Charles University, Prague, Czech Republic
Faculty of Mathematics and Physics

BACHELOR THESIS

Petr Zajíček

Simulace statiky budovy
Building statics simulation

Department of Software Engineering

Advisor: RNDr. David Kronus, Ph.D.

Study program: Computer Science, Programming

2009

I would like to take this chance and thank my parents for pushing me to work on the thesis. I would also like to thank COL Stephen J. Ressler, P.E., Ph.D. for answering my e-mail thus pointing me in the right direction and my advisor, RNDr. David Kronus, Ph.D. for accepting this project and all the help he gave.

I hereby certify that I wrote the thesis myself, using only the referenced sources. I agree with lending the thesis.

Prague, 29 Maj 2009 Petr Zajíček

Contents

Název práce: Simulace statiky budovy
Autor: Petr Zajíček
Katedra: Katedra softwarového inženýrství
Vedoucí bakalářské práce: RNDr. David Kronus, Ph.D.
e-mail vedoucího: david.kronus@matfyz.cz

Abstrakt: V předložené práci studujeme metody analýzy statiky budovy. Základem této práce je analýza statiky prutových konstrukcí. Na tomto základe, vybudujeme program schopný simulovat chování prostorové budovy zatížené kromě vlastní váhy také libovolnými silami působícími na jednotlivé její elementy. Budeme se snažit najít rovnováhu mezi přesností výpočtu a jeho rychlostí tak, aby výsledný program byl vhodný pro demonstraci obecného chování zatížených konstrukcí při skoro real-time rychlostech.

Klíčová slova: statika budovy, prutové konstrukce, zátěžové testování

Title: Simulation of building statics
Author: Petr Zajíček
Department: Department of Software Engineering
Supervisor: RNDr. David Kronus, Ph.D.
Supervisor's e-mail address: david.kronus@matfyz.cz

Abstract: In the presented work we study methods of building statics analysis. The basis of this work is the statics analysis of truss structures. On this basis we build a program capable of simulating a 3D building loaded with its own weight and arbitrary forces acting on its elements. We will try to find a compromise between the accuracy of the simulation and it's speed so that the resulting program will be fit for demonstrating the general behavior of loaded structures in near real-time.

Keywords: building statics, truss structures, stress testing

Chapter 1

Introduction

1.1 Preface

The aim of this project is to create an application to design simple structures and then analyze their statics to determine whether the design can withstand its weight and the forces applied to it. The application is partially inspired by the bridge builder games like Pontifex. It aims to extend their functionality to non-symmetric structures with more then one deck. The basis of the analysis is a statics analysis of a space truss. Structures consisting of floor and wall segments, beams and columns are broken down into a space truss that approximates their inner structure and weight distribution. After which the before mentioned analysis is conducted.

It is not the aim of this project to create a viable engineering software rather to create a software to demonstrate general effects of load on structures and patterns of weight distribution. Therefore the ability to perform deatialed analysis is sacrificed for the benefit of faster calculation. This means that no plain stress distributions are computed, but the analysis should be near real-time for properly designed structures and still give generally accurate results.

The statics analysis of the truss will be done using the Direct Stiffness Method, which is a form of the Finite Element Method, therefore the resulting application will have the poticial for further development.

1.2 Outline

This work consist of five chapters and an appendix, which describe the following: general overview of statics principles used in the work, local analysis vs global analysis, direct stiffness method, speeding up the DSM by using cholesky factorization instead of Gaussian elimination to solve linear equations, simulation structure building and modeling instructions. The appendix gives a short user's manual to the program, and a description of the sample scenes.

Chapter 2

Statics

This chapter will provide an overview of relevant information from the fields of Statics and Strength of Materials. It is meant only as an orientation to concepts later used.

2.1 General introduction to statics

Statics is a branch of mechanics which studies forces, loads applied to physical systems that are in static equilibrium. This means that in these systems according to Newton's first law their subsystems don't change position relative to each other. Statics studies what kind of reaction has the system to the applied loads. Statics is used for analyzing structures. One of the fields where statics is most used is architecture, where it is used amongst others to calculate the forces within the elements of a building. Methods from a related field Strength of Materials are then used to tell whether the elements of the building will fail under pressure. Since this project sets out to do exactly this the methods from these field will be used to accomplish it.

There are two general conditions of statics equilibrium the *Transitional Equilibrium* and the *Rotational Equilibrium.*[1]

The transitional equilibrium states that the forces applied to the structure must sum up to 0, so that the structure is not accelerated. This is described by the symbolic notation:

$$\sum \vec{F_{all}} = 0$$

Sometimes it is useful to consider forces broken down into their elements in 3 dimensions. In that case the transitional equilibrium can be written as:

$$\sum \vec{F_x} = 0\,, \sum \vec{F_y} = 0\,, \sum \vec{F_z} = 0$$

The rotational equilibrium states that the sum of Torque acting on the structure must be 0 for the structure to be in a rotational equilibrium. Symbolically written:

$$\sum \tau = 0\, or\ decomposed: \sum \tau_x = 0\,, \sum \tau_y = 0\,, \sum \tau_z = 0\,,$$

2.2 Trusses and truss structures

Trusses are common structural elements used in construction. They are composed of axial member which mean they carry only axial forces. Axial forces are forces acting along the axis of the members of the truss. This way trusses can be only under compression or tension, where compression develops from external forces trying to shorten the truss, while tension should elongate the truss. Axial members do not carry bending and shear forces. These forces are in later simulations approximated via including more axial members in a specific pattern. An ideal truss is composed of members which can be considered weightless or their weight can be evenly distributed to their endpoint. The members of a truss are connected in pinned joints and form a triangular structure. This is an idealization of trusses that does not precisely matches the reality, but it is used in statics because experience showed that it is satisfactory for determining most design needs. Trusses are used in architecture to support constructions. Most common application include bridges and roof structures.

There are several methods used to calculate the external (support) end internal forces in a truss. Most of these methods rely on applying the conditions of static equilibrium. Here a simple method will be presented to solve a simple statically determinate truss. Consider the truss shown in 2.2. The steps carried out while solving this truss would be:

Drawing free body diagram with forces broken down to their x and y components

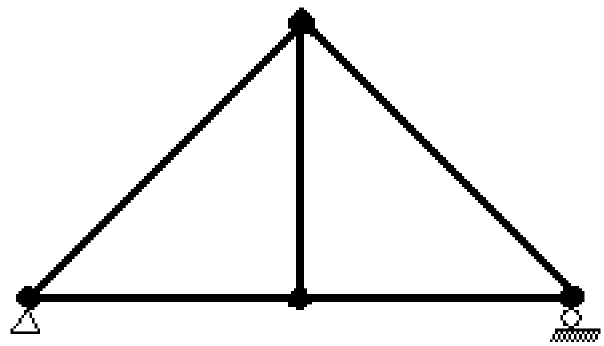

Figure 2.1: Statically determinate truss.

Applying the static equilibrium conditions to resolve the external support forces. In this case the transitional condition for x-direction forces and y-direction forces and rotational condition for one of the supports:

$$\sum \vec{F_x} = 0\,, \sum \vec{F_y} = 0\,, \sum \tau_{point} = 0$$

Since there are 3 unknowns (2 support forces at fixed support 1 support force at roller support) and 3 equations the external forces can be solved

Using the joint method[1] to solve for forces within the members. This method considers always one joint at a time applying the transitional conditions and solves for unknowns. When the unknown forces are solved they will become known in the member they are acting in, thus allowing for incremental solution in the neighboring joints. The stages of the joint method are shown int 2.2.

2.3 Hook's law

Simply determining the forces acting in the members is not enough to reach a conclusion about the safety of the structure. To arrive at the some conclusions we need to consider problems associated with the field of strength of materials[2]. First to introduce the concept of *Axial Stress*. Axial stress is the stress within a member. It is defined as the axial force divided by the cross section area: $\sigma = \frac{F}{A}$

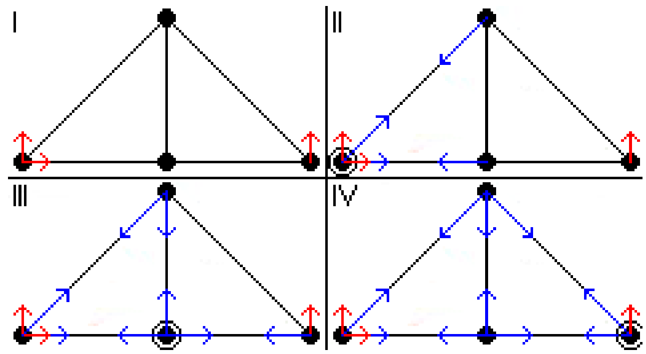

Figure 2.2: Stages of join method red are external forces while blue known internal forces. Encircled joints are the ones solved to arrive at the stage.

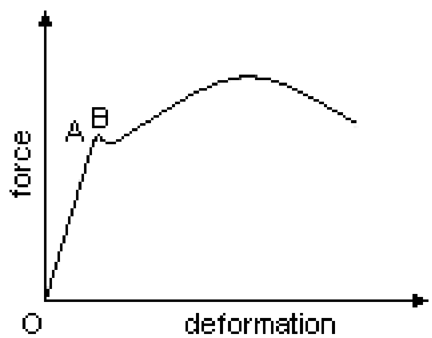

Figure 2.3: Relation force - deformation, A - Elastic limit, B - Linear limit

Forces applied to a member cause the member to deform. For a metal rod under tension the deformation along the length of the rod is most significant. If its studied under increasing tension force it shows that the deformation is linearly proportional to the applied force, until a certain point is reached as shown in 2.3.

The deformation in the *Linear Region* is a form of Hook's law that can be written as: $F = k\delta$, where k is a constant depending on the material also called *spring constant* and δ is the deformation. When enough force is applied the deformation becomes nonlinear, 2.3 shows this as the plastic region. The point where the elastic region ends is called *elastic limit* or the *proportional limit*. The two are actually not the same but for the purposes

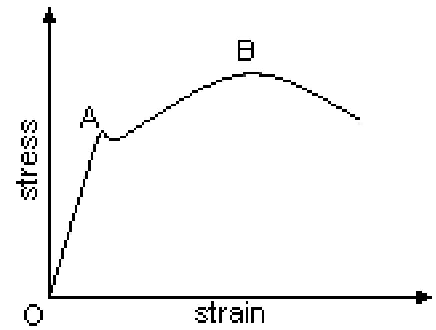

Figure 2.4: Relation stress - strain, A -Yield stress, B - Ultimate stress

of this analysis the difference is not significant. For the constructions to be safe it is required for all elements to stay within their elastic regions where when the load is lifted they return to their original form.

Rather then examining the *force-deformation relation* it is useful to examine the *stress-strain relation*. The concept of *axial stress* was already introduced, now for the *axial strain*. *Axial strain* is defined as the fractional change in length: $\epsilon = \frac{\delta}{L_0}$, where L_0 is the rest length of the rod.

The *stress-strain relation* shown in 2.3 is the same as the *force-deformation relation*, which means that in the *Linear Region* region the $\frac{\sigma}{\epsilon}$ ratio is a constant. This constant is the *Young's Modulus* also called *Elasticity* symbolized as **E**.

In the *stress-strain relation* the *elastic limit* is also called *Yield stress* which is the property of the material and can be found in physics tables. Along with the *Density* of the material the *Young's Modulus* and the *Yield stress* are going to be the important material properties for the analysis in this project. To recap:

$$Stress: \ \sigma = \frac{F}{A}$$

$$Strain: \ \epsilon = \frac{\delta}{L_0}$$

$$Hook's\ law: \ F = k\delta\,, \ E = \frac{\sigma}{\epsilon} \tag{2.1}$$

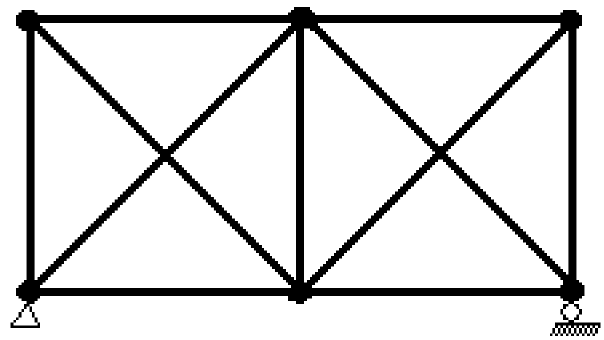

Figure 2.5: Statically indeterminate truss

$$Spring\ constant:\ k = \frac{EA}{L_0} \tag{2.2}$$

The last equation is only a useful combination of the first three.

2.4 Statically determinate and indeterminate structures

When solving a statics problem, like shown in section 2.2 it is important to notice that there are are situations when the conditions of static equilibrium are not enough. This happens when there are more unknowns that equations. Structures that can be solved only by applying the conditions of static equilibrium are called statically determinate structures[1]. Unfortunately more complex structures are almost always not like this. The structures that are not statically determinate like the structure shown on 2.4 are statically indeterminate structures.

To solve statically indeterminate structures one must take into accord the deformation of the structure caused by the forces acting on it. This is usually done by finding a way to express the deformation of the structure members relative to each other. After the relationship between the member deformation is established the deformation is expressed using Hook's law. This way we obtain new independent equations which help as solve the previously singular system.

One of the methods used to solve even statically indeterminate systems is the *Direct Stiffness Method* which is used int this project to do the structure analysis. For detailed description of this method see chapter 3.

Chapter 3

Local Analysis and Global Analysis

Even though the final version of this project uses the Direct Stiffness Method it was not the first choice when the development started, therefore the reasons for choosing this method will be briefly described in this chapter.

This project started out with the idea of analyzing the structure always only in the local context. The structure consisting of members connected in joints would be analyzed joint by joint. Members would be threated as linear springs and their elongation calculated according to Hook's law. For all joint a position would be calculated where the sum of forces acting on the joint would be zero (transitional equilibrium). For this the neighboring joints (joints on the other end of the member connecting in the current joint) are considered immovable, thus the situation in 3 arises.

A form of Hook's law is used:

$$-\vec{F} = \bar{K}\vec{u} \tag{3.1}$$

Form the 3 in local coordinate system$f_{\bar{y}} = 0$ and $u_{\bar{y}} = 0$. Combining this with the equation 2.1 and the equation 3.1 in which the F and u we break down into their components, we get the matrix form:

$$\begin{pmatrix} f_{\bar{x}} \\ f_{\bar{y}} \end{pmatrix} = \frac{EA}{L} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_{\bar{x}} \\ u_{\bar{y}} \end{pmatrix}$$

To arrive at a general solution the K matrix needs to be transformed into global coordinate system. The transformation is done by multiplying the

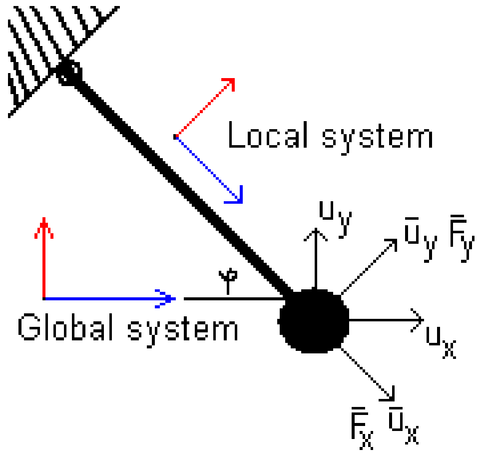

Figure 3.1: Statically indeterminate truss

matrix from left and from right whit transformation matrices which contain the direction of cosines of $\bar{x}$ and $\bar{y}$ with respect to x and y. The resulting matrix is:

$$K = \frac{EA}{L} \begin{pmatrix} c^2 & sc \\ sc & s^2 \end{pmatrix}$$

where $c = \cos\Phi$ and $s = \sin\Phi$.

For more members connected to joint we simply add up their effect on the joint.

$$-\sum \vec{F_i} = \sum K_i \vec{u_i}$$

According to the condition transitional equilibrium the sum of forces acting on a joint is 0, therefore $\sum \vec{F_i} = -\vec{F_{ext}}$, where $\vec{F_{ext}}$ is the external force applied to the joint. Since the joint can be only in one place at a time the displacements must be identical, therefore $\sum K_i \vec{u_i} = \sum (K_i)\vec{u}$. Combining the previous equations:

$$\vec{F_{ext}} = \sum (K_i)\vec{u},\ where\ K_i = \frac{E_i A_i}{L_i} \begin{pmatrix} \cos{\Phi_i}^2 & \sin\Phi_i \cos\Phi_i \\ \sin\Phi_i \cos\Phi_i & \sin{\Phi_i}^2 \end{pmatrix} \quad (3.2)$$

By solving equation 3.2 we obtain the displacement for one joint. For 3D structure the idea is the same only the number of components will be 3 and the transformation matrix is more complex.

The displacements are calculated for every joint in the structure, and then the joints are moved by the calculated displacement. This is one simulation step. After one step the whole structure will get closer to a global state of equilibrium but it will not yet be reached. Several iterations of simulation steps are needed for a structure to settle in an equilibrium. Unfortunately the number of steps needed to find the global equilibrium state depend on several factors and can not be determined beforehand. The main factor is the number of joints but the geometry of the structure and its rigidity are important factors too.

This method has some advantages, mainly its low memory requirements which are constant and its ability to handle dynamic forces(simply by changing the force while the simulation is running). The undetermined number of required simulation steps is however a major disadvantage. Tests done in the early development stages of this project showed that the time needed to arrive at an equilibrium was for structures containing several hundred joints unrealistically high. Careful consideration led to the conclusion that a global system like a truss structure can not by analyzed in local context because each member in the structure does influence every other member in some way.

The failure of the local analysis led to the use of an analysis that considers the whole system at once. This analysis is the Direct Stiffness Method which is described in the next chapter.

Chapter 4

Direct Stiffness Method

The *Direct Stiffness Method*[4] [5] is a commonly used implementation of *Finite Element Methods*. It is a method that analyzes the structure as a whole, therefore being able to calculate the real deformation of the entire structure in only one step. In post processing of the result the axial stresses in the members will be calculated and the conclusion about the safety of the structure made.

Even though this method does give a global result in one iteration it is sometimes interesting running more then one iteration. Running more than one iteration might show that with one or even more failed members the structure might still stand. This is an important information for emergency scenarios. In this chapter the application of the *DSM* on space trusses will be shown in detail. The post processing and preparing the structure for consecutive iterations of the method will be discussed too.

4.1 Descriptions, labels, local and global coordinate systems

To describe the DSM some labeling rules will be presented. Since this application of DSM is focused on space trusses the main concern are the joints and the members connecting them. The joints will be identified by numbers, and the members by the joints they are connecting i.e. joints (1),(2),(3) and members (1,2) (2,3) (3,1). The properties and variables associated with joints or members are identified by the identifier of the joint or member in superscript i.e. $L^{(1,2)}$-length, $F^{(1)}$.

In general joints are identified by letters i.e. i, j, k, ... The rest of labeling stays the same. This way if there are joints (i),(j) and a member connecting them (i,j) the variables and constants in the system would be:

- $L^{(i,j)}$ - member length.
- $A^{(i,j)}$ - member cross section area.
- $E^{(i,j)}$ - member Young's modulus.
- $K^{(i,j)}$ - member stiffness matrix.
- $f^{(i)}$ - force acting on joint.
- $u^{(i)}$ - displacement of the joint.
- $x^{(i)}$, $y^{(i)}$, $z^{(i)}$ -joint coordinates.

The geometry of a structure is referred to either in global context in common Cartesian coordinate system x,y,z, or a local context which is a Cartesian coordinate system chosen so that it simplify the calculations needed to be done. Therefore it is important to differentiate between values taken in local systems and global systems when the given value has a direction besides its magnitude. The values taken in local systems are signed with a bar i.e. $\bar{u}^{(i)}$.

This method deals mainly with the forces and displacements in the joints. The forces and displacements both are broken down into their components that form an orthogonal system.

4.2 The global stiffness matrix

According to classical structural mechanics: "Displacement of the truss are defined by the displacement of the joints". The forces and displacements in the system are written down in a vector arranged by the number of the joint they are acting on. Hook's law tels us that the displacements are linearly proportional to the applied forces:

$$f = Ku$$

This can be expressed in component form as:

$$\begin{pmatrix} f_x^{(1)} \\ f_y^{(1)} \\ f_z^{(1)} \\ \vdots \\ f_x^{(n)} \\ f_y^{(n)} \\ f_z^{(n)} \end{pmatrix} = \begin{pmatrix} k_{x_1x_1} & k_{x_1y_1} & k_{x_1z_1} & \cdots & k_{x_1x_n} & k_{x_1y_n} & k_{x_1z_n} \\ k_{y_1x_1} & k_{y_1y_1} & k_{y_1z_1} & \cdots & k_{y_1x_n} & k_{y_1y_n} & k_{y_1z_n} \\ k_{z_1x_1} & k_{z_1y_1} & k_{z_1z_1} & \cdots & k_{z_1x_n} & k_{z_1y_n} & k_{z_1z_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ k_{x_nx_1} & k_{x_ny_1} & k_{x_nz_1} & \cdots & k_{x_nx_n} & k_{x_ny_n} & k_{x_nz_n} \\ k_{y_nx_1} & k_{y_ny_1} & k_{y_nz_1} & \cdots & k_{y_nx_n} & k_{y_ny_n} & k_{y_nz_n} \\ k_{z_nx_1} & k_{z_ny_1} & k_{z_nz_1} & \cdots & k_{z_nx_n} & k_{z_ny_n} & k_{z_nz_n} \end{pmatrix} \begin{pmatrix} u_x^{(1)} \\ u_y^{(1)} \\ u_z^{(1)} \\ \vdots \\ u_x^{(n)} \\ u_y^{(n)} \\ u_z^{(n)} \end{pmatrix} \tag{4.1}$$

The K matrix is the *Global Stiffness Matrix*. It's dimensions are $3n \times 3n$, where n is the number of joints which is multiplied by 3, because all 3 dimensions of the space need to be considered. The equation (4.1) is the *global stiffness equation*, in which the displacement vector is the unknown. This matrix has some important properties. It is symmetrical and sparse. Reasons for these properties to arise will be shown later. Also their usefulness will become clear in the next chapter.

4.3 DSM Step by Step

The steps of the DSM will be described using the generic member shown on 4.3.

4.3.1 Disconnection

This step of the DSM takes the truss apart considering each of the members alone. To each member (i,j) a local Cartesian system is assigned. To make the calculations in the local system simple the axis $\bar{x}^{(i,j)}$ is set so that it aligns whit the axis of the member (i,j).

Assuming i¡j, the origin of the local system is set at the joint i, and the direction of the $\bar{x}^{(i,j)}$ axis is set from joint i to joint j. The difference between the orientations of the local and global coordinate systems will be described as the cosines $\bar{x}^{(i,j)}$ with respect to x,y,z , the cosines $\bar{y}^{(i,j)}$ with respect to x,y,z , and the cosines $\bar{z}^{(i,j)}$ with respect to x,y,z.

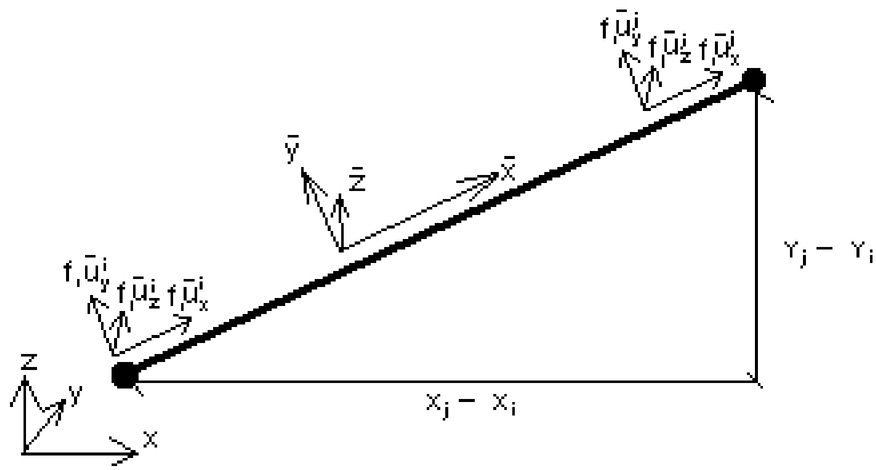

Figure 4.1: General member of the truss

4.3.2 The Local Stiffness Matrix

Now that each member has it's local coordinate system computations can be done on an x axis aligned generic (i,j). There are 6 force components acting on the member and it has 6 displacement components (degrees of freedom). Same as for the whole truss the relation between the force and displacement vectors of this member can be described as

$$\bar{f}^{(i,j)} = \bar{K}^{(i,j)} \bar{u}^{(i,j)} \tag{4.2}$$

in component form its

$$\begin{pmatrix} \bar{f}_x^{(i)} \\ \bar{f}_y^{(i)} \\ \bar{f}_z^{(i)} \\ \bar{f}_x^{(j)} \\ \bar{f}_y^{(j)} \\ \bar{f}_z^{(j)} \end{pmatrix} = \begin{pmatrix} k_{x_ix_i} & k_{x_iy_i} & k_{x_iz_i} & k_{x_ix_j} & k_{x_iy_j} & k_{x_iz_j} \\ k_{y_ix_i} & k_{y_iy_i} & k_{y_iz_i} & k_{y_ix_j} & k_{y_iy_j} & k_{y_iz_j} \\ k_{z_ix_i} & k_{z_iy_i} & k_{z_iz_i} & k_{z_ix_j} & k_{z_iy_j} & k_{z_iz_j} \\ k_{x_jx_i} & k_{x_jy_i} & k_{x_jz_i} & k_{x_jx_j} & k_{x_jy_j} & k_{x_jz_j} \\ k_{y_jx_i} & k_{y_jy_i} & k_{y_jz_i} & k_{y_jx_j} & k_{y_jy_j} & k_{y_jz_j} \\ k_{z_jx_i} & k_{z_jy_i} & k_{z_jz_i} & k_{z_jx_j} & k_{z_jy_j} & k_{z_jz_j} \end{pmatrix} \begin{pmatrix} \bar{u}_x^{(i)} \\ \bar{u}_y^{(i)} \\ \bar{u}_z^{(i)} \\ \bar{u}_x^{(j)} \\ \bar{u}_y^{(j)} \\ \bar{u}_z^{(j)} \end{pmatrix} \tag{4.3}$$

The matrix linking the joint forces and the joint displacements int a member is the *local stiffness matrix*. To construct the local stiffness matrix an approach similar to the one in chapter 3 will be used. Combining 2.1 and

2.2 we get

$$F = \frac{EA}{L}\delta \tag{4.4}$$

where F is the axial force in the member and δ is the axial elongation of the member. Form 4.3 we can express

$$F = \bar{f}_x^{(i)} = -\bar{f}_x^{(j)}\ ,\ \delta = \bar{u}_x^{(i)} - \bar{u}_x^{(j)} \tag{4.5}$$

Since the truss is made up of axial members that do not carry other than axial forces

$$\bar{f}_y^{(i)} = \bar{f}_z^{(i)} = \bar{f}_y^{(j)} = \bar{f}_z^{j} = 0 \tag{4.6}$$

Combining (4.2), (4.3), (4.4), (4.5) and (4.6) we get the matrix relation for the member where the main unknowns are the displacements.

$$\bar{f}^{(i,j)} = \begin{pmatrix} \bar{f}_x^{(i)} \\ \bar{f}_y^{(i)} \\ \bar{f}_z^{(i)} \\ \bar{f}_x^{(j)} \\ \bar{f}_y^{(j)} \\ \bar{f}_z^{(j)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{u}_x^{(i)} \\ \bar{u}_y^{(i)} \\ \bar{u}_z^{(i)} \\ \bar{u}_x^{(j)} \\ \bar{u}_y^{(j)} \\ \bar{u}_z^{(j)} \end{pmatrix} = \bar{K}^{(i,j)}\bar{u}^{(i,j)} \tag{4.7}$$

4.3.3 Globalizing the Member Stiffness Matrix

Now that relation (4.7) links the force vector and displacement vector in local coordinate system it is needed to find the relation doing this in the global coordinate system so that relationships between members of the truss can be stated. To transform a vector from one coordinate system to another it is multiplied by a transformation. Standardly a transformation matrix from system S_1 to S_2 is acquired by determining the coordinates of the base vectors of S_2 in S_1. Using that the local coordinate systems are aligned to the axes of the members the constructed transformation matrix for displacement is

$$\begin{pmatrix} \bar{u}_x^{(i)} \\ \bar{u}_y^{(i)} \\ \bar{u}_z^{(i)} \\ \bar{u}_x^{(j)} \\ \bar{u}_y^{(j)} \\ \bar{u}_z^{(j)} \end{pmatrix} = \begin{pmatrix} c_x & c_y & c_z & 0 & 0 & 0 \\ \lambda_{2,1} & \lambda_{2,2} & \lambda_{2,3} & 0 & 0 & 0 \\ \lambda_{3,1} & \lambda_{3,2} & \lambda_{3,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_x & c_y & c_z \\ 0 & 0 & 0 & \lambda_{2,1} & \lambda_{2,2} & \lambda_{2,3} \\ 0 & 0 & 0 & \lambda_{3,1} & \lambda_{3,2} & \lambda_{3,3} \end{pmatrix} \begin{pmatrix} u_x^{(i)} \\ u_y^{(i)} \\ u_z^{(i)} \\ u_x^{(j)} \\ u_y^{(j)} \\ u_z^{(j)} \end{pmatrix} = T^{(i,j)} u^{(i,j)} \tag{4.8}$$

where $c_x = \frac{x^{(j)} - x^{(i)}}{L^{i,j}}$, $c_y = \frac{y^{(j)} - y^{(i)}}{L^{i,j}}$, $c_z = \frac{z^{(j)} - z^{(i)}}{L^{i,j}}$ and the λ coefficients are not determinable, because the $\bar{y},\ \bar{z}$ axes can be rotated any way around the $\bar{x}$ axis. These indeterminate coefficients are not needed to arrive at a solution.

Similarly as for the displacements the transformation matrix for the forces is constructed

$$\begin{pmatrix} f_x^{(i)} \\ f_y^{(i)} \\ f_z^{(i)} \\ f_x^{(j)} \\ f_y^{(j)} \\ f_z^{(j)} \end{pmatrix} = \begin{pmatrix} c_x & \lambda_{1,2} & \lambda_{1,3} & 0 & 0 & 0 \\ c_y & \lambda_{2,2} & \lambda_{2,3} & 0 & 0 & 0 \\ c_z & \lambda_{3,2} & \lambda_{3,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_x & \lambda_{1,2} & \lambda_{1,2} \\ 0 & 0 & 0 & c_y & \lambda_{2,2} & \lambda_{2,3} \\ 0 & 0 & 0 & c_z & \lambda_{3,2} & \lambda_{3,3} \end{pmatrix} \begin{pmatrix} \bar{f}_x^{(i)} \\ \bar{f}_y^{(i)} \\ \bar{f}_z^{(i)} \\ \bar{f}_x^{(j)} \\ \bar{f}_y^{(j)} \\ \bar{f}_z^{(j)} \end{pmatrix} = (T^{(i,j)})^{-1} \bar{f}^{(i,j)} \tag{4.9}$$

The matrix in (4.9) is certainly the inverse of the matrix in (4.8) since it describes a inverse transformation. Combining (4.2) with (4.8) and (4.9) gives

$$f^{(i,j)} = (T^{(i,j)})^{-1} \bar{K}^{(i,j)} T^{(i,j)} u^{i,j}$$

where

$$K^{(i,j)} = (T^{(i,j)})^{-1} \bar{K}^{(i,j)} T^{(i,j)} u^{i,j}$$

is the *Member Stiffness Matrix in global coordinate system.* With the matrix multiplication carried out the λ coefficients drop out and the the final matrix is

$$K^{(i,j)} = \frac{E^{(i,j)}A^{(i,j)}}{L^{(i,j)}} \begin{pmatrix} c_x^2 & c_xc_y & c_xc_z & -c_x^2 & -c_xc_y & -c_xc_z \\ c_xc_y & c_y^2 & c_yc_z & -c_xc_y & -c_y^2 & -c_yc_z \\ c_xc_z & c_yc_z & c_z^2 & -c_xc_z & -c_yc_z & -c_z^2 \\ -c_x^2 & -c_xc_y & -c_xc_z & c_x^2 & c_xc_y & c_xc_z \\ -c_xc_y & -c_y^2 & -c_yc_z & c_xc_y & c_y^2 & c_yc_z \\ -c_xc_z & -c_yc_z & -c_z^2 & c_xc_z & c_yc_z & c_z^2 \end{pmatrix} \tag{4.10}$$

As we can see the member stiffness matrix is a symmetric. This property will be useful later

4.3.4 Global Stiffness Matrix Construction

In the disconnection step the truss was taken apart now it has to be reassembled so that it can be analyzed as a whole. The key to the assembly process is to add the contributions of each member to the global stiffness matrix. This construction process has 2 main rules.

- "*Compatibility of displacements*:The displacement of all members meeting at a joint are the same."
- "*Force equilibrium*: The sum of forces exerted by all members that meet at a joint balances the external force applied to that joint."

The meaning of the first rule is easy to interpret it means that the whole truss moves as one. The second rule simply states what is implied by the transitional equilibrium condition.

For the merging of the globalized stiffness equations the displacement vectors of all members are extended to incorporate all components from all the joints. The stiffness matrices are extended by rows and columns of zeros, respectively and the force vectors are extended also to incorporate the components from other nodes. This way if the original truss was $n \times n$ for an arbitrary member (i,j) it's member stiffness equation is extended to

$$f^{(i,j)} = K^{(i,j)}u^{i,j}$$

where $f^{(i,j)}$, $u^{(i,j)}$ are $n \times 1$ and $K^{(i,j)}$ is $n \times n$. According to the first rule of merging the superscript of u can be dropped since the displacements are the same relative to all members.

Now all the equations are summed up.

$$\sum f^{(i,j)} = \sum (K^{(i,j)})u$$

According to the second rule of merging $\sum f^{(i,j)} = f$, where f is the external force vector (external forces applied to each joint). As we can see if $\sum K^{(i,j)}$ is denoted as K(the global stiffness matrix) the resulting equation

$$f = \sum (K^{(i,j)})u$$

is the global stiffness equation.

This way of constructing the global stiffness matrix shows how are the rules of merging utilized and shows the symmetric property of the global stiffness matrix(we are adding up symmetric matrices that were extended by the same number of zero columns and rows). However extending the matrices so they have the same dimensions and can be summed up is not necessary when implementing the assembly process. The member matrices can be mapped onto the global matrix using the number of the joint. Each entry in the member stiffness matrix is added to a corresponding entry in the global stiffness matrix.

$$K_{p,q} = \sum K_{l,m}^{(i,j)}, \ \ for \ \ l = 1, ...6 \; m = 1, ...6 \; where \; p = MFT^{(i,j)}(l) \ \ q = MFT^{(i,j)}(m)$$

MFT is the *Member freedom table*. It is determined by the joint numbers in the member

$$MFT^{(i,j)} = \{i, i+1, i+2, j, j+1, j+2\}$$

Constructing the global stiffness matrix using the MFTs shows the second important property of the matrix. Only those entries of the matrix are not zero which are represented in at least one member's MFT. This means that if there is no member between joint (i) and (j) than the entries $K_{i,j}$ trough $K_{i+3,j+3}$ are 0. Since in a common truss joints are connected only to a relatively few joints (in this application it is around 20, while the total number of joints border 3000) there are lots of "missing" members so the resulting global stiffness matrix is sparse.

In the implementation of the DSM in this project all the steps up until now are done concurrently member by member. This way the member stiffness matrix is never actually calculated and stored as a whole. It's entries are calculated only as they are about to be merged into the global stiffness matrix using the MFT. This is a simple way to save some storage space.

4.3.5 Boundary Conditions And Solving

To solve the global stiffness equation the first the external forces are applied thus constructing the force vector. As mentioned in the beginning of this chapter this implementation makes it possible to run the analysis several times consecutively. For this to work the reaction forces calculated in the previous iteration of the analysis need to be applied at this point too. If this is the first iteration of the analysis the reaction forces are still zero thus they will not influence the result.

The global stiffness matrix constructed so far is singular to, so the global stiffness equation can not jet be solved. This is because up until now no boundary conditions were applied. Boundary conditions in this project are the support conditions of the truss. Joints that are externally supported are considered immovable.

This condition is applied by setting the corresponding entries in the force vector to zero clear (set to zero) the corresponding rows and columns in the global stiffness matrix and set 1 on the corresponding diagonals entries in the matrix to get a non-singular matrix.

After the support conditions have been applied we should be able to solve the global stiffness equation. If it still yield no result it means that the truss was not statically stable to begin with. If there is a single solution to the equation then it is the displacement vector for the truss. In this project the Cholesky factorization was used to solve the equation. This is described in the next chapter.

4.3.6 Post Processing

In post processing the the main goal is to recover the axial stresses within the members of the truss and the reaction forces exercised by the members.

This project uses the most straightforward method there is, for the recovery of the viable data. The displacements calculated by solving the master stiffness equation are applied to the truss while the members retain their property L(rest length). Than using the combination of Hook's law in (2.1) the axial stress is determined.

$$\sigma = \frac{E}{L}\delta, \quad where\ \delta = |(i,j)| - L^{(i,j)}$$

, the sign of the axial stress determines whether the member is under tension or compression. For positive axial stress the member is under tension.

The magnitude of the reaction force exercised by the member is determined by

$$F = \sigma A$$

This force has always the acts along the direction of the member and its orientation is determined by the sign of F. For positive sign its tension therefore the force points inward. These forces are the ones used to allow consecutive running of the analysis.

When the axial stress and reaction forces are calculated the axial stress is compared to the yield stress of the material the member is made of. The sign of the axial stress is not considered in this case. If the axial stress is higher than the yield stress the member is declared unsafe and is not included in the next iteration of the analysis. The ratio of axial stress to yield stress to is used to determine the color of the member when displaying the results of the analysis. Green is used for members under low stress, red is used for members under compression and blue for members under tension.

Chapter 5

Solving DSM Equations

This chapter will detail the equation solver and the storage format that is used in the implementation of DSM in this project. If we consider the system requirements of storing and solving the global stiffness matrix, it is obvious that they would be enormous when the whole matrix is stored and common Gaussian elimination is used. The storage requirements are $(n)^2$ where n is the order of matrix. For order of 10000 (which matches roughly 3330 joints since the degree of freedom of one joint is 3) it is 400MB. The computation time for the Gaussian elimination is full $O(n^3)$, this is about 10^{12} FLOPs, which translates to hours of computing on a PC. The solving and storing of the global stiffness matrix needs to be made more efficient.

5.1 Skymatrix

As has been already pointed out the global stiffness matrix is symmetric and sparse. These properties can be exploited to save a great deal of storage space. Since the matrix is symmetric it is enough to store one half of it. In this implementation the lower triangular matrix is stored. This measure saves half of the storage space the next to use the sparse property.

There are several method of storing sparse matrices[3] here one of them will be used. The idea is not to store the entries which are zero. Assuming that at least one member end in each joint (otherwise that join would be redundant) the diagonal of the matrix is non-zero. Therefore the diagonal of the matrix will be stored ether way. The lower triangular matrix is stored row by row in a value vector so that only the entries form the first nonzero entry to the diagonal entry are stored. An indexing array is needed to store the indexes of the diagonal entries (to be able to find entries in the value vector). Since as all entries on the diagonal are non-zero the indexing vector's size is n for a matrix of order n. A matrix

$$K = \begin{pmatrix} 3 & 9 & 0 & 5 & 0 & 0 \\ 9 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 3 & 0 & 0 \\ 5 & 0 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 \end{pmatrix}$$

would be stored as

$$K = (D, V) \;\; D = \{0, 2, 3, 7, 9, 10\}, \; V = \{3, 9, 7, 8, 5, 0, 3, 2, 1, 5, 6\}$$

The resulting storage format is a variation of the *skyline* storage format. In accordance with the storage format name the matrix stored in this format is named *skymatrix*. The ratio $B = \frac{S}{n}$ where S is the space required for the storage of an n ordered matrix is the *mean bandwidth*. For this application the ratio greatly depends on the geometry of the structure, but can be in order of $B = n^{\frac{2}{3}}$(By laying the nodes in layers). Comparison test ran against a fully stored matrix confirm this. For the matrix of order n=10000 discussed beforehand the storage space will be 18 MB using 4 byte floating point numbers.

The solver used to solve the global stiffness equation needs some additional information about the skymatrix. Namely it's needed to know at which index a row start, and the index of last entry in a column. The row starts are computed easily and could by subtracting the diagonal vector entries form each other. This could be done while solving the equation, but it is a frequently required information therefore its advantageous to precalculate it.

The index of the last entry in a column can not be recovered directly from the skymatrix. It is done by the simple algorithm:

```
for(int i=0; i<N; ++i) ColumnEnds[i] = i;
for(int i=N-1; i>=0; --i)
```

```
for(int j=RowStarts[i]; j<i; ++j)
        if(ColumnEnds[j]<i) ColumnEnds[j] = i;
```

In this context N is the order of the matrix.

5.2 Cholesky solver

The cholesky solver uses the cholesky factorization of a symmetric matrix to solve the global stiffness equation. The main solving steps are:

- Factorization: $K = LDU$
- Forward reduction: $Lz = f$
- Diagonal scaling: $Dy = z$
- Back substitution: $Uu = y$

In the steps K is the global stiffness equation, u is the primary unknown vector, y and z are intermediate vectors.

5.2.1 Factorization

Symmetric matrices can be factored as:

$$K = LDU = LDL^T$$

where L is a lower triangular matrix and D is a nonsingular diagonal matrix.

An algorithm for the factorization can be accuired by carrying out a sample multiplication

$$K = LDL^T = \begin{pmatrix} D_1 & & & (symmetric) \\ L_{2,1}D_1 & L_{2,1}^2 D_1 + D_2 & & \\ \vdots & \vdots & \ddots & \\ L_{j,1}D_1 & L_{j,1}L_{j-1,1}D_1 + L_{j,2}D_2 & \cdots & \sum_{k=1}^{j-1}(L_{j,k}^2 D_k) + D_j \end{pmatrix} \tag{5.1}$$

From 5.1 the following recursive relations can be written.[6]

$$L_{ij} = \frac{1}{D_j}(K_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{jk}D_k), \text{ for } i > j \tag{5.2}$$

$$D_i = K_{ii} - \sum_{k=1}^{j-1} L_{ik}^2 D_k \tag{5.3}$$

From relations 5.2 and 5.3 we can see that an entry in the factor matrix can be calculated if the entries left and above are already calculated. This is the basis of the factorization algorithm. It simply starts at the entry $K_{1,1}$ and works trough the matrix, replacing the entries as it goes. There is no extra storage space required.

For every row its enough to start at the first nonzero entry (which is actually the first entry sored in the sky matrix for that row) since 5.2 implies that all entries before that would be zero (K_{ij} and L_{ik} would be zero). For the entries that actually need to be calculated the k can be started at the firs nonzero entry in the row. This way the factorization time can be expressed as

$$O(nB^2)$$

where B is the mean bandwidth. Assuming $B = n^{\frac{2}{3}}$ the number of FLOP's required for cholesky factorization is smaller by orders than when using Gaussian elimination. For the example matrix of order 10000 it's FLOP's in order of 10^9, which is still manageable on fast PC.

5.2.2 Solving

With the completion of the factorization the global stiffness equation is not yet solved. The last 3 steps still need to be applied. The computational complexity of these steps is however expressed as

- Forward reduction: $O(nB)$
- Diagonal scaling: $O(n)$
- Back substitution: $O(nB)$

therefore the factorization time is the main concern regarding the computational complexity of the global stiffness equation solving.

The final implementation note to this chapter is that while the truss is built so that it should be stable at the beginning of the analysis process the consecutive iterations could lead to destabilization (too many members are declared unsafe under pressure). To assure correct run of the solving process after factorization is done the D matrix is checked whether it is really nonsingular.

Chapter 6

Simulation Structure

Thus far the statical analysis of structure was described on 3D space trusses, this project however concerns itself with more than just simple trusses. It does simulate structural members that carry bending and sheer forces(beams), and also members that have significant dimensions in more than one dimension(wall segments, columns). The simulation of these structural elements is done by approximating their inner structure by a complex space truss. The description of how is it done follows.

In this chapter significant differences are between *scene nodes and elements* and *simulation nodes and elements*. Scene nodes and elements are built by the user who also sets their properties, while simulation nodes and elements are parts of the *simulation structure* and are generated automatically. The simulation structure is the space truss the analysis is carried out on.

6.1 Scene Nodes

Scene nodes are the basic building blocks of a complex structure to be analyzed. They are to be interconnected using different elements. While a simulation node has no volume this is not true for a scene node. It does have volume and is approximated with a cube so elements that have thickness or any kind of cross section could be connected to it. This decision was made so it would be possible to regard a wall as an element that has 2 significant dimensions (it's thickness is usually far smaller than its height or width) and a beam as an element with only 1 significant dimension, the non significant dimensions are in this case set to a constant.

The constant that the non significant dimensions are set to is actually the size of the cube's edges that symbolizes the scene node. This constant was set to 0.1 meters. This decision was made based upon what kinds of structures would be simulated using this project. The shortest assumed wall segments in this project should be around 1 meter therefore the 0.1 meter resolution should be enough.

The cube symbolizing the scene node has 8 corners which translate to 8 simulation nodes (joints of the truss). The simulation nodes are connected each to each other with simulation elements (members of the truss). The properties E and Y (Elasticity and Yield stress) of the simulation elements are set so that they reflect the hardest of the scene elements material connected to the node. The lengths of the simulation elements are calculated from the position of the simulation nodes they connect, while their cross section areas are set so that the sum of the volumes of the 28 simulation elements making up the node is equal to the volume of the cube symbolizing the node.

The scene node has one last property that is set to the simulation nodes approximating it. This property is tells whether the scene node is grounded(is the structure supported in this scene node). This grounded property is used by the DSM to set the boundary conditions.

Connecting scene nodes is an important operation when building scene elements. To connect 2 nodes the first we determine which faces of their symbolic cubes are facing each other. That are the faces that would be intersected by the line segment connecting the centers of the cubes. Than the simulation nodes of selected face of the first scene node are connected to every simulation node in the selected face of the second simulation node. Thus adding 16 new elements to the simulation structure. This kind of connection can carry forces of any direction from every simulation node involved in it, because it forms a statically stable truss (members form triangles).

6.2 Scene Elements

6.2.1 Scene Beams

Scene beams are the simplest scene elements. They consist of 2 interconnected scene nodes. A beam has material properties and a cross section area property. The simulation elements created during the connection of the nodes are referred to as the simulation elements of the beam. The material properties of the simulation elements are set to the material properties of the beam. The length of simulation elements is calculated from the simulation node positions. Their cross section is calculated again so that the sum of volumes equals the volume of the beam.

The simulation properties of a beam constructed this way do approximate the properties of a real life beam to which forces are applied only on its ends. The ability to carry bending forces is assured by the cross connections within the simulation elements of the beam. The accuracy of the simulation is best when we try to simulate a beam that has a cross section area of $0.1^2 m^2$ because that is the actual cross section are of the simulation structure beam. The difference between this and the cross section are property of the beam is compensated by the setting of the simulation elements cross section area.

6.2.2 Wall and Floor Sections

Walls and floors are designed by the same scene elements the simple and complex plane element. These elements have 4 interconnected scene nodes which lay in one plane . The simulation elements of these scene elements are set up the same way as the simulation elements of the beam. Again the cross connection between the nodes (both simulation and scene) should give the element the ability to carry forces applied to the scene elements approximately the same way as an they would be carried be a real life wall or floor.

The difference between the simple and complex plane element is that the complex element allows for nodes within its outer edges. These nodes are connected to the nodes in the opposing edge to ensure approximation of force carrying ability. The number of nodes in the opposing edges does not need to equal. This scene elements was included so that building floors having different inner design could be sacked on each other without the need of propagating node structure of each floor upwards instead just include them in one of these more complex scene elements. The force propagation will be approximately the same and the number of scene nodes can be decreased.

6.2.3 Columns

Columns are scene elements that have 8 interconnected scene nodes. These nodes form a parallelepiped. This is the only scene element that has its volume calculated directly form the positions of its scene nodes, because it has no non significant dimensions. Because of the resolution set by the size of the scene node approximation the column is however not a good choice for simulating columns smaller then $0.3 \times 0.3 \times 0.3\ m^3$.

6.3 Forces in the Simulation Structure

For the simulation to make any sense forces need to be applied to the simulation structure. Every simulation node stores the force applied to it. This force is set to zero when the simulation structure is built. After the structure is built the forces are applied. Forces might be applied in more parts, in that case they are added to the force already applied to the simulation node.

6.3.1 Gravitation

Gravitation is the most important force when studying the statics of a building. It gives the elements their weight. The force created by gravitation can be calculated as

$$F = V\rho g \tag{6.1}$$

,where V is the volume of the element ρ is the density of the material, and $g = 9,81\frac{N}{Kg}$ is the standard gravity. The force generated by gravity always acts along the negative z axis.

To apply the gravity not the scene elements are considered but the simulation elements. Since they are calculated so that the sum of their volumes equals the sum of the scene element volumes the total force distributed will be the same. By considering the simulation elements the distribution of weight on the simulation structure does match its geometry and since the geometry of the simulation structure was built so that it approximates the inner structure of the elements this distribution of weight approximates the real distribution of weight amongst the scene elements.

For one simulation member the weight distribution is done by calculating the force according to (6.1) and than evenly divided into the simulation nodes connected by the simulation member.

6.3.2 External Forces

External forces are used to test the structure under load, i. e. if we were to roll a car on a bridge or put furniture and people in a building. In this project 2 kinds of external forces are used. The first kind is applied on the scene nodes, the second kind is applied distributed on any scene element.

Applying the firs kind of force is easy. Simply dividing it evenly between the simulation nodes making up the scene node.

For applying the distributed force the same method will be used as when applying the gravitation. The force is divided between the simulation members of the scene member at the rate of the ratio of their length to the sum of the lengths of all of the simulation members of the scene member. When the force for one simulation member is calculated it is evenly distributed between the simulation nodes connected by the simulation member. This approach gives satisfactory results for the same reasons as the applied gravitation.

6.4 Modeling Guidelines

Constructing the simulation structure as described in this chapter does let a degree of error into the simulation. The accuracy loss stands mainly from the approximations made during describing the inner structure of the scene elements. This way the simulation does simulate the how the different elements act on each other but the it does not resolve the plane stress problem(how is the stress distributed inside a scene member).These errors can be managed by adhering to some guidelines during the design of the scene.

One important guideline is not to build too big or too small scene elements. The optimal size of the scene elements should be between 1 to 10 meters along their significant dimensions. For small elements the 0.1 meter size of a scene node would significantly influence the actual size of simulation members in comparison to the size of the scene member. For big elements the plain stress problem becomes significant. Also the ratio of sizes of the scene member along their significant dimensions should be at most 1:5 so the cross connections in the simulation structure are not at too acute angles which would make them ineffective.

To cope with the plane stress problem smaller scene elements should be used in the regions of the structure, where high loads are expected. For example it would be wrong to build a 20m long bridge with deck sections 5m long. Other high stress areas are support beams and columns of a building, arches etc.

Scene elements that have non significant dimensions should not be used to simulate structure components where the size along the "non significant" dimension is larger the 0.4m for this use of the column scene element is better suited.

6.5 Conclusion

In this work the basic elements of statics simulator were showed. The described statics simulator works in space while most of the statics simulator I have encountered be it games or even applications used by engineers work by projecting the problem into a plane and running the analysis as a 2D problem. By using the cholesky method of solving the equations during the analysis a structure containing more than 300 scene nodes can be simulated in under 1 minute on a 2 G Hz CPU as shown in one of the sample scenes. Although admittedly the accuracy of the results the simulation gives is not absolute, it is high enough so that this application can be used as a teaching aid or for demonstrating how structures behave under loads in general. This project also allows for adding new materials just by editing the material database file. In this project the simulation structure on which the analysis was carried out was kept as simple as possible by using only a space truss, but it is possible to incorporate other simulation elements using methods form a more general field of Final Element Method. Further development would add more complex simulation elements with higher degrees of freedom thus allowing to calculate bending and rotation directly by the DSM. Also effort would be made to parallelize the matrix solving allowing for better use of multi core processors.

Appendices

.1 About the contents of the CD

The CD contains the binaries for the application in the "Binaries" folder. In the folder "Binaries/Samples" some sample scenes are saved. Also the CD contains the PDF format of this thesis in the "Thesis" folder and the source files in the "Source" folder.

BridgeSample This sample file contains a warren truss bridge design spanning 10m loaded to its limits.

BuildingSample This sample shows a building design, to demonstrate the use of more complex design elements.

GridSample This sample contains 300 scene nodes connected into a grid by beams supported along one of its edges. The sample is supposed to show computational complexity. By my estimate it sill should run in under 1 minute. In comparison 40 scene nodes were enough to design the bridge and under 100 for the house. The grid in this case is not able to withstand its own weight.

IterationSample This sample is supposed to show how even when some elements of the structure break the iterative steps show that the structure might still stand.

.2 User manual for the program

The program on the CD is really easy to use every function that it has is tooltiped for the convenience of the user. To program runs basically in one of the two modes. In both modes the scene can be translated by dragging the right mouse button, and rotated by dragging the right mouse button while holding left ctrl. The mouse wheel zooms in and out in orthogonal view while it changes the camera height in perspective view. The + and - buttons on the keypad increase/decrease the height of the drawing plane. This height can be directly set through the input in the top right corner.

Select mode This mode allows manipulation with already constructed scene elements and nodes. Nodes and elements are selected using the left mouse button. Only nodes or elements can be selected at one time. There are operations that can be applied to the selection. Some of these operations are in the selection menu. Other can be accessed by the apply material, apply cross section and toggle grounded buttons. If a force is right clicked a non modal window will appear in which the force can be modified, while when a the apply force menu item is selected the force given in the modal window which appears is applied to all selected items. For forces pointing downward use forces whit negative z component.

Build mode This mode allows building the scene itself. To switch to this mode toggle the appropriate button (it has a hammer icon on it). Once in the build mode the most important control elements are the choice inputs on the top. In these inputs the element to build it's material and it's cross section are chosen. Once the right element to build is chosen building it entail simply left clicking on the drawing grid for the nodes or clicking on the nodes to connect into the elements for every other element. For beam two nodes need to be chosen. For plane sections 4 or more nodes in the order to be connected in need to be selected. For column 8 nodes. For every element here are restrictions i. e. a plane segment must be in a plane the column must be a parallelepipedon.

Simulation The operations associated with simulation are in the simulation menu. Run simulation starts one the simulation step. For relatively small structures it finishes within second. To run another step the simulation must be started again, this will not cause the simulation to reset. The simulation structure is automatically there was none before running the simulation. Using the ReBuild menu item will cause the structure to reset. Only those scene nodes are included in the simulation structure that are at least part of one scene element.

Simulation results The results of the simulation can be observed by switching on the coloring scheme according to the inner stress. The simulation structure can be observed by switching off the drawing solids toggle. The simulation structure is always colored according to the stress and it also reflects the displacement of calculated by the analysis also the broken members are not included in the simulation structure. To color only the broken members of the simulation structure switch on the appropriate toggle button. If coloring in not enough information about the stress, the information bar on the bottom displays stress as a ratio between yield stress and axial stress for the item the mouse point at.

Bibliography

[1] L. G. Kraige J. L. Meriam. *Engineering Mechanics Volume 1 Statics, 5th Edition*, pages 138, 168–179, 188. John Wiley & Sons Inc, NY, USA, 2002.

[2] W. A. Nash. *Schaum's Outline of Theory and Problenc of Strenght of Materials, 4th Edition*, pages 1–8. The McGtaw-Hill Companies Inc, USA, 1998.

[3] M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. Van der Vorst R. Barrett. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, pages 64–68. Society for Industrial and Applied Mathematics, Philadelphia, USA, 1994.

[4] The direct stiffnes method i. http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/IFEM.Ch02.d/IFEM.Ch02.pdf, April 2009.

[5] The direct stiffnes method ii. http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/IFEM.Ch03.d/IFEM.Ch03.pdf, April 2009.

[6] D. WatkinsWilliam. *Fundamentals of Matrix Computations, 2nd Edition*, page 84. John Wiley & Sons Inc, NY, USA, 2000.