theory. London, U.K., Thomas Telford Publishing.
- Sloan, S.W. (1987). Substepping schemes for numerical integration of elasto-plastic stress-strain relations. International Journal for Numerical Methods in Engineering, 24 (5), 893–911.
- Sloan, S.W., Abbo, A.J. and Sheng, D. (2001). Refined explicit integration of elastoplastic models with automatic error control. Engineering Computations, 18 (1), 121–194.
- Taborda, D.M.G., Potts, D.M. and Zdravković, L. (2016). On the assessment of energy dissipated through hysteresis in finite element analysis. Computers and Geotechnics, 71, 180–194.
- Taborda, D.M.G and Zdravković, L. (2012). Application of a Hill-Climbing technique to the formulation of a new cyclic nonlinear elastic constitutive model. Computers and Geotechnics, 43, 80-91.