

AUTOMATIC CONSTRUCTION OF GROUND MODELS FOR 3D FINITE ELEMENT ANALYSIS BY DATA PROCESSING

Ayana Tamura^{1*} and Tomohide Takeyama²

^{1,2} Graduate School of Engineering, Kobe University, Japan;

*Corresponding Author, Received: 12 Feb. 2023, Revised: 01 March 2023, Accepted: 19 March 2023

ABSTRACT: In the construction industry, 3D models are often used to share information among stakeholders. It is expected that the 3D model can be used for numerical analysis to perform detailed and advanced simulations. However, the models that can be constructed with commercially available 3D modeling software cannot be used for numerical analysis or can only be used with specific analysis tools. In addition, because commercial products are sold in packaged form, the use of the models is extremely limited. In 2021, Takeyama et al. developed a method for constructing a Grid Ground Model, which is a general purpose mediation model with general ground information, including boring data and soil test results. By building a common data structure once, flexible data manipulation can be performed. Based on this technology, this study developed a model building function that can be used for more advanced applications, including analysis using finite element analysis (FEA), by providing a high level of generality in the use of internal parameters and the geometry processing process. Using the developed technology, a detailed numerical analysis model was constructed through simple operations. This technology is useful for disaster damage estimation and advanced design or maintenance of construction works. In addition, the models can be highly integrated with a wide variety of software, contributing to a significant reduction in the number of man-hours required for work and to the advancement of the use of 3D models in the future.

Keywords: Boring Data, Data Processing Platform, 3D model, Grid Ground Model, Finite Element Analysis

1. INTRODUCTION

National and local governments manage boring data through platforms, such as the National Land Transport Data Platform [1]. Various methods have been proposed for the construction of geotechnical databases based on boring data, including the National Electronic Ground Map [2], which is a database of surface geotechnical information in Japan, a GIS application for Seoul developed by Chang et al. [3] and a geological framework model for the Glasgow area developed by Merritt et al. [4]

In the construction industry, such databases are often used to create 3D models to share information among stakeholders and to speed up consensus building. 3D models, if used for numerical analysis, can be useful for detailed disaster damage estimation and advanced design or maintenance of construction works. It has been found that seismic simulations using models based on geotechnical information from boring data can explain the periodic characteristics of seismic motions [5]. Honda et al. [6] constructed a wide-area analytical model based on the Tokyo boring database to assess the risk of liquefaction. As in this case, it is desirable to conduct various analyses in different regions. However, models that can be constructed using the technology in this case study or commercially available software cannot be used for numerical analysis or can only be used with specific

analysis tools. In addition, commercial modeling technology is sold in packaged form, so the use of models is extremely limited.

In 2021, Takeyama et al. [7] proposed a ground modeling method focused on generalization of data processing. The 3D ground modeling technology utilizes the data processing platform (DPP) [8], a program developed by RIKEN, for data processing. The DPP is a data processing technology that enables data sharing according to various needs by extracting essential quantities of data and converting them into intermediate data consisting of a common structure. Users can execute the data processing steps by writing scripts if these steps are predefined as DPP functions. Takeyama et al. developed a series of systems that extract various types of soil information as intermediate data from various types of geotechnical data, such as boring data and soil test values, by writing simple scripts and constructing a Grid Ground Model as a mediation model. This technology is increasingly being used in the field of ocean development [9]. If a DPP-based mediation model can be used for finite element analysis (FEA), it can be applied to various software and configuration models and would be very useful. However, the current Grid Ground Model cannot be used directly for FEA because it comprises a grid of points with parameter information. Soils properties typically differ

according to their classification and are distributed over an area (often in layers). Therefore, the ground model for FEA should reflect the soil classification information in terms of the physical properties of each element and mesh geometry. Two functions are necessary for using the Grid Ground Model for FEA: appropriate parameter information for each soil type and the ability to set a detailed mesh geometry along the soil type boundaries.

This study extended the method reported by Takeyama et al. by focusing on two functions. First, the parameterization system was generalized by defining an object (ParamSetter) comprising a combination of specific conditions and parameter values or calculation formulas to enable general parameter setting. The second function was to compensate for the lack of shape information in the Grid Ground Model by defining two new mediation models: the "SolidModel," representing the regions where particular attribute quantities are distributed, and "FEModel," a model in which each region of the SolidModel is divided into elements. Various models with any parameter can be constructed by applying the ParamSetter to the mediation model. Consequently, the user has a wide range of applications by performing two tasks: defining the ParamSetter and creating a library that performs processes on the mediation model. After setting up the ParamSetter to determine the parameters of a particular constitutive model, the user can prepare a library for a software program and create a function that outputs the mediation model information in a specific format which can be applied to any software program. However, it is not useful if these tasks require considerable time and effort. Therefore, the usefulness of the extended technique was demonstrated through parameterization examples and library implementation for building analysis files for the FEM simulator DACSAR-I [10] (EC model [11] as a constitutive model).

The technology developed in this study can construct innovative 3D models that can be used in a wide range of applications by providing a high level of generality in the file formats handled, the use of internal parameters, and the geometry processing process. The proposed general-purpose model can be highly integrated with a wide variety of software, contributing to a significant reduction in the number of man-hours required for work and to the advancement of the use of 3D models in the future. The remainder of this paper is organized as follows: Section 2 discusses the research significance. Section 3 describes a new method for determining material parameters (ParamSetter). Section 4 describes the procedure for generating mediation models (Grid Ground Model, SolidModel, and FEM models). Section 5 presents a practical example of the DACSAR-I. Finally, Section 6 presents the conclusions of this study.

2. RESEARCH SIGNIFICANCE

This study developed a technique for constructing a 3D model of the ground for finite element analysis based on boring data and soil test results. This technology constructs a mesh considering the distribution area of the soil classification. It has a general purpose parameter system that enables the construction of analytical models compatible with various software and constitutive models. Linking with various data platforms enables the advanced utilization of electronic data accumulated in the government and private sectors, leading to innovations.

3. PARAMETER SETTING

The numerical analysis software incorporates constitutive models to represent material properties. Many constitutive models for geomaterials have been proposed, including the Cam-clay model [12]. Each model requires different material parameters. For some parameters, estimating equations have been proposed for when laboratory test results are lacking. For example, part of the method proposed by Miyata [13] for estimating the material parameters of sandy soil is illustrated in Fig. 1. Similarly, estimation methods were proposed by Iizuka et al. [10] for the material parameters of clayey soils. In general, the parameter values are calculated according to the proposed formulas to obtain information not included in the underlying data, or the values are fitted by referring to previous reports and papers. This study developed ParamSetter as an object for performing such operations. ParamSetter is a DPP object with a "condition" and "value or calculation formula of a material parameter to be applied when the condition is true". A DPP object is an instance of a data type defined in a DPP that can be handled using scripts. There is no need to modify the library code because it can be set in a script. Fig. 2 shows an example of a script used to set ParamSetter. Fig. 2(a) shows a script for defining categories according to the soil-rock type classification of the boring data. In line 1, the ParamSetter library is loaded. In line 2,

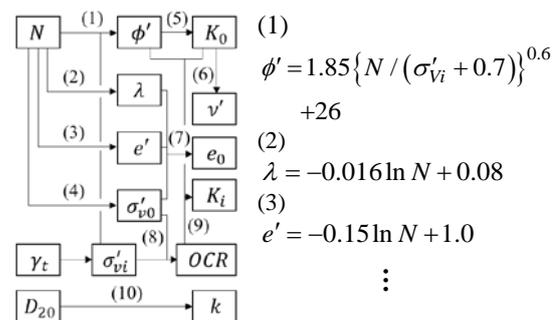


Fig. 1 Material parameters for sand soil.

```

1 library("ParamSetter");
2 ParamSetter ps_category;
3 ps_category.addCondition
  ( 1, "{SoilTypeName} == Landfill" );
4 ps_category.addCondition
  ( 2, "{SoilTypeName} == Sandy soil" );
5 ps_category.addCondition
  ( 2, "{SoilTypeName} == Sand" );
6 ps_category.addCondition
  ( 3, "{SoilTypeName} == Clay" );
7 ps_category.addCondition
  ( 4, "{SoilTypeName} == Gravel" );
8 ps_category.addFunction
  ( "{Category} = Landfill", id=1 );
9 ps_category.addFunction
  ( "{Category} = Sand", id=2 );
10 ps_category.addFunction
  ( "{Category} = Clay", id=3 );
11 ps_category.addFunction
  ( "{Category} = Gravel", id=4 );

```

(a) ParamSetter for categories

```

1 library("Quantity");
2 ParamSetter ps_material;
3 ps_material.addCondition
  ( 1, "{Category} = Landfill &&
      [NValue] >= 5" );
4 ps_material.addCondition
  ( 2, "{Category} == Landfill &&
      [NValue] < 5" );
5 ps_material.addCondition
  ( 1, "{Category} == Sand" );
6 ps_material.addCondition
  ( 3, "{Category} == Clay" );
7 ps_material.addCondition
  ( 4, "{Category} == Gravel" );
8 ps_material.addFunction
  ( "[D20,mm] = 0.4", id=1 );
9 ps_material.addFunction
  ( "[D20,mm] = 0.05", id=2 );
10 ps_material.addFunction
  ( "[D20,mm] = 0.005", id=3 );
11 ps_material.addFunction
  ( "[D20,mm] = 2.0", id=4 );
12 ps_material.addFunction
  ( "[sigv0,kN/m2] = 40*[NValue]" );
13 ps_material.addFunction( "[kw,m/s] =
    0.0034*pow([D20,mm],2.2954)" );

```

(b) ParamSetter for material parameters

Fig. 2. Script to set ParamSetter.

ParamSetter is constructed as a DPP object. The third to seventh lines use the DPP function "addCondition" to define the combination of a condition and an ID. The first argument is the ID, and the second argument is the condition. Lines with the same ID (Lines 4 and 5) indicate a logical AND relationship. Line 8 uses the DPP function "addFunction" to define the parameter values or calculation expressions. The first argument defines the formula to be applied, and the second argument defines the conditions under which the formula is applied. The categories are set according to the IDs. Fig. 2(b) shows a script for defining the values of various parameters or calculation formulas. As with Fig. 2(a), the second line constructs ParamSetter, which is a DPP object. The third to seventh lines set the ID using the addCondition function, and the rest of the lines set the parameters using the addFunction function. The "&&" symbol indicates a logical AND. For example, the description in line 3 means that "ID=1" is assigned under the condition that "Category is Landfill AND N-value is 5 or more". In addition, the function of the quantity library loaded in the first line allows the parameters to be specified in units by appending commas. For example, in line 9, when ID=2, the value of D_{20} , which means a 20% grain size, is specified with a unit of 0.05 (mm). The units are automatically converted during data processing. The 12th and 13th lines are defined by excerpting two estimation equations in the parameter determination chart shown in Fig. 1: the prior consolidation effective stress σ'_{v0} and permeability k_w . If the second argument is not given to the addFunction function, it unconditionally validates the equation of the first argument. k_w is expressed as a function of D_{20} ; thus, the value of D_{20} specified in lines 8–11 is used according to the ID. Therefore, the calculation method is processed recursively when it is defined as a relational expression between multiple parameters. The calculations are performed sequentially, starting with the calculable parameters. Other detailed settings are also possible, such as defining multiple estimation formulas in priority order and preparing multiple alternative simple formulas in case data is insufficient. The parameter settings are defined as a single object and operated in the code to perform general purpose parameter setting rather than reading fixed parameter information and performing fixed processing in the code.

4. GENERATING MEDIATION MODELS

The DPP includes the function "BoringXML"[7]. The function reads boring data in extended markup language (XML) format published in KuniJiban [14], a geotechnical information search site of the Ministry of Land,

Infrastructure, Transport, and Tourism.

Fig. 3 shows an example script for loading the boring data. The first three lines load the necessary libraries: the first line loads the XSD library, a library for handling XML documents; the fourth line uses the BoringXML function. The BoringXML function reads the XML file specified in the argument (or an XML file in a directory), stores the contents of the XML file once in an XML DPP object with a structure comprising a tag name/value pair, and constructs a Boring DPP object (Fig.3 bdata). The Boring DPP object created contains boring data, such as latitude, longitude, maximum depth, borehole mouth elevation, water level in the borehole, standard penetration test results (N value), and soil-rock type classification. The Grid Ground Model, a mediation model, can be created based on this information.

4.1 Grid Ground Model

The Grid Ground Model is a mediation model comprising a grid of points with parameter information, as shown in Fig. 4. This model does not differ significantly from the Grid Ground Model developed by Takeyama et al. [7]. However, two primary changes were made. First, the model can be built without parameters other than category (e.g., N values). This change was made by considering the case where the Grid Ground Model is used as a basic model for constructing other mediation models; when the Grid Ground Model is used as a basic model, the parameters that are not necessary for constructing subsequent models do not need to be calculated. Second, a digital elevation model (DEM) was used to determine the ground surface shape. A DEM can handle numerical elevation model [15] published by the Geospatial Information Authority of Japan (GSI) and bathymetry data [16] published by the Japan Oceanographic Data Center. Fig. 5 shows a script for creating a DPP object Ground::DEM, based on the numerical elevation model. The first line loads the DEM library, and the second line prepares the Ground::DEM object, which is a DPP object. The addData function in the third line is a DPP function for adding data to the Ground::DEM object. The GSI_DEM function used in the argument reads in a numerical elevation model file, converts it to Ground::DEM type, and returns it. The first argument of this function is the file path, and the second argument, origin_no, is the origin number of the plane rectangular coordinate system [17] defined by the GSI. Similarly, Ground::DEM objects can be constructed from various types of elevation data, such as drone

survey results, by defining functions to read user-specific data.

```

1 | library("XSD");
2 | library("Quantity");
3 | library("Ground");
4 | bdat = BoringXML( dir_xml );

```

Fig. 3 Script for reading boring data.

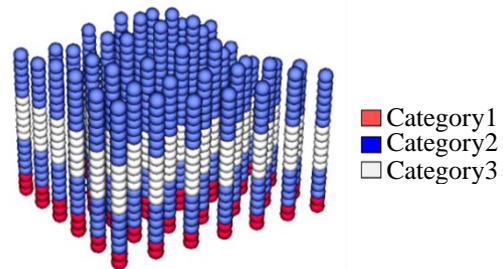


Fig. 4 Schematic of Grid Ground Model.

```

1 | library("DigitalElevationModel");
2 | Ground::DEM dem;
3 | dem.addData
  | ( GSI_DEM ( fpath_gsi, origin_no=5 ) );

```

Fig. 5 Script for reading DEM data.

```

1 | library("XSD");
2 | library("Ground");
3 | library("Geometry");
4 | library("DataTable");
5 | source( dem_script_fig5 );
6 | source( ps_script_fig 2-a );
7 | source( ps_script_fig 2-b );
8 | bdat = BoringXML( dir_boring );
9 | gbdat =Ground::GridBoringSet
  | ( bdat, ps_category, zspacing="1m" );
10 | GridModel = Ground::GridModel
  | ( gbdat, dem,
  |   xspacing="2m", yspacing="2m" );
11 | SolidModel = Ground::SolidModel
  | ( gridmodel, ps_category );
12 | FEModel = Ground::FEModel
  | ( SolidModel, ps, dirpath );

```

Fig. 6. Script for building mediation models.

Fig. 6 shows the scripts for generating various mediation models. Lines 10 and above are primarily used for creating the Grid Ground Model. Lines 1–4 load the necessary libraries, and lines 5–7 load the scripts for creating the DEM object (Fig. 5) and ParamSetter (Fig. 2). Thus, the DPP can read scripts written in another file using the source command. Line 8 constructs the DPP object for the boring data, as shown in Fig. 3. The `Ground::GrigBoringSet` function in line 9 is used for constructing an object with equally spaced parameters in the depth direction. A category is assigned to each grid based on ParamSetter (second argument) after generating grids aligned in the depth direction at each boring test location based on the Boring DPP object (first argument). The resulting one-dimensional model group is stored in a DPP object `gridbdat`. The third argument, `zspacing`, sets the depth spacing of the grids with units. This vertical interpolation process speeds up the interpolation of the category when creating the Grid Ground Model. The 3D interpolation of parameters is computationally demanding, particularly when generating a high-resolution model. Therefore, one-directional (1D) completion in the depth direction is performed in advance, and the two-dimensional (2D) interpolation formula is used. In line 10, the `Ground::GridModel` function creates a Grid Ground Model based on the Boring DPP (first argument) and `Ground::DEM` (second argument) objects. The third and fourth arguments, `xspacing` and `yspacing`, set the grid spacing in the xy direction with units. In the current Grid Ground Model, the category of each grid is interpolated in the same way as the category of the same depth in the nearest-neighbor boring data. Consequently, a Grid Ground Model with a structure in which categories are distributed systematically in 3D is created. In addition to the category, the Grid Ground Model can be used as a ground model using the inverse distance weighted method to complement the N values of the boring data and by assigning various parameters using ParamSetter.

4.2 Solid Model

`SolidModel` is a model with surfaces surrounding the area where grids of the same category are distributed among the grids comprising the Grid Ground Model, as shown in Fig. 7. The surfaces comprise multiple points, lines, and planes and form a closed region (volume). The `SolidModel` is constructed in a script using the `Ground::SolidModel` function, as shown in line 11

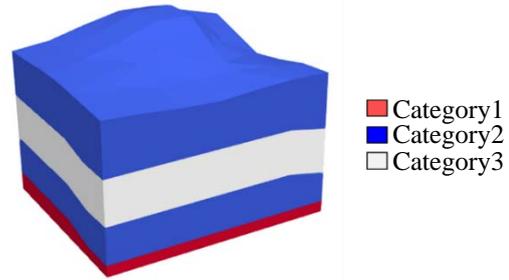


Fig. 7. Schematic of SolidModel.

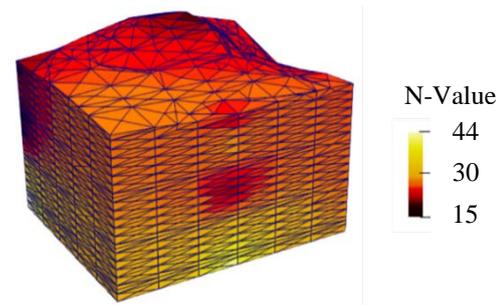


Fig. 8. Schematic of FEModel.

of Fig. 6. This function has a Grid Ground Model DPP object as its first argument and a ParamSetter as its second argument. This function refers to each grid in the Grid Ground Model and generates a polygon-like boundary surface comprising triangles to enclose the area where grids of the same category exist. The marching cube method [18] is used for this process. This geometric information processing is implemented in the geometry library, loaded in line 3 of Fig. 6. Consequently, the `SolidModel` is constructed as a model with a distribution area for each category set by the user in ParamSetter.

4.3 FEModel

`FEModel` is a model of the structure with information about the mesh and parameters of each element necessary for FEA, as shown in Fig. 8. Nodes have IDs and coordinate values. Elements primarily have information, such as ID, TYPE (information on element shape and number of constituent nodes), parameter objects, and IDs of nodes comprising the element. A model suitable for FEA with geometry and physical property information can be constructed by applying the ParamSetter to the parameter objects of the element. Line 12 of Fig. 6 shows the script for constructing the `FEModel`. The `Ground::FEModel` function constructs an `FEModel` object from a `SolidModel` DPP object (first argument) and ParamSetter (second argument). The third argument is the path of the directory where the mesh-related files are to be outputted. `SolidModel`, the model on which the `FEModel` is based, has a polygon mesh surrounding

the area where each category is distributed. The Ground::FEModel function performs meshing for each region, such that a mesh along the category boundary can be generated. As shown in Fig. 2(a), the model will contain a mesh along the boundary of the soil layer when the category is set according to the soil-rock type classification. The free meshing software Gmsh C++ API [19] is used for the meshing process. Other software can be used if the library is properly maintained.

In addition, the attribute information of a string can be provided to a face or region (volume) of SolidModel, and the FEModel element can inherit the information. For example, suppose that the attribute "Boundary Surface" is given to one side of SolidModel. Any element built on the side with the attribute "Boundary Surface" will retain that attribute when an FEModel is created based on the SolidModel using the Ground::FEModel function. This function can be used in various ways during FEA, such as creating a model with extracted regions or automating the setting of boundary conditions.

5. APPLICATION TO DACSAR-I

Software must have a specific format. The mediation models built through DPP serve as the underlying intermediate data to create models for various software. The mediation model can be used for any software by outputting the mediation model information in a specific format. A function for constructing an analysis model for DACSAR-I [10], an FEM geotechnical analysis software, was developed to demonstrate the use of the developed technology for constructing an FEA model for a specific software. A standard model comprising material properties, initial stress conditions, and geometric information, such as nodes and elements, was generated based on a mediation model. The EC model [11], an elastoplastic constitutive model of geomaterials implemented in DACSAR-I, was used as the constitutive model. Twelve boring data points around Uzumoridai, Sumiyoshiyamate, Nada-ku, Kobe, which are available on KuniJiban [14], were used as the ground data (Fig. 9 and 10). The modeling domain was defined as the x- and y-coordinate ranges of -139337.0 m to -139717.0 m and 83490.9 m to 84250.9 m, respectively, in the plane rectangular coordinate system. The numerical elevation model [15] of the Fundamental Geospatial Data was used as the DEM data.

First, the process of setting the category and determining the necessary parameters using ParamSetter is discussed in this section. The category was set by classifying the soil-rock types in the data into three primary categories: sand, clay, and gravel. As shown in Fig. 2(a), the script has 28

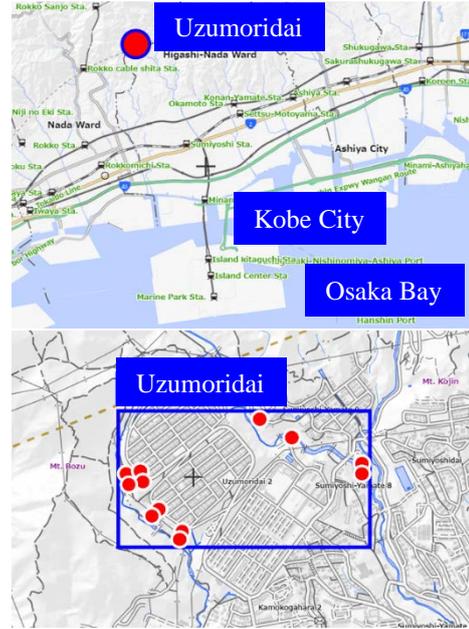


Fig. 9 Map of areas to be modeled.

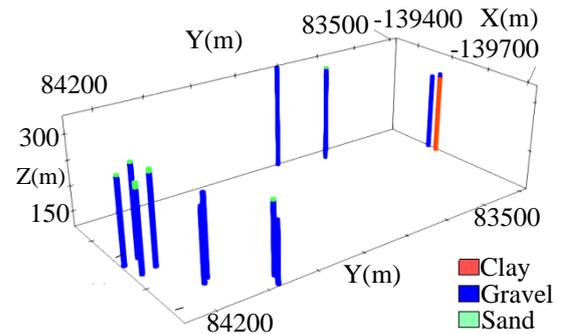


Fig. 10. Boring data group read (plane rectangular coordinate system display).

lines when the category is set for each soil type in the boring data used in this study. In addition to the parameters shown in Fig. 1, the critical stress ratio M , irreversible ratio Λ , and dilatancy coefficient D are necessary parameters of the EC model. Each parameter was defined and estimated using Eq. (1) and (3), respectively.

$$M = \frac{6 \sin \phi'}{3 - \sin \phi'} \quad (1)$$

$$\Lambda = \frac{M}{1.75} \quad (2)$$

$$D = \frac{\lambda \Lambda}{M(1 + e_0)} \quad (3)$$

Fig. 11 shows the scripts for defining the formulas for these three parameters. The total script

length is 34 lines when expressions are defined for all the parameters of the EC model. Thus, the total number of lines in the script related to ParamSetter is 62. Depending on how detailed the category and parameters are, they are generally considered to be several dozen to several hundred lines. Scripts can be loaded from other scripts; thus, creating template scripts for setting categories for representative soil types and determining parameters for various constitutive models and using them in combination can be performed to significantly reduce the amount of description each time.

Subsequently, a series of processes for constructing various mediation models were described in a script. This process can be executed in a script similar to that shown in Fig. 6. The script can be written in approximately 10 to 20 lines, including the loading of the library. Figs. 12–14 show a visualization of the various mediation models constructed. The FEModel shown in Fig. 14 visualizes the value of the critical stress ratio, M , a parameter of the EC model specified in the first line of Fig. 11, as a reference. The grid spacing of the Grid Ground Model in Fig. 12 was 10 m in the x- and y-directions and 4 m in the vertical direction in the plane rectangular coordinate system.

Finally, the DACSAR library should be created, and the outputDACSAR function was developed as a DPP function to output FEModel in the format required by DACSAR-I. This function outputs information held by the FEModel object in the file format required by DACSAR-I. The output file is a standard DACSAR-I model comprising material properties, initial stress conditions, and geometric information, such as nodes and elements. The OutputDACSAR function developed in this study comprised 150 lines, and the entire DACSAR library had 180 lines. The number of lines of a code does not necessarily indicate the complexity of the program; however, most of the code content are file output instructions, which can be easily written by someone familiar with programming. As described above, scripts and libraries for building files for a specific software can be written without extensive effort.

```

1 | ps.addFunction( "[M]
   |     = 6*sin([phi,rad])/(3-sin([phi,rad])) );
2 | ps.addFunction( "[LAMB] = [M]/1.75" );
3 | ps.addFunction( "[D]
   |     = [lamb]*[LAMB]/([M]*(1+[e0])) );

```

Fig. 11. Script for setting ParamSetter of ECmodel.

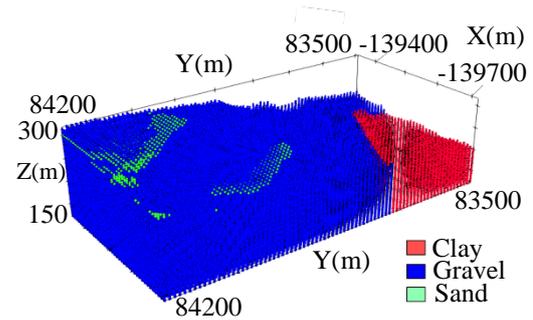


Fig. 12 Visualization of Grid Ground Model

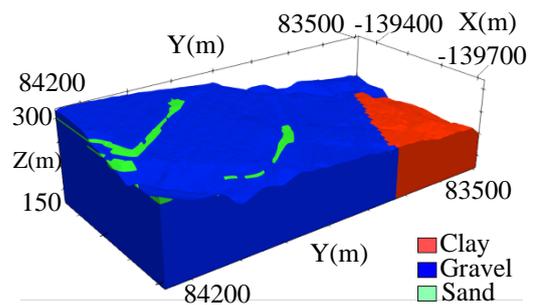


Fig. 13 Visualization of SolidModel: Areas by category

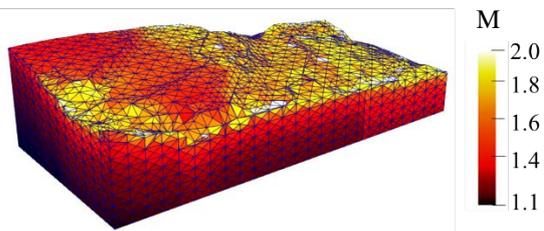


Fig. 14 Visualization of FEModel: the color map is a parameter of the ECmodel, M

6. CONCLUSION

This study extended in two ways the method of using a DPP to create ground mediation data reported by Takeyama et al. The first was a more generalized parameterization system that enables generic parameterization. This enables convenient introduction of parameters, including those expressed as complex functions. The second was the development of two new mediation models: the FEModel, which assumes FEA, and SolidModel, which has regional information that serves as the basis for constructing the FEA model. Implementing these two models enabled more advanced operations, such as developing a mediation model with a mesh along the stratum boundary.

Using the developed program, three mediation models (Grid Ground Model, SolidModel, and

FEModel) were developed based on the boring data and ground parameter estimation equations. A 3D analysis model for DACSAR-I was developed to demonstrate the use of mediation models; it was confirmed that the process did not require much effort or advanced skills. The same procedure can be applied to various software packages. Although this study was limited to building a standard model, users can perform various processes according to the design of the library. For example, users can integrate with other software, such as a visualization software, or automate the setting of boundary conditions.

The developed technology can be used, for example, in disaster prevention, to construct wide-area models based on detailed numerical analysis and to create advanced hazard maps. In the construction process, man-hours can be reduced by making the created models available for use in various types of software. In future works, the usability of the mediation model can be extended by increasing the number of methods for creating the mediation model, such as building the SolidModel by reading data from drawings, including stratigraphic maps and CAD data.

7. REFERENCES

- [1] Ministry of Land Infrastructure, Transport and Tourism, Land Infrastructure, Transport and Tourism Data Platform (tentative name) Development Plan, May. 2019, pp. 1-20.
- [2] Murakami, S., Yamamoto, K. and Wakabayashi, R., Development of National Electronic Geoinformation Maps and its Applications, Geotechnical Engineering Magazine, Vol. 61, 2013, pp. 12-15.
- [3] Y.-S. Chang, and H.-D. Park, Development of a web-based Geographic Information System for the management of borehole and geological data, Computers and Geosciences, Vol.30, 2004, pp. 887-897.
- [4] J. Merritt, D. Entwisle, and A. Monaghan, Integrated geoscience data, maps and 3D models for the City of Glasgow, UK, IAEG Paper 394, 2006, pp.1-10.
- [5] T. Takeyama, K. Honda, S. Tachibana, A. Iizuka: A procedure to determine material parameters based on borehole data in dynamic analysis, International Journal of GEOMATE, Vol.21, Issue 85, 2021, pp.1-9.
- [6] K. Honda, T. Takeyama, S. Tachibana, A. Iizuka: Liquefaction risk assessment in the 23 wards of Tokyo using elastoplastic analysis, International Journal of GEOMATE, Vol.21, Issue 86, 2021, pp.48-54.
- [7] Takeyama, T., O-Tani, H., Oishi, S., Hori, M. and Iizuka, A., Automatic Construction of Three-Dimensional Ground Model by Data Processing, IEEE-ASME Transactions on Mechatronics, Vol.26, No6, 2021.12, pp. 2881-2887.
- [8] Fujiwara, H., Nakamura, H., Senna, S., Otani, H., Tomii, N., Ohtake, K., Mori, T. and Kataoka, S., Development of a real-time damage estimation system, Journal of Disaster Research, Vol.14, No.2, 2019, pp. 315-332.
- [9] Ito, H., Takeyama, T. and O-Tani, H., Developments of an Automatic Setting System of Input Data and a Linkage System for Earthquake Simulation and Tsunami Simulation, Proceedings of JSCE, Vol. 78, No2, 2022, pp. 889-894.
- [10] Iizuka, A. and Ohta, H., A determination procedure of input parameters in elastoviscoplastic finite element analysis, Soils and Foundations, Vol. 27, No. 3, 1987, pp. 71-87.
- [11] Ohno, S., Iizuka, A. and Hideki Ota, Two categories of new constitutive model derived from non-linear description of soil contractancy, Journal of Applied Mechanics, Vol.9, 2006, pp. 407-414.
- [12] Roscoe, K. H., Schofield, A. N. and Thurairajah, A., Yielding of clays in states wetter than critical, Géotechnique, Vol. 13, No. 3, 1963, pp. 211-240.
- [13] Miyata, T., Deformation of stability of sandy soil during excavation work, M.S. thesis, Dept. Civil Eng., Tokyo Institute of Technology, 2000, pp.1-100.
- [14] Ministry of Land Infrastructure, Transport and Tourism, Land Infrastructure, National Geotechnical Information Search Site, Available from: <http://www.kunijiban.pwri.go.jp/jp/index.html>, Accessed on: Dec. 18, 2022.
- [15] Geospatial Information Authority of Japan, Fundamental Geospatial Data site, Available from: <https://www.gsi.go.jp/kiban/>, Accessed on: Dec. 18, 2022.
- [16] Michida, Y., Development of Marine Environmental Information, Environmental research quarterly, Vol. 70, 1988, pp. 116-123.
- [17] Geospatial Information Authority of Japan, Ministry of Land, Infrastructure, Transport and Tourism Notification No. 9, 2002. Available from: <https://www.gsi.go.jp/LAW/heimencho.html>, Accessed on: Dec. 18, 2022.
- [18] Lorenson, W.E. and Cline, H.E., Marching Cubes: A High Resolution 3D Surface Construction Algorithm, Computer Graphics, Vol.21, No.4, 1987, pp. 163-170.

- [19] C. Geuzaine and J.-F. Remacle, Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities, *International Journal for Numerical Methods in Engineering*, Vol. 79, 2009, pp. 1309-1331.

Copyright © Int. J. of GEOMATE All rights reserved,
including making copies, unless permission is obtained
from the copyright proprietors.
