ASIAN JOURNAL OF CIVIL ENGINEERING (BUILDING AND HOUSING) VOL. 2, NO. 3, (2001)
PAGES 235-257

FINITE ELEMENT MODELS FOR GEOTECHNICAL ENGINEERING PROBLEMS

*Udo F. Meissner, Jochen Ruben and Inke Terlindent
Institute for Numerical Methods and Informatics in Civil Engineering,
Darmstadt University of Technology, Darmstadt, Germany

ABSTRACT

A system is introduced in this paper which allows a time-dependent management of the individual stages of the construction by a dynamic model in consideration of the ground properties, the geotechnical construction elements and the construction processes. An automatic mesh generation enables the generation of hexahedron meshes for the finite element representation.

Methods are described and modeling techniques used for numerical investigations with particular emphasis on a multi-component-continua theory for partially saturated porous media and a finite element model for inelastic soils.

By means of this complex theory the paper shows that efficient numerical investigations are done with C++. A calculation method based on object oriented modeling techniques is introduced.

The development of a visualization application is discussed. Solution has been developed to overcome the bottleneck of the post-process visualization. In order to allow visualization of data simultaneously to the parallel calculation process several aspects have to be taken into account: an efficient data structure, fast mechanisms for communication between the calculation processes and the visualization application, algorithms to handle large data sets and methods for data reduction.

Keywords: geotechnics, finite elements, modeling, simulation, soil, visualization, multi-component-continua.

1. INTRODUCTION

The comprehensive modeling of soil-structure interaction requires three-dimensional dynamic design systems, that capture the changes which occur on the geotechnical components during the construction progress. The powerful generation and administration of the complex time-dependent geotechnical model need holistic computational tools for the management, the visualization and the proof of the engineering system. The system introduced allows a time-dependent management of the individual stages of the construction by a dynamic model in

* E-mail address of the author: sekretariat@iib.tu-darmstadt.de

consideration of the ground properties, the geotechnical construction elements and the construction processes. An automatic mesh generation enables the generation of hexahedron meshes for the finite element representation.

Within the framework of a macroscopic formulation, porous media models can be described using of the theory of multi-component continua. In this field the phenomena of seepage flow, drainage with filter layers, liquefaction and consolidation occur under excitation by stationary and time dependent external forces. This paper describes the methods and modeling techniques used for numerical investigations with particular emphasis on a multi-component-continua theory for partially saturated porous media and a finite element model for inelastic soils.
By means of this complex theory the paper shows that efficient numerical investigations are done with C++. We introduce a calculation method based on object oriented modeling techniques.

The use of parallel computers has significantly increased the size and complexity of problems to be solved. The result of a finite element calculation is a huge amount of numerical data that has to be judged and evaluated. The most efficient way is to visualize the data. Subsequently, the development of a visualization application will be discussed. Our solution has been developed to overcome the bottleneck of the post-process visualization. In order to allow visualization of data simultaneously to the parallel calculation process several aspects have to be taken into account: an efficient data structure, fast mechanisms for communication between the calculation processes and the visualization application, algorithms to handle large data sets and methods for data reduction.

2. MODELING OF SOIL-STRUCTURE INTERACTION

Modern large scale buildings are erected on rather complex foundations. Deep excavations in densely built-up areas require retaining walls and other geotechnical construction elements. The planning and design of the pit and the geotechnical engineering is normally performed using different software tools. The stability of the construction is investigated in necessary spots at different times independently from each other. An integrated design is not always guaranteed. The following approach enables a holistic computer aided modeling of the ground as well as the geotechnical construction considering the construction time schedule, reference [4].

1.1. Model management for dynamic geotechnical engineering problems

The basic problem connected with geotechnical engineering is the multidimensionality n pace and time, references [3, 12]. Planning work is carried out in several individual phases which differ in their degree of complexity and generally rely on information obtained from preceding phases: Phase 1: Prior clarification of geotechnical conditions for a project (e.g. site investigation etc.), Phase 2: Foundation soil investigations for the multi-stage project (e.g. determination of soil parameters etc.), Phase 3: Detailed investigations for the concretised project (e.g. design, dimensioning and checking of construction elements, etc.), Phase 4: Additional clarifications and inspections during construction work (e.g. control of the actual foundation soil based on the excavated material).

The planning process constitutes a highly time-dependent system. The order in which the individual planning phases are carried out prescribes a specific sequence f ngineering odels.

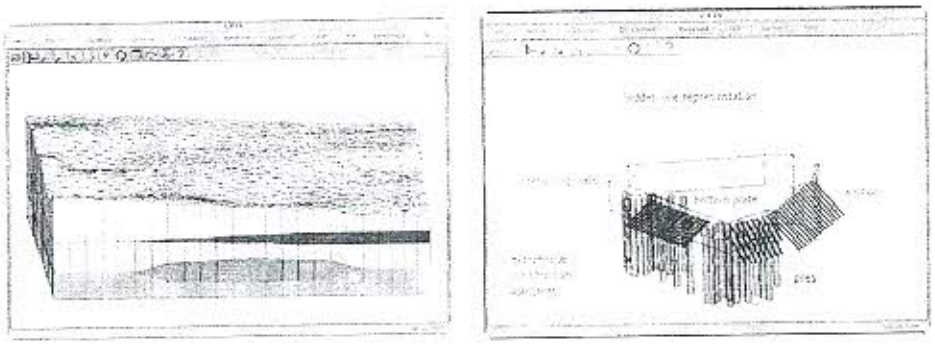
In Phase 1 the preliminary site investigation is conducted on the basis of two-dimensional layout plans and geological information. Once the site investigation has been completed, further information concerning e.g. soil composition, neighboring structures etc. are added in Phase 2. By combining the information obtained from Phases 1 and 2, a new level of detail is attained for the purpose of dimensioning the construction elements, which may then be represented in layout plans, sectional drawings and 3-D models. Once the dimensioning stage has been reached (Phase 3) decisions must be made concerning the geotechnical construction based on the available planning information. The necessary starting parameters (e.g. width, materials, number of structural elements etc.) must be specified for this purpose. The parameters obtained during the preliminary design stage subsequently serve as input values for the final dimensioning and analysis of the geotechnical construction. In order to carry out the necessary structural stability and load safety verifications for the selected construction, suitable models are developed in the first instance. These models include all relevant information and basic planning data. For this reason, both simple one- and two-dimensional models as well as complex three-dimensional models are applied. For verification purposes it is necessary to set up and analyze models for all critical construction stages. Only by this means it is possible to guarantee the structural stability of the construction throughout the entire construction period.

For the simulation of different scenarios of the construction state we developed and implemented the research prototype of the GeoTechnical Information System (GTIS). Here are some requirements resulting from the described scenario in geotechnical practice.

- The ground model requires a time-dependent structure to represent all the critical states. This is only possible by a software design using a dynamic data structure.
- The dynamic data structure has to be extended by a special access mechanism to reduce the access complexity within the ground model because of the complexity of the ground model in time and space. Therefore we use an octree-data-structure.
- A direct relationship between the ground model objects and the geotechnical construction elements is necessary.
- A construction manager is required for the handling of the different critical construction states.
- Because of the three-dimensional complexity of the ground and geotechnical construction, a finite element computation is needed for the static proofing of all building states.

The approach for an integrated design is shown in [3], [12] and [9]. A central three-dimensional time-dependent ground model was developed using the methods of the object-oriented modeling. The ground model consists of the following components:

Ground Model: The ground model includes all data of the building ground such as the boundary of the area and information about the ground materials from bored drills (Figure 1 a)). Undisturbed foundation soil is assumed as a starting point for the ground model. By integrating individual changes in soil geology and soil properties (e.g. due to excavation work or soil improvement) during construction, subsequent construction states are generated and stored in the ground model for retrieval as required at a later stage.



(a) (b)

Figure 1 a) Ground Model b) Geotechnical Construction Model

Geotechnical Construction Model: The geotechnical construction model contains the geotechnical construction elements designed by the civil engineer. This permits individual construction elements (piles, foundations, retraining walls) to be generated in parameterized form from construction component catalogues and to be combined as required (Figure 1 b)).

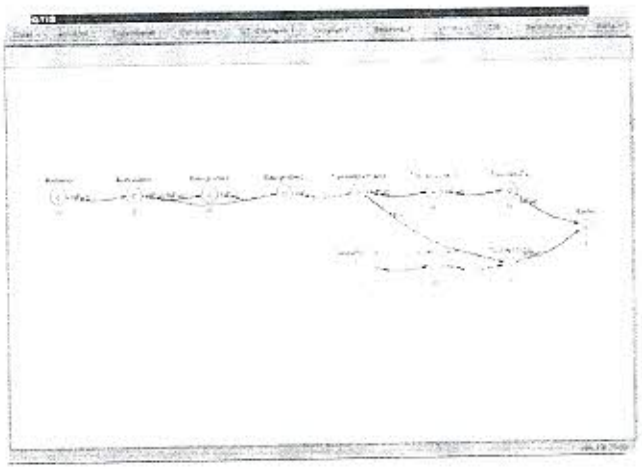


Figure 2. Construction Sequence Model

Construction Sequence Model: The construction sequence model permits the definition of the temporal sequence of construction phases. Network planning technology is applied for this aim. Critical and relevant construction states and processes are specified in the network (Figure 2). By this means, it is possible to systematically describe and manage the construction sequence. With the aid of construction sequence control during each construction stage, a valid model comprised of foundation soil, construction and excavated trench is created, which may be further processed for computation, dimensioning or construction purposes. As usual, processes and states, which are optionally coupled with geotechnical construction elements may be

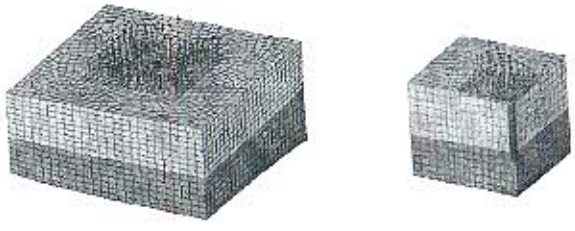
modeled in the form of network plans, which take account of time and resource factors. Optimization is possible by network plan variations and computations.

Finite Element Model: For all relevant states finite element models are generated. hese models contain all information for the calculation such as elements, nodes, loads, boundary conditions and contact areas.

2. FINITE ELEMENT MESH GENERATION

For the automatic generation of meshes representing ground and the construction, the following criteria are needful:

- The integral description of the ground is especially reasonable if the calculation is three-dimensional. The complex structure of group and construction elements should be managed in a CAD-system like in GTIS.
- Hexahedron or tetrahedron elements are possible geometric elements for three-dimensions. In the finite element program that is used, a 20-node hexahedron element is implemented for the simulation of the ground.
- For the convergence of the calculation mesh refinements must be possible in critical areas of the continuum. To discover these critical areas an error estimator is to be used [10].
- For the simulation by the finite element method boundary conditions and loads are required. These are assigned automatically to the nodes and elements of the discretised model when it is derived from GTIS.



(a) (b)

Figure 3. Finite Element Mesh a) complete model b) symmetric cut model

Due to the complexity of the ground and the construction unstructured hexahedron meshes are employed. These are generated with a projection method. This method consists of the following three steps:

1- Projection: All geometrical information of the geotechnical construction elements are projected on a horizontal layer.

2- 2D-Mesh: Using the information of the horizontal layer, a two-dimensional mesh is produced with quadrangle elements.

3- Hexahedron Elements: Considering additional information in depth of the construction elements and the thickness of the ground layers, soils of hexahedron elements are generated for each quadrangle.

3. MESH REFINEMENT

Local mesh refinements are necessary in order to obtain convergence for the finite element method. While using the projection technique, refinement of the initial 2D-mesh is not useful. The resulting mesh would contain badly shaped elements. Therefore the initial mesh is generated with a unique mesh density and the refinements are realized in a subsequent step.

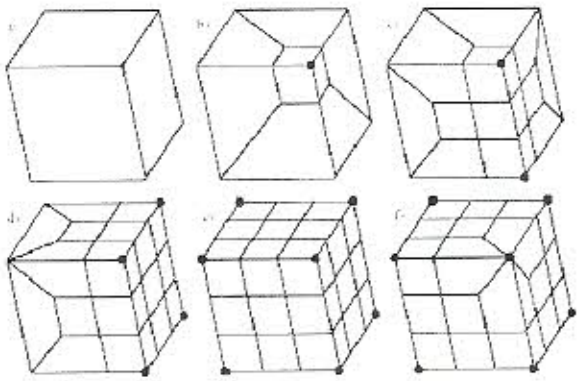


Figure 4. Refinement of Hexahedron Elements

For the refinement of hexahedron elements edges of an element have to be split into three parts. For the refinement of arbitrary regions 22 interface-meshes are necessary. Some of these interface meshes cannot be generated, like the mesh shown in Figure 4 f). To refine convex regions only 5 of these 22 interface-meshes are necessary shown in Figure 4 a) - e). The refinement of non-convex regions is possible by enlarging the refinement region to obtain a convex region. Recursive refinements are possible using this technique in order to achieve a higher density.

4. FINITE ELEMENT SIMULATION

For most mechanics problems in engineering practice the material is treated as a unique continuum with no consideration of the inner structure of the material or of the fact that most materials consist of a compound of different parts. Especially for geotechnical problems, there are several materials which can hardly be described by the classical continuum theory. Mixtures as well as porous media belong to these materials. A porous medium is understood as a porous solid skeleton with its pores filled with fluid and/or gas. The character of these materials is, on one hand, that they consist of more than one material, on the other hand, there exist relative motions and forces between the parts of the body. Based on the theory of mixtures combined with the volume fraction concept, we will explore a general approach to porous media linear-elasticity for a binary system consisting of a linear-elastic solid matrix

saturated by an incompressible viscous pore liquid. In the framework of the finite element theory the concept generally leads to a geometrical and physical non-linear problem.

In order to obtain a possibly simple and practical theory several simplifications will be introduced. Considering the limitations of the problem the fundamental equations of thermodynamics lead to a set of constitutive equations for linear elastic fluid-filled porous solids which are suitable to describe linear-elastic continua. Transformation toward a weak formulation yields a matrix scheme of differential equations for the incompressible model under study. The matrix scheme is solved by a modified Newmark method. The solution procedures are implemented with object-oriented programming techniques to allow extension and flexible development. [13]

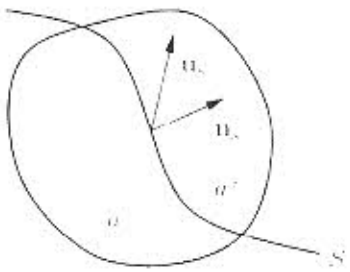


Figure 5. Interface between material parts (u_s is the surface n_s the unit normal vector and $[a] = a^+ - a$ the jump of a).

5. MIXTURE THEORY

5.1.1 Microscopic equations

The underlying multi-component continuum C is microscopically regarded as a combination of material parts C_α (e. g. solid, fluid, gas), which are separated by the surfaces $S_{\alpha\beta}$. The expansion of these parts is large compared to the molecular dimension. The physical variables (e. g. density, velocity) are sufficiently smooth in the sub-areas and have jumps on the interfaces (Figure 5).

The basic equations of the theory of continuum are valid in their local form on every part of the body:

Conservation of mass

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \mathbf{v}) = 0 \tag{1}$$

Balance of momentum

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \operatorname{div}(\rho \mathbf{v}\mathbf{v} - \mathbf{T}) = \rho \mathbf{f} \tag{2}$$

5.1.2. Averaging method

The transition from microscopic to macroscopic equations is made by an averaging method. Let $\omega(\xi, \tau)$ be an integrable function, such that

$$\int\int\omega(\xi,\tau)d\xi d\tau = 1$$

We assume that $\omega(\xi, \tau)$ has compact support. This assumption is automatically satisfied if C is a bounded domain, because then $\omega(\xi, \tau)$ will be assumed to be zero outside of the given domain and all the integrals will live now on the domain C.

Let $a(\xi, \tau)$ be a microscopic quantity, which can be scalar, vector or tensor function. By definition the macroscopic quantity $<a>(\xi, \tau)$, associated to the microscopic quantity $a(\xi, \tau)$, is

$$< a > (x,t) := \int\int a(x-\xi, t-\tau)\omega(\xi,\tau)d\xi d\tau \tag{3}$$

For the averages of a scalar a, a vector v and a tensor T the following equations can be proven:

$$\text{grad} < a >=< \text{grad}\, a + [a]\, n_s \delta_s > \tag{4}$$

$$\frac{\partial}{\partial t} < a >=< \frac{\partial a}{\partial t} - [a]\, \mathbf{u}_s . \mathbf{n}_s \delta_s > \tag{5}$$

$$\text{div} < \mathbf{v} >= \text{div}\, \mathbf{v} + [v] . \mathbf{n}_s \delta_s > \tag{6}$$

$$\frac{\partial}{\partial t} < \mathbf{v} >= \frac{\partial v}{\partial t} - [v]\, \mathbf{u}_s . \mathbf{n}_s \delta_s > \tag{7}$$

$$\text{div} < \mathbf{T} >=< \text{div}\, \mathbf{T} + [\mathbf{T}] . \mathbf{n}_s . \delta_s >, \tag{8}$$

where u_s is the velocity of the interface $S_{\alpha\beta}$, [a] the jump of a on the surface $S_{\alpha\beta}$ of separation between the constituents α and β, n_s the normal unit vector at $S_{\alpha\beta}$ oriented to the constituent α and δ_s, the Dirac's distribution in the 3-dimensional space with respect to $S_{\alpha\beta}$. For a surface discontinuity $S_1 <a, \delta_s>$ equals the area integral

$$< a\delta_s >:= \int\int_S a(x-\xi, t-\tau)\omega(\xi,\tau)d\xi, d\tau$$

5.1.3. Conservation of mass

From Eqn. (1) we get

$$< x_\alpha \left\{ \frac{\partial \rho}{\partial t} + \text{div}(\rho v) \right\}$$

Through transformation of Eqn. (5) and (6)

$$< \frac{\partial a}{\partial t} >= \frac{\partial}{\partial t} < a > + < [a]\mathbf{u}_s.\mathbf{n}_s\delta_s >$$
$$< \operatorname{div} \mathbf{v} >= \operatorname{div} < \mathbf{v} > - < [\mathbf{v}].\mathbf{n}_s\delta_s >$$

one gets

$$\frac{\partial}{\partial t} < x_\alpha \rho > + \operatorname{div} < x_\alpha \rho \mathbf{v} >= \hat{c}_\alpha$$

where c_α is the gain of mass due to chemical reactions.
The total mass has to be conserved

$$\sum_{\alpha=1}^{n} c_\alpha = 0$$

According to Eqn. 3 the definition to the macroscopic density and the velocity are given by

$$n_\alpha \rho_\alpha =< x_\alpha \rho > \qquad n_\alpha \rho_\alpha v_\alpha =< x_\alpha \rho \mathbf{v} >$$

and we get the macroscopic balance of mass for the component α

$$\frac{D}{Dt}(n_\alpha \rho_\alpha) = c_\alpha, \tag{9}$$

where n_α is the volume fraction and ρ_α the bulk density function of the component α.

5.1.4. Balance of momentum
With Eqn. (2) we get

$$< x_\alpha \left\{ \frac{\partial(\rho \mathbf{v})}{\partial t} + \operatorname{div}(\rho \mathbf{v}\mathbf{v} - \mathbf{T}) \right\} >=< x_\alpha \rho \mathbf{f} >$$

Transforming Eqn. (7) and (8)

$$< \frac{\partial}{\partial t} >= \frac{\partial}{\partial t} < \mathbf{v} > + < [\mathbf{v}\mathbf{u}_s]\mathbf{n}_s\delta_s >$$
$$< \operatorname{div} \mathbf{T} >= \operatorname{div} < \mathbf{T} > - < [\mathbf{T}]\mathbf{n}_s\delta_s >$$

we get the following equation for the balance of momentum

$$\begin{aligned} &\frac{\partial}{\partial t} < x_\alpha \rho \mathbf{v} > + \operatorname{div} < n_\alpha \rho_\alpha \mathbf{v}_\alpha \mathbf{v}_\alpha + n_\alpha \mathbf{M}_\alpha - x_\alpha \mathbf{T} > \\ &=< x_\alpha \rho \mathbf{f} > + < [x_\alpha \rho \mathbf{v}(\mathbf{v} - \mathbf{u}_s) - \mathbf{T}]\mathbf{n}_s\delta_s > \end{aligned} \tag{10}$$

with $n_\alpha M_\alpha = < x_\alpha \rho (v - v_\alpha)(v - v_\alpha) > n$ Egn.10 we implicitly assumed that the external body force f is uniform through all constituents, like the gravity. Eqn. 10 is the macroscopic equation of balance of linear momentum for the constituent a, using the definition introduced above it takes the form

$$\frac{D}{Dt}(n_\alpha \rho_\alpha \upsilon_\alpha) - \mathrm{div}(n_\alpha T_\alpha^c) = n_\alpha \rho_\alpha f_\alpha + \hat{t}_\alpha \tag{11}$$

with $T_\alpha^c = T_\alpha - M_\alpha$ and chemical reactions neglected.

5.2. Two-phase continuum

5.2.1. *Extra stress, extra interaction force, material law*

If we define the pressure p in the sense of a Lagrange multiplier as an indetermined hydrostatic pressure of the mixture, we get the following equation for the partial stress of solid and fluid

$$\mathbf{T}_\alpha = -n_\alpha p \mathbf{I} + \mathbf{T}_E^c \tag{12}$$

The interaction force between solid and fluid is of the form

$$\hat{\mathbf{p}}^F = \frac{\hat{t}_F}{n_F} = p\,\mathrm{grad} n_F + \hat{\mathbf{p}}_E^F \tag{13}$$

From the mixture balance equation, the momentum production terms are constrained by

$$\hat{\mathbf{p}}^F + \hat{\mathbf{p}}^S = 0 \tag{14}$$

For the extra interaction force the following equation applies

$$\hat{p}_E^F = -b(\dot{\mathbf{w}} - \dot{\mathbf{u}}) \tag{15}$$

where w is the displacement of the fluid, u the displacement of the solid and b the Darcy permeability coefficient for isotropic conditions.

5.2.2. Integration of the equation of continuity

Dividing Eqn. (11) by n_α and integrating we obtain for the solid and fluid phase

$$\int_B \left(\frac{D(\rho_S \upsilon_S}{Dt} - \mathrm{div}\mathbf{T}_S \right) dV = \int_B \left(\rho_S f_S + \hat{\mathbf{p}}^S \right) dV \tag{16}$$

$$\int_B \left(\frac{D(\rho_F \upsilon_F}{Dt} - \mathrm{div}\mathbf{T}_F \right) dV = \int_B \left(\rho_F f_F + \hat{\mathbf{p}}^F \right) dV \tag{17}$$

with

$$\frac{D\rho_\alpha}{\rho_\alpha} = \frac{1}{K} Dp,$$

where K is the coefficient of compressibility of fluid [2], and Eqn. (9) we obtain for pressure p

$$p = -Q\mathrm{div}\left(n_F \mathbf{w} + (1 - n_F)\mathbf{u}\right) \tag{18}$$

Q tends to infinity for most soils and denotes the storage due to compressibility of the solid grains and of the fluid. We can write

$$\frac{1}{Q} = \frac{n_F}{K_f} + \frac{n_S}{K_S},$$

where K_S and K_f are the bulk moduli of the fluid and solid material respectively. The solid extra stress T_E^S for an elastic material is governed by

$$\mathbf{T}_E^S = \lambda \mathrm{div}\, \mathbf{u}\mathbf{I} + \mu\left(\mathrm{grad}\, \mathbf{u} + \mathrm{grad}\, \mathbf{u}^T\right) \tag{19}$$

For simplicity, the extra fluid stress T_E^S is neglected.

Inserting Eqn. 12, 13, 14, 18 and 19 in Eqn. 16 and 17 we finally obtain for the solid

$$\begin{aligned} &\int_B \rho_S \ddot{\mathbf{u}} dV - \int_B b(\dot{\mathbf{w}} - \dot{\mathbf{u}}) dV \\ &\int_B \left(\lambda + Q(1 - n_F)^2\right) \mathrm{grad}\, \mathrm{div}\, \mathbf{u} dV \\ &\int_B Q(1 - n_F) n_F \mathrm{grad}\, \mathrm{div}\, \mathbf{w} dV \\ &= \int_B \rho_S f dV \end{aligned} \tag{20}$$

and the fluid

$$\begin{aligned} &\int_B \rho_F \ddot{\mathbf{w}} dV - \int_B b(\dot{\mathbf{w}} - \dot{\mathbf{u}}) dV \\ &\int_B n_F (1 - n_F) Q \mathrm{grad}\, \mathrm{div}\, \mathbf{u} dV \\ &\int_B n_F^2 Q \mathrm{grad}\, \mathrm{div}\, \mathbf{w} dV \\ &= \int_B \rho_F f dV \end{aligned} \tag{21}$$

5.3. Global fem system

For the FEM-approximation we chose the following shape functions

$$\delta \mathbf{u} = \sum_K N_K(x)\delta \mathbf{u}^K(t)$$
$$\delta \mathbf{w} = \sum_K N_K(x)\delta \mathbf{w}^K(t) \tag{22}$$

and the FEM-formulation of Eqn. (20) and (21) lead to the matrix formulation of the system

$$\begin{bmatrix} M_s & 0 \\ 0 & M_f \end{bmatrix} \cdot \begin{bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{w}} \end{bmatrix} + \begin{bmatrix} C & -C \\ -C & C \end{bmatrix} \cdot \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{w}} \end{bmatrix} + \begin{bmatrix} K_s & K_{sf} \\ K_{sf} & K_f \end{bmatrix} \cdot \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} r_s \\ r_f \end{bmatrix}, \tag{23}$$

where M_s, M_f are the solid and fluid mass matrices, C is the viscous damping matrix, K_S, K_f, K_{sf} are the solid and fluid stiffness matrices and r_s, r_f are the nodal forces of the solid and fluid.

5.4. Object oriented solver design

Finite element methods are mathematical procedures usually involving a huge hierarchy of functions in procedural languages such as FORTRAN or C. The object-oriented approach allows for the solution of a number of design drawbacks of procedural languages, namely the lack of information encapsulation, of generalization and specialization, of reusability and extendability.

5.4.1. Concept of a matrix-class

In the project we used a matrix class, which shows a very high performance. We implemented a template class and used the temporary base class idiom. The idea is to introduce temporary matrices as independent objects in a new class TMatrix. The class has the same member data as the class Matrix and Matrix can be derived from T Matrix.
The difference between the two classes lies in the behavior of the member functions. Temporary matrices are duplicated very fast by manipulating the pointers to the data blocks, whereas matrices are copied component-wise.

The result of, for example the operator+ is not of type Matrix, but TMatrix. Therefore, when the result is handed back, the copy constructor of TMatrix is executed, which means that only pointers are manipulated. Only Matrix::operator= transforms the TMatrix in a Matrix by modifying the pointer to the data block. Furthermore some optimizations can be done, for example in the matrix-matrix multiplication. Note that the innermost loop in a matrix-matrix-multiplication computes a scalar product of vectors which can be optimized by special means. The obtained matrix-class shows a very good performance in comparison to other libraries (e. g. BLAS).

5.4.2. Modified newmark method

As time integration scheme to solve Eqn. (23) we use a modified Newmark method [6]. This one-step algorithm is unconditionally stable and shows numerical damping properties

that can be continuously controlled.
This algorithm is embedded in a system of solver for differential equations. We extended the object-oriented class library ODE++ [11] by a hierarchy for direct solvers. The structure of the class library also allows to add other solving algorithms. The graph in Figure 6 shows the actual class hierarchy.

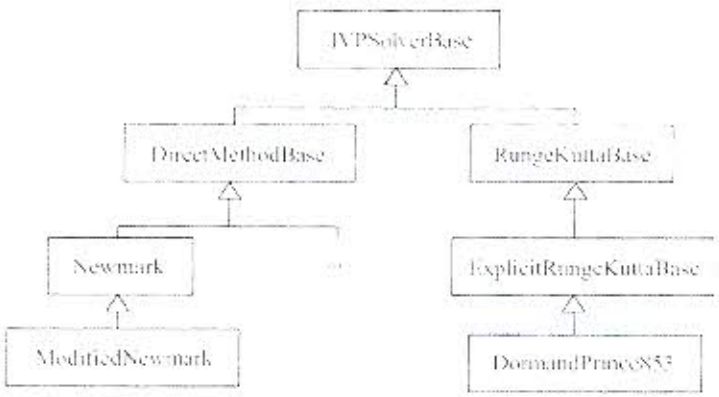


Figure 6. Class Hierarchy

The realization of the classes above reflects the three main aims:

- all functions that allow the user to control the integration process are public member functions of the class IVPSolverBase. This results in a consistent user interface independent from the used integration method.
- for every parameter of a special method there exists a member function that allows to change the value of the parameter. By default, every parameter is set to a reasonable value. The resulting solver can be successfully applied to a great variety of problems. Only a single function call is needed to change the value of a parameter.
- an analysis of existing solvers showed the possibility to divide the integration process in several subtasks. Each subtask is realized as a member function of IVPSolverBase. If a subtask is method-dependent the member function belonging to it is declared pure virtual, if the subtask is method-independent the function is declared virtual and IVPSolverBase contains a definition suitable for all methods.

5.3. Examples

As an example we chose a two-dimensional consolidation problem with a linear-elastic solid and viscous pore liquid. The geometry and boundary conditions of this example are shown in Figure 7. The example represents a saturated soil block, where the load is applied instantaneously at t = 0s. Figure 8 displays the numerical results for the two-component material.

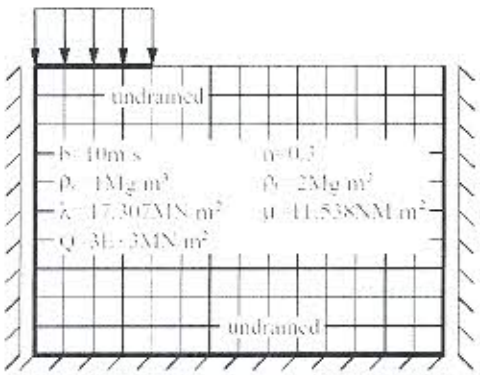
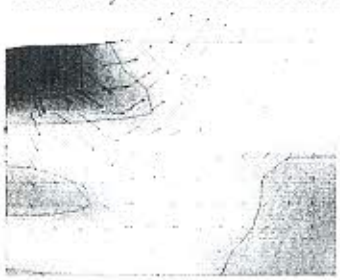
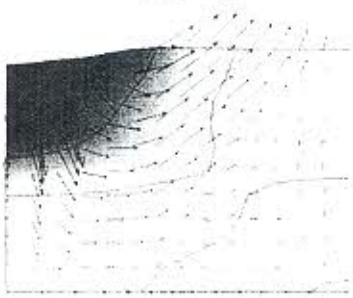


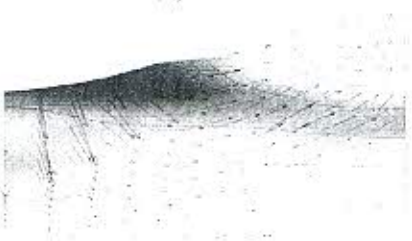
Figure 7. Geometry of the consolidation example



(a) t = 1s



(b) t = 2s



(c) t = 4s

Figure 8. Deformation, fluid velocity and pore pressure at different time steps.

6. VISUALIZATION

The calculation produces results that have to be interpreted and evaluated by the user. The employment of parallel computers has resulted in a considerable increase in complexity and size of solved problems. Although the procedure of parallel computing is more complicated because of the necessity for interprocessor communication, it is used for many large problems where sequential computers fail due to memory and calculation time requirements.

Apart from the idealization of the problem itself, the interpretation of results is the most important step of a practical analysis [1]. Special focus must be given to the visualization of result data produced by the calculation. The model with the associated results needs to be transformed into an intuitively comprehensible and easily interpretable form that enables the engineer to obtain a fast overview of the entire calculation.

Issues of finite element calculation and visualization of the results, in practice are often considered as separate and independent aspects. The usual approach is to store all data connected to the simulation and to visualize them in a second step. This is especially critical when time dependent problems are regarded. The storage of all data requires a huge external memory capacity and - on the other hand - the visualization process can only be started when the calculation has completely terminated.

To avoid this resource bottleneck, data could be immediately processed and visualized as soon as they are available. Thus, the engineer has the possibility to control and judge the calculation at a very early point in time. For example, inconsistencies in the model or erroneous input parameters can be detected and the calculation can be restarted with modified parameters prior to the production and storage of a large amount of data. This approach to visualization is an indispensable tool for the development of programs and for the validation of new strategies in parallel computing that leads to shorter production times and a better quality of the results [S].

6.1. Visualization of finite element calculations

Visualization of scientific data is supported by a large variety of existing software solutions. Almost all major software providers offer an approach for data visualization. Apart from the large scope, the application of such standard software packages leads to disadvantages and limitations. [7]

- The user is in many cases limited to the use of proprietary data formats.
- Mechanisms for parallel communication are quite often not taken into consideration. Thus, communication between a parallel computer architecture and a visualization workstation, at the same time as the calculation process runs, is not contemplated.
- The capability of these systems to cover all aspects of visual requirements often comes along with high demands in resources such as main memory or disk space. This cannot be controlled or influenced by the user. Especially, a simple display of large models pushes these systems to their limits.
- In many cases there is no possibility to manipulate and extend the data structure of the visualization software. Data needs to be transformed from the data structure of the simulation process to the data structure of the visualization software without the option to offer the calculation data in the fashion of the visualization software and therefore making the copying

approaches have been implemented: one approach uses *sockets*, the other *CORBA*.
Socket-communication: Communication between the processors of the parallel FEM application and the visualization application takes place using *sockets*. Data is being exchanged using a predefined interface for communication. The structure of the whole system is given in Figure 10.

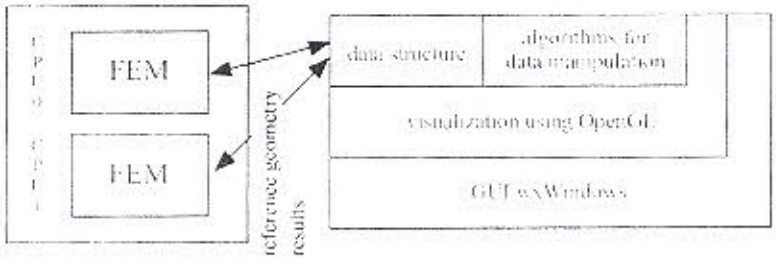


Figure 10. Connection of an FEM application to the visualization application.

CORBA-communication: A lot more of flexibility can be added to the system using mechanisms for communication and data exchange provided by the implementation of a CORBA architecture. Using the Object Request Broker (ORB) the data structure is transparently accessible in the net.

The structure of the implementation using CORBA is shown in Figure 11. The processes of the parallel FEM calculation program compile a data structure provided by a data server using the ORB. After the completion of a data set (more data can be added later) the collector is notified. The collector gathers information on what data is offered by which data server. This information can be evaluated by the visualization application. If for example a data server reports additional data to the collector, the visualization application realizes this when a connection to the collector is performed. Knowing that more data is available for visualization the associated data server can be contacted.
Finally, CORBA offers the possibility of programming language independence: the FEM program might be written in C and the visualization application in Java.

6.2.2. Data structure
In order to make the visualization application widely applicable special care has to be taken for an easy and fast connection to an existing calculation or simulation program. For that reason a particular data structure has been developed.

- The data structure can be designed to be efficient in terms of visualization requirements.
- Linking to an existing FE-program is significantly simplified because the data structure of that program can have any form. The connection to the data structure of the visualization application is implemented with an easy-to-use programming interface.

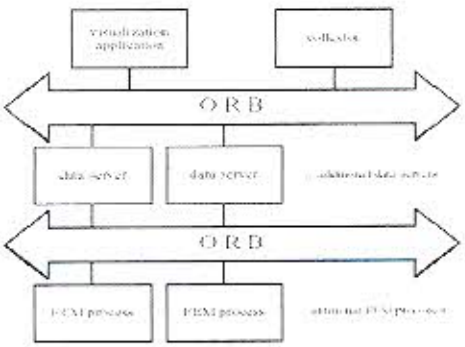


Figure 11. Data exchange using a CORBA architecture.

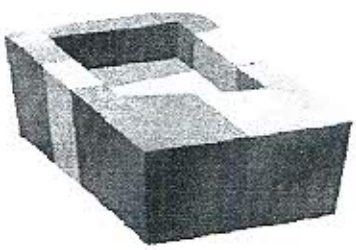


Figure 12. A geotechnical model is partitioned into 4 sub-models on 4 processors.

Requirements to the data structure: The initial situation is a finite element in terms of the finite element method being defined uniquely with its associated nodes. Surfaces and lines for the visualization have to be generated by the visualization application on demand and made available in an optimized form. Data can be structured in many data sets, e.g. sub-models distributed on several processors (Figure 12). This allows for a structured hierarchy of the model to be visualized.

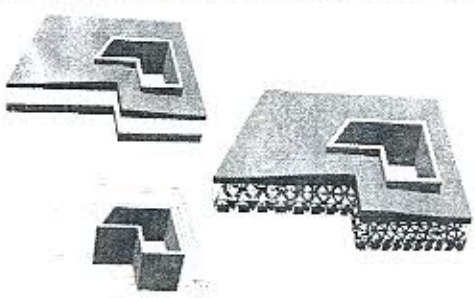


Figure 13. A model is structured using 4 data sets. They can be displayed independently.

Data sets initially hold the information of the finite element calculation: elements referring to nodes. Later, information on surfaces (volume mode) and lines (wire frame mode) can additionally be generated and stored in the data set. Figure 13 shows a model subdivided into several data sets to structure the model in a purposeful manner. Every layer of soil is

represented in a separate data set with an additional set for the geotechnical construction elements. All data sets can be displayed independently from each other.

A catalogue of finite elements has been implemented as prototype (Figure 14). Using the object-oriented programming the list can be easily expanded.

Implementation in C++: The object-oriented analysis according to Rumbaugh has lead to an object hierarchy that is given in Figure 15. Finite elements with their attributes and methods are represented in separate classes derived from a generalized base object *(=Shape)*. A data set is an aggregation of elements and nodes. For visualization purposes data sets can have additional information on one-, two- and three-dimensional rendering elements. Finally, one instance of the object *zzDataStructure* can consist of any number of data sets.

The administration of result values in the object hierarchy is performed by the class *zzValueSet* that can be associated with data sets. Separate classes are implemented for tensors of zero, first and second order.

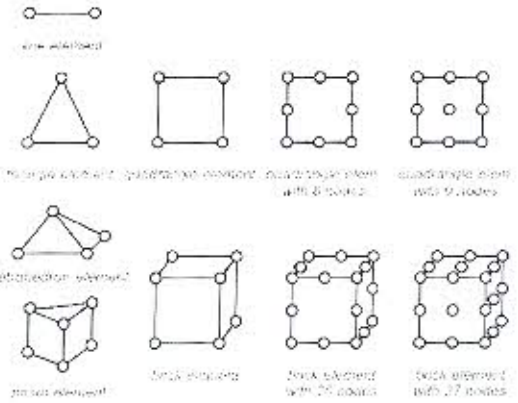


Figure 14. Prototypically implemented finite elements.

6.2.3. Display of the state of stress

Contrary to the display of tensors of zero or first order as results, the display of tensors of second order is not possible in a straightforward way. An example might be the state of stress in a finite element model given by the symmetric stress tensor:

$$S = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{pmatrix} \tag{24}$$

The display of such a matrix with 9 (6 different) values cannot be performed in a direct and intuitively perceptible way. A possibility to enable the evaluation of such numerical data is to display the associated principal tensor values. Mathematically, this leads to the solution of the eigenvalue problem for the stress tensor S.

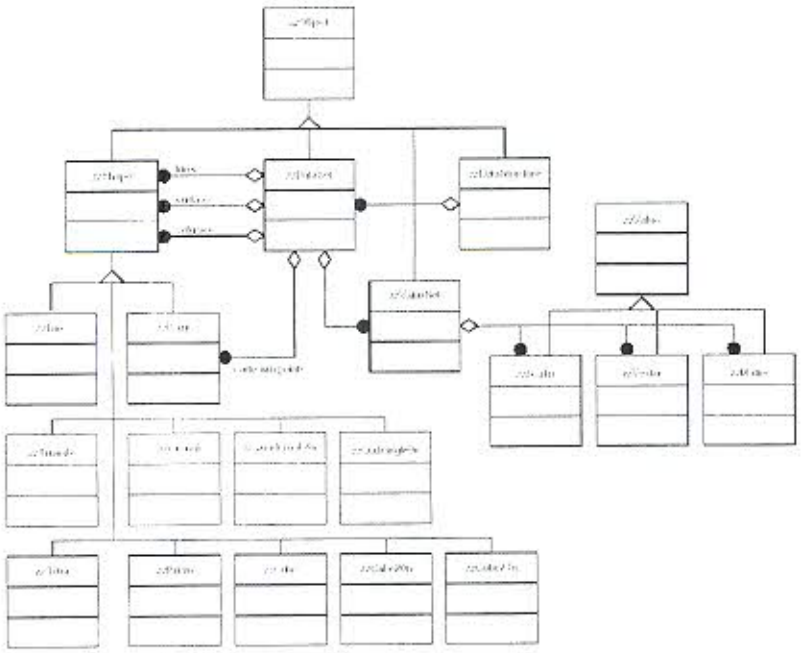


Figure 15. Object hierarchy of the data structure

The eigenvalues are values for A that fulfill the following equation:

$$\mathrm{Det}\,(S-? \; ? \; I) = 0 \tag{25}$$

For the display of the eigenvalues the prime axes need to be calculated as well. They give the direction for the display of the state of stress. The length is given by the absolute value of the eigenvalues. The sign of the eigenvalues can be displayed with two different colors.

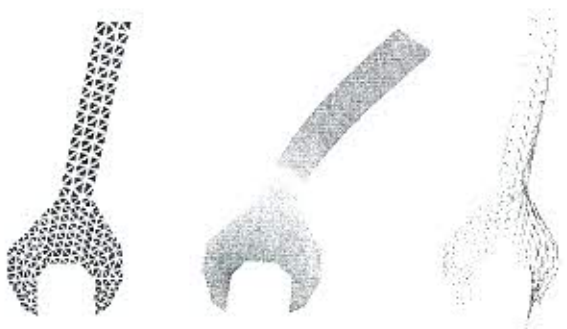


Figure 16. Example for the display of the state of stress in a system. Finite element mesh (left), displacements (exaggerated, middle) and state of stress (right)

An example for the visualization of a state of plane stress in a wrench is given in Figure 16. It shows a finite element system under a horizontal load in the upper corner, the system in its deformed state and the calculated stresses for the system.

6.2.4. Data reduction techniques

Large models often don't need to be displayed in full detail. To reduce the amount of data to be processed, data reduction might be reasonable. The user first wants to obtain a rough overview over his model. This can in most cases be done with a coarse display of the model. Consequently, interesting areas can be recognized and displayed later in full detail.

Data of parallel calculation processes are distributed on many processors urging he need to sort and send them for visualization purposes. The amount of communication can significantly be reduced when the model is transformed to a coarse model prior to data exchange. This is useful when the requirement is considered that the user obtains a fast and rough overview over his simulation. Several techniques can be performed on a model to obtain a coarse overview. Two of them are discussed here (Figure 17).

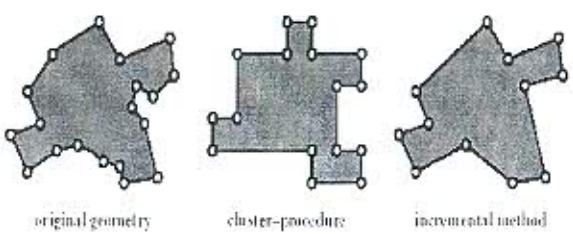


Figure 17. Data reduction techniques. The original geometry (left) is transformed into a coarse model using cluster procedures (middle) or incremental methods (right).

Cluster procedures transform the model into a rasterized version. In 2D a grid is imposed over the model and a pixel might be set or not in order to represent the original geometry as good as possible. The original nodes move to the junctions of the grid lines. In 3D the model will be approximated by "Lego"-bricks. Cluster procedures lead to fast but qualitatively poor results (Figure 18).

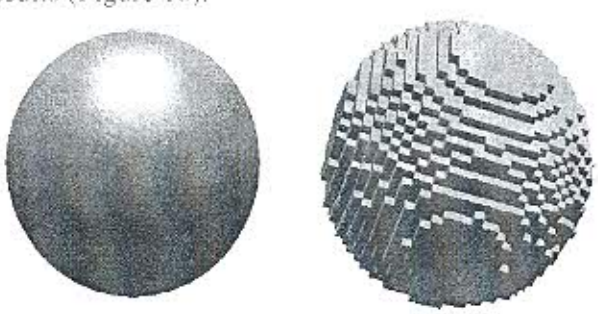


Figure 18. Cluster procedure applied to a model of a sphere reducing the 14.160 elements to a coarse model with 3.058 elements.

Incremental methods transform the original model iteratively into a coarse model by deleting nodes, edges or elements. This involves the evaluation of a quality criterion, e.g. for the local curvature. Operations that have the least effect on the quality of the whole model are performed first (e.g. the deletion of a node in a "flat" area). Incremental methods offer good results with high computational effort.

The techniques for data reduction can easily be performed in parallel. Figure 19 shows the incremental method performed on a model consisting of six sub-models on different processors. Each processor applies the incremental method to the associated sub-model independently resulting in a coarse model for each processor. It can be seen that data reduction is performed to a certain degree depending on the local curvature of the sub-models leading to reduction rates from 23 % to 98.2 % of the elements.

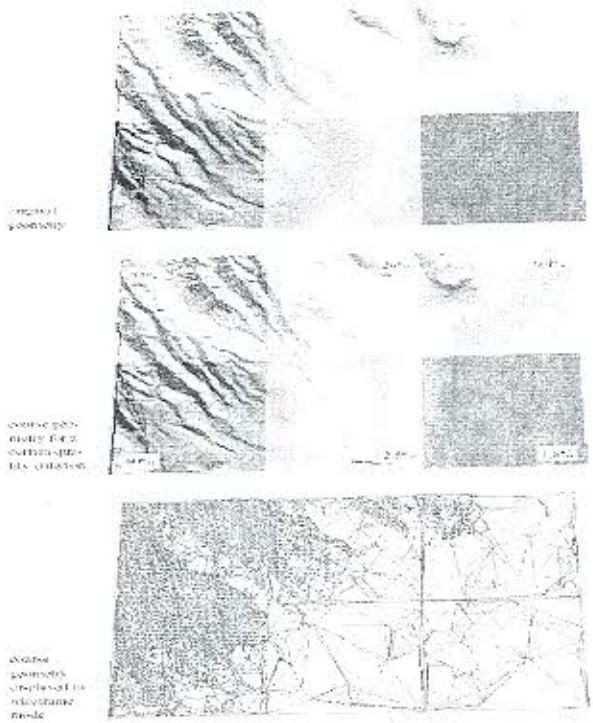


Figure 19. Data reduction using incremental methods. The geographical model is transformed into a coarse model considering the local curvature.

REFERENCES

1. K.-J. Bathe. *Finite Element Procedures.* Prentice Hall, 1996.
2. J. Bear and Y. Bachmat. *Introduction to Modeling of Transport Phenomena in Porous Media,* volume 4 of *Theory and applications of transport in porous media.* Kluwer Academic Publishers, 1991.

3. J. Diaz. Objektorientierte Modellierung geotechnischer Systeme. *Bericht 2198, Institut fur Numerische Methoden and Informatik im Bauwesen, TU-Darmstadt.* 1998.
4. J. Diaz, U. F. Meissner, and M. Burghardt. Time-Dependent Three-Dimensional Finite-Element Ground Model for Geotechnical Engineering Problems. *Proceedings of the 8th International Conference on Computing in Civil & Building Engineering,* pages 1458-1465, 2000.
5. T. Hartmann and U. F. Meissner. A Multi-Component Finite-Element Model for the Dynamic Analysis of Soil-Water-Air Structures. *Finite Element Analysis in Fluids,* pages 623-628, 1989.
6. H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics. *Earthquake Engineering and Structural Dynamics,* 5:283-292, 1977.
7. H. Katz and L. Lammer. On-Line-Visualisierung von Ergebnisdaten bei der parallelen Berechnung von Finite-Element-Problemen. *9. Forum Bauinformatik,* pages 141-147, 1997.
9. L. Lammer. Parallelisierung von Anwendungen der Finite-Elemente-Methode im Bauingenieurwesen. Institut fur Numerische Methoden and Informatik im Bauwesen, Berichte 1/97, Technische Hochschule Darmstadt, 1996.
10. U. F. Meissner, J. Diaz, I. Schonenborn, and R. Kruger. Object-Oriented Modelling of Three-Dimensional Hydro-Geotechnical Systems. In A.A. Aldama, J. Aparicio, C.A. Brebbia, I.Herrera W.G.Gra and. and G.F. Pinder, editors, *Computational Methods in Subsurface Flow and Transport Problems,* volume 1, pages 709-716. Computational Mechanics Publications, Southampton, Boston. 1996.
11. M. Milde. Ode++ a Class Library for Ordinary Differential Equations. Technical report, Institut fur Angewandte Mathematik, Friedrich-Schiller-Universitat Jena. *1998.*
12. Schonenborn. ProzeBorientierter Entwurf einer Boden-Tragwerk-Struktur. *Dissertation, Technische Universitat Darmstadt, Fachbereich Bauingenieurwesen, Bericht 1199 des Instituts fur Numerische Methoden and Informatik im Bauwesen, Technische Universitat Darmstadt, 1999.*
13. I. Terlinden and U. F. Meissner. Porous Media Modelling with Multi-Component Finite Elements and Object Oriented Techniques. *Computational Methods in Water Resources,* pages *3-8,* 2000.