The design of a software program for the simulation of a two-dimensional model of the knee-joint

Citation for published version (APA):

Document status and date:
Published: 01/01/1992

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
The design of a software program for the simulation of a two-dimensional model of the knee-joint
Report number 92.013 A

by: Niek van Nunen
Frank Peters

Eindhoven University of Technology, December 1991
Summary

This paper describes the design of a software program for the simulation of a two-dimensional model of the knee-joint. The purpose of the program is to train students in using and understanding the basics of parameter variation. The joint is seen as a simple planar four bar mechanism with rigid links. The analysis is divided into three parts:
First an exact problem definition is defined, followed by a graphical analysis. This graphical analysis is again the basis for the mathematical analysis, the third part. With the results of the mathematical analysis the software program is written, which calculates the centrodes of the mechanism, and given the tibial contact profile (a straight line or a parabola), it also calculates the femoral contact profile. Furthermore it is possible to calculate the effects of the variations of parameters, and a diagram which shows the slip ratio as a function of the flexion angle is drawn. The most important reason to conclude that the model is correct, is the written subroutine MOVIE, which draws the mechanism and the four curves in several positions. Finally a short description of the program is given, as well as the program text itself. It is recommended to run the program on at least a 80286 PC/AT with coprocessor. Also a colour graphics card with colour monitor is recommended. The last appendix contains a user's manual.
The design of a software program for the simulation of a two-dimensional model of the knee-joint.
## Contents

10. **Model validation** ........................................ 33  
   10.1 Conditions resulting from the model .................. 33  
   10.2 Symmetrical mechanism .................................. 33  
   10.3 The Perspex model ..................................... 33  
   10.4 Subroutine MOVIE ..................................... 34  

11. **Conclusions** ........................................... 35  

12. **Literature** .............................................. 37  

Note: This report is accompanied by a number of appendices and a user's manual. These parts have the same number as this report, but are indexed B and C.
2. Symbols

This list of symbols contains the symbols used in the sections 1 to 12. The symbols that are used in the program text (see the appendices) are named in the program text itself, so they are omitted here.

\[ \begin{align*}
A & : \text{Hinge} & h & : \text{Vertical displacement} \\
B & : \text{Hinge} & IC & : \text{Instantaneous center of velocity} \\
O & : \text{Hinge} & l & : \text{Distance centrode-contact profile} \\
P & : \text{Hinge} & l_i & : \text{Length of link } i \\
OA & : \text{Anterior Cruciate Link (ACL)} & l"_i & : \text{Distance } A \text{ to } IC \\
AB & : \text{Femoral Link} & R_i & : \text{Rotation matrix} \\
OP & : \text{Tibial Link} & x_f & : x\text{-coordinate tibial centrode} \\
PB & : \text{Posterior Cruciate Link (PCL)} & x_i & : x\text{-coordinate tibial contact profile} \\
a & : \text{Slope or concavity/convexity} & x_0 & : \text{Horizontal displacement} \\
a' & : \text{Vector component} & y_f & : y\text{-coordinate tibial centrode} \\
A_i & : \text{Polynome coefficient} & y_i & : y\text{-coordinate tibial contact profile} \\
b' & : \text{Vector component} & \\
C_l & : \text{Length of vector} & \\
CA_x & : \text{Constant} & \\
CB_x & : \text{Constant} & \\
CC_x & : \text{Constant} & \\
\end{align*} \]

Greek symbols: angles (degrees)

\[ \begin{align*}
\alpha & : \text{Variable} & \delta & : \text{Variable} \\
\alpha_s & : \alpha \text{ at input} & \theta & : \angle OPB \\
\beta & : \text{Variable} & \varphi & : \angle AOP \\
\gamma & : \text{Flexion angle} & \chi & : \text{Rotation angle} \\
\end{align*} \]

Vectors:

\[ \begin{align*}
\vec{A} & : \text{Vector from } O \text{ to } A & \vec{n}_x & : \text{Unit vectors} \\
\vec{\varepsilon}_{\text{f,t}} & : \text{Tangent to femoral centrode} & \vec{F}_f & : \text{From } O \text{ to tibial centrode} \\
\vec{\varepsilon}_{\text{t,t}} & : \text{Tangent to tibial centrode} & \vec{F}_t & : \text{From } O \text{ to tibial contact profile} \\
\vec{F}_C & : \text{From } O \text{ to femoral centrode} & \vec{T}_C & : \text{From } O \text{ to tibial centrode} \\
\vec{I}_i & : \text{Vectors representing the four links} & \vec{r}_i & : \text{Tangent to tibial contact profile} \\
\end{align*} \]
3. Preface

In fundamental research parameter variation has become an often used tool. So it is necessary to give students the opportunity to train themselves in using the principle of parameter variation. The curriculum of the faculty of Medical Engineering Technology of the Eindhoven University of Technology therefore contains a simulation practical. In order to expand this practical a software program had to be designed, which would make a simulation with parameter variation possible. The object to be simulated was chosen to be a 2 dimensional kinematic model of the knee joint. The software should be able to calculate the effects of the change in various parameters, which further on in this paper will be specified.

This problem was considered very suitable to be solved in the form of one of the two larger practicals (about 240 hours each) that occur in our curriculum. This paper is the result of our work.

Because it would be foolish to do unnecessary work, we went searching for already existing solutions. There is a lot of literature on four bar mechanisms, but there seemed to be little interest in the type of solution we are looking for. Specialists on kinematics at the Eindhoven University of Technology made the same suggestion. Therefore we had to start from scratch.

When the mathematical model was finished, we expected that the programming should not take too much time, but here we were wrong. Especially the use of mathematical and graphical libraries proved to be cumbersome. After these problems had been tackled, the program worked satisfactorily, and we think it is a usable tool for a simulation practical. We hope it will be used for that purpose.

Finally we would like to thank the persons who have been very helpful to us. These persons are:

Prof. dr. A. Huson
Dr. ir. A.A.H.J. Sauren
Ing. H.A. Bulten
Drs. A.W.M. de Jong
Dr. G.J. Visser

Niek van Nunen
Frank Peters

The design of a software program for the simulation of a two-dimensional model of the knee-joint.
4. Introduction

The bones that meet in a joint, as well as the ligaments that hold the bones together, can be analysed as a mechanical linkage. Since the movements allowed to the bones at the human knee occur mainly in the sagittal plane, much can be learned by treating the knee as a two-dimensional single-degree-of-freedom linkage moving in a single plane. Figure 1 shows a human knee from which the lateral femoral condyle has been excised, exposing the cruciate ligaments. The ligaments, together with the two bones, form the cruciate linkage. OP will be called the *tibial link*. AB is the *femoral link*. The lines OA and PB represent the anterior (ACL) and posterior (PCL) cruciate ligaments, respectively.

![Image](image_url)

Figure 1.

As mentioned before, the initial task was to design or write a software-program which simulates a two-dimensional kinematic model of a knee joint. The program initially had to satisfy several demands. Whether or not it would be possible to satisfy all demands would appear along the way. The initial task description was:

*The design of a software program for the simulation of a two-dimensional model of the knee-joint.*
The model is based on the concept of a four bar mechanism. Two (not necessarily) crossed bars represent the cruciate ligaments. These two bars can be either rigid or slightly elastic. The other two bars each are expanded with a plane, in such a way that during the motion of the four bar mechanism there will be a continuous contact between the two profiles. These profiles represent the contact planes of the femur and tibia.

The input of the model contains:
- the coordinates of the rotation centers of hinges of the links. Herewith the dimensions of the hinges are determined.
- The stiffness of the links.
- The profile of one of the two contact profiles.
- The initial position of the mechanism i.e. the flexion-angle of the joint.
- The increment of the flexion angle

The output should contain:
- The curve of the second profile.
- The position of the contact point.
- The position of the instantaneous center of rotation, as well as both centroids.
- Instantaneous translation velocities (slip) in the contact point.

The input possibilities of the parameter variation:
- A new position for one or more of the revolutes of the bars.
- A change in the geometry of one of the contact profiles.

The output of the parameter variation:
- The strains in the bars in case of a change in the position of revolutes or contact profiles.
- the forces in the bars.
- forces in the contact point.

In the next section a strategy will be presented, according to which the problem will be worked out.
5. Strategy

In order to carry on through the whole process smoothly it is necessary to use a well considered strategy. The strategy chosen is the following:

First the exact problem definition will be determined. Which of the above mentioned demands will be realised? Should all demands be realised, but then taking the risk not to be able to end up with a completely working software program (the time to be put in is about 240 hours)? The strategy chosen here is to first consider the most elementary parts of the problem, and then with an eye on the time left extend the model with the less important specifications. Whether or not any specifications will be implemented will be decided before starting to write the software program, so that the program can be structured well and does not have to be changed afterwards. What the most important specifications are will be determined in the next section.

The next step is the construction of a graphical analysis. This will give a first indication of how the mathematical problem is to be solved. Then a mathematical model will be formulated which consists of both analytical and numerical parts. In this step the model will be formulated in such a way that the next step, the programming, can be carried out rather easily. Also an eye will be kept on the fact that the program shouldn't work to slow. The consequence of this is that sometimes numerical approximations will be necessary where an exact but slow solution is possible.

The last step contains the programming itself. This part also includes the testing of the program and if necessary the writing of a manual.

6. Adjusted problem definition

In this section a global view is presented from which specifications initially are intended to be worked out. Later on this may very well be extended or reduced depending on the time left.

In the first place the users of the software will have to be able to study the effects on the form of the femoral contact profile when the input consists of:

a. several geometries of the tibial contact profile.

b. crossed and uncrossed ligament configurations.

c. slip velocities.

These elements now will be worked out in the global method that will be used as the basis of the mathematical model.
7. Global method: a graphical analysis

7.1 The flexion angle $\gamma$

The basis of the model will be a simple four bar mechanism. This is one of the most elementary mechanisms in kinematics. For the mechanism configurations as drawn in figure 2 are taken. To make a link to a real situation an arbitrary flexion angle $\gamma$ is defined. Figure 1 suggests that when the tibial link OP is parallel to the femoral link AB, the joint is flexed to about 50°. So positions can be defined where the joint is fully extended and fully flexed. From literature [1] it is known that when the angle $\alpha$ in figure 2 is about 50°, the joint is in full extension (in other words, $\gamma$ is zero). In other literature [2] it is found that the maximum flexion angle is about 140° (this means $\alpha$ is -90°). In the following analysis this range (-90° < $\alpha$ < 50°) will be used as the range within which the calculations will be made.

7.2 The input.

When the software program is finished, the students will have to give some input. This input will consist of:

1. The form of the mechanism, i.e. whether the mechanism is crossed or uncrossed.
2. The coordinates of the hinges A and B. The coordinates of the hinges O and P are already determined. Herewith all the dimensions (i.e. lengths and a certain flexion angle $\gamma$) and kinematics are determined.
3. The curve of the tibial contact profile. The choice out of a straight line or a parabola will be given. The choice for a circle or an ellipse was neglected, because these curves are rather similar to the parabola, when the latter is only slightly concave or convex. These forms (straight line and parabola) can be adjusted by changing the parameters of the basic equation of each of the two possibilities.

When this input is given the situation is as shown in figure 2. Now a method to calculate the curve of the femoral contact profile will be presented.
7.3 Construction of the centrodes.

When the relative positions of the links are changed, for example when we bend the knee, the tibial link and the femoral link carry out a motion relative to each other. At each position of the mechanism a point can be determined were the two links have zero relative velocity. This means that at this point the two links only rotate relative to each other. These points are called *Instantaneous Centers of velocity (IC's)* or *poles*. In figure 3 the IC is constructed for an uncrossed mechanism in a particular position.
All IC's together (from all positions of the mechanism) form a curve, called a *centrode*. Two centrodes can be constructed. In the first case when the mechanism is moved with the tibial link fixed, which results in the *tibial centrode* and in the other case the femoral link is fixed, from which the *femoral centrode* is the result. These two centrodes are given in *figure 4*, for the crossed version of the mechanism.

![Figure 4](image)

*Figure 4.* The corresponding points on the two centrodes.

It appears that these curves form a pure rolling contact. So along the curves corresponding points can be defined (one point of the tibial centrode will contact only one point of the femoral centrode). This is shown also in *figure 4*. With the help of these centrodes and the combinations of the corresponding points the curve of the femoral contact profile can be constructed.

### 7.4 Construction of the femoral contact profile.

From the given tibial profile a line perpendicular to this tibial profile is constructed (see *figure 5*). This line will intersect the tibial centrode, and will make an angle $\chi$ with the tangent to the centrode at the intersection point. The length of the line segment is $l$.

To maintain contact between the tibial and femoral contact profiles and to avoid pinching, the same combination of $\chi$ and $l$ also has to be found on the femoral contact profile. Since it is known which point on the femoral centrode corresponds with the intersection point on the tibial centrode, a line with length $l$ and an angle $\chi$ to the tangent of the femoral centrode can be drawn.
In this way a point of the femoral contact profile is identified (see figure 5). When the same procedure is followed for a sufficient number of points the whole femoral contact curve can be constructed.

This procedure will be translated into a mathematical model, which is needed to be able to write the software program.
8. Mathematical model

8.1 The centrodes

8.1.1 Definitions

In this analysis a planar four bar linkage \( OABP \) shall be considered (see Figure 6).

\[
\begin{align*}
\text{1. uncrossed mechanism} & \quad \text{2. crossed mechanism} \\
\vec{v}_1 &= I_1 \cos(\varphi) \vec{e}_1 + I_1 \sin(\varphi) \vec{e}_2 \\
\vec{v}_2 &= I_2 \cos(\theta) \vec{e}_1 + I_2 \sin(\theta) \vec{e}_2 \\
\vec{v}_3 &= I_3 \cos(\alpha) \vec{e}_1 + I_3 \sin(\alpha) \vec{e}_2
\end{align*}
\]

Figure 6. The two types of mechanisms.

\( l_1, l_2, l_3 \) and \( l_4 \) are the lengths of the four links, and the angle \( \alpha \) is the input. first the angles \( \varphi \) and \( \theta \) will be calculated as a function of \( \alpha \). With the definitions as shown both the uncrossed and the crossed version can be described with the same formulae. The only difference will be the range of the angle \( \alpha \). In the uncrossed version \( \alpha \) will vary between 50° and -90°, and in the crossed version the range will be from 230° to 90°. It is assumed here that for the chosen lengths these two ranges are possible. Later on in the software program this will be checked and if necessary the range will be adapted. Now the following vectors are defined:

\[
\begin{align*}
\vec{v}_1 &= I_1 \cos(\varphi) \vec{e}_1 + I_1 \sin(\varphi) \vec{e}_2 \\
\vec{v}_2 &= I_2 \cos(\theta) \vec{e}_1 + I_2 \sin(\theta) \vec{e}_2 \\
\vec{v}_3 &= I_3 \cos(\alpha) \vec{e}_1 + I_3 \sin(\alpha) \vec{e}_2
\end{align*}
\]
\[ \vec{r}_4 = \vec{l}_4 \vec{e}_1 \] (4)

*Figure 6* shows that the following equation is valid in any position:

\[ \vec{r}_1 + \vec{r}_3 = \vec{r}_2 + \vec{r}_4 \] (5)

Using (5) \( \varphi(\alpha) \) and \( \theta(\alpha) \) can be calculated. \( \varphi(\alpha) \) will be calculated first:

### 8.1.1.1 The angle \( \varphi(\alpha) \)

The lengths of the links are fixed, so

\[ (\vec{r}_1 + \vec{r}_3 - \vec{r}_4) \cdot (\vec{r}_1 + \vec{r}_3 - \vec{r}_4) = l_2^2 \] (6)

Substitution of (1), (3) and (4) then yields:

\[ (l_1 \cos \varphi + l_3 \cos(\alpha) - l_4)^2 + (l_1 \sin \varphi + l_3 \sin \alpha)^2 = l_2^2 \] (7)

After some simplifications this can be written as:

\[ CA_{\varphi} \sin \varphi + CB_{\varphi} \cos \varphi = CC_{\varphi} \] (8)

where

\[ CA_{\varphi} = \sin \alpha \quad CB_{\varphi} = \cos \alpha - \frac{l_4}{l_3} \quad CC_{\varphi} = \frac{l_4}{l_1} \cos \alpha + \frac{-l_1^2 + l_2^2 - l_3^2 - l_4^2}{2l_1 l_3} \] (9)

Now equation (8) has to be solved. Such a solution is obtained by expressing \( \sin \varphi \) and \( \cos \varphi \) in terms of \( \tan(\varphi/2) \),

\[ \sin \varphi = \frac{2 \tan(\frac{\varphi}{2})}{1 + \tan^2(\frac{\varphi}{2})} \quad \cos \varphi = \frac{1 - \tan^2(\frac{\varphi}{2})}{1 + \tan^2(\frac{\varphi}{2})} \] (10)

Substitution and simplification then yields:

\[ \tan(\frac{\varphi}{2}) = \frac{CA_{\varphi} \pm \sqrt{CA_{\varphi}^2 + CB_{\varphi}^2 - CC_{\varphi}^2}}{CB_{\varphi} + CC_{\varphi}} \] (11)
So two distinct values of $\phi$ are found:

$$\phi^+(\alpha) = 2\arctan \frac{CA_\phi + \sqrt{CA^2_\phi + CB^2_\phi - CC^2_\phi}}{CB_\phi + CC_\phi}$$  \hspace{1cm} (12)$$

$$\phi^-(\alpha) = 2\arctan \frac{CA_\phi - \sqrt{CA^2_\phi + CB^2_\phi - CC^2_\phi}}{CB_\phi + CC_\phi}$$  \hspace{1cm} (13)$$

In section 8.1.2 these results will be discussed.

\subsection*{8.1.1.2 The angle $\theta(\alpha)$}

For the calculation of $\theta(\alpha)$ the following equation can be used:

$$\langle l_2 - l_3 + l_4, l_2 - l_3 + l_4 \rangle = l_1^2$$  \hspace{1cm} (14)$$

When the same procedure as in (7) until (13) is followed, this results in:

$$\theta^+(\alpha) = 2\arctan \frac{CA_0 + \sqrt{CA_0^2 + CB_0^2 - CC_0^2}}{CB_0 + CC_0}$$  \hspace{1cm} (15)$$

$$\theta^-(\alpha) = 2\arctan \frac{CA_0 - \sqrt{CA_0^2 + CB_0^2 - CC_0^2}}{CB_0 + CC_0}$$  \hspace{1cm} (16)$$

where

$$CA_0 = \sin \alpha \quad CB_0 = \cos \alpha - \frac{l_4}{l_3} \quad CC_0 = \frac{l_4}{l_2} \cos \alpha - \frac{l_1^2 - l_2^2 - l_3^2 - l_4^2}{2l_2l_3}$$  \hspace{1cm} (17)$$

How these results can be interpreted and how they can be used, will be explained in the next section.
8.1.2 Interpretation of results

As shown in the previous section, there are, given an angle $\alpha$, two solutions for $\varphi$ and $\theta$. What this means for the possible forms of the mechanisms is shown in figure 7. With the help of this figure it is easy to see that the solutions which have any physical meaning are $\varphi'(\alpha)$ and $\theta'(\alpha)$. These two solutions are the positive angles, since in both formulae (12) and (13) as in formulae (15) and (16) the terms $CB + CC$ appear to be negative. So these two solutions will be used, and $\alpha$ will vary between 50° and -90° in case of the uncrossed mechanism, and between 230° and 90° in case of the crossed mechanism, as already mentioned in section 7.1.

1. Uncrossed mechanism
2. Crossed mechanism

*Figure 7.* The possible solutions for $z$ and $h$.

8.1.3 The tibial centrode

In order to find the position of the IC of the tibial and femoral links it is necessary to calculate the intersection point of the vectors $l_1$ and $l_2$ (the ligaments). The unit vectors of $l_1$ and $l_2$ need to be determined for this purpose.
Construction of a point of the tibial centrode.

This results in (see also figure 8):

\[ \vec{n}_A = \cos(\varphi) \vec{e}_1 + \sin(\varphi) \vec{e}_2 \]
\[ \vec{n}_B = \cos(\theta) \vec{e}_1 + \sin(\theta) \vec{e}_2 \]  \hspace{1cm} (18)

The intersection point now is defined by:

\[ C_1 \vec{n}_A = I_4 + C_2 \vec{n}_B \]  \hspace{1cm} (19)

To find the coordinates of the intersection point it is sufficient to calculate only \( C_1 \). After substitution of (18) in (19) and some simplifications, this results in:

\[ C_1 = \frac{L \sin(\theta)}{\sin(\theta - \varphi)} \]  \hspace{1cm} (20)

so

\[ TC = \frac{L \sin(\theta)}{\sin(\theta - \varphi)} (\cos(\varphi) \vec{e}_1 + \sin(\varphi) \vec{e}_2) \]  \hspace{1cm} (21)

Hereby two assumptions are made. The first one is that \( \varphi \) and \( \theta \) will not become equal to 0° or 180°. The second assumption is that \( \varphi \) and \( \theta \) do not become equal to each other. In case of realistic mechanisms these assumptions are fair.

Now the femoral centrode will be calculated, using the found tibial centrode. Here again the calculations are made for the uncrossed mechanism, and the formulae which will be found are valid also for the crossed mechanism (again only the range is rotated over 180°).
8.1.4 The femoral centrode

The components of the vector $l_1''$ with length $l_1''$, which points from A to FC (see Figure 9c), now have to be calculated. These components can be calculated as follows:

When the tibial centrode was calculated, the tibial link was fixed. From this position the angle $\alpha$ was changed and the angles $\varphi$ and $\theta$ were calculated. Now the femoral centrode is calculated by fixing the femoral centrode. The angle $\alpha$ in the input situation now will be called $\alpha_s$. When the mechanism is moved into different positions with the tibial link fixed, an angle $\alpha$ different from $\alpha_s$ can be found (see Figure 9a). But since the femoral link was to be fixed, this position has to be translated and rotated so, that the tibial link is again in the input situation (see Figure 9b). In this way an angle $\beta$ is found.

Figure 9c: Construction of a point of the femoral centrode.
Now it is clear that:

$$\beta = \alpha - \alpha_s$$

(22)

Now angle $\delta$ is defined by:

$$\delta = \varphi - \beta = \varphi + \alpha_s$$

(23)

Furthermore $l_1''$ is determined by:

$$l_1'' = |l_1 - C_1|$$

(24)

The position of $FC$ now can be calculated with:

$$FC = A + a'/e_1 + b'/e_2$$

(25)

Where:

$$a' = l_1'' \cos(\delta + R) \quad b' = l_1'' \sin(\delta + R)$$

(26)

Note that in this last formula $R$ is equal to zero in case of the uncrossed mechanism and equal to $\pi$ in case of the crossed mechanism!! Now that both the tibial and femoral centrode are known, the femoral contact profile can be constructed. This will be worked out in section 8.2.
8.2 The femoral contact profile

8.2.1 The distance \( l \)

Since the two centrodes have been constructed in the previous sections, the curve of the femoral contact profile now also can be constructed. To this aim the combination \((a,t)\) has to be found, according to the description given in section 7. The length \( l \) is calculated by using the next procedure (see also figure 10):

\[
\text{Figure 10. construction of unit vector } n, \text{ and distance } l
\]

Given the profile in figure 10, the position vector \( r_t \) is defined by:

\[
\vec{r}_t = x\vec{e}_1 + y(x)\vec{e}_2
\]  \( (27) \)

Differentiation of the position vector \( r_t \) with respect to \( x \) yields:

\[
\frac{d\vec{r}_t}{dx} = \vec{e}_1 + y\vec{e}_2
\]  \( (28) \)

\( t_t \) is the unit tangent vector to the tibial contact profile at position \( r_t \), so:

\[
\vec{t}_t(x) = \frac{d\vec{r}_t}{dx} = \frac{1}{\sqrt{1 + y_t^2}}(\vec{e}_1 + y_t\vec{e}_2)
\]  \( (29) \)
The vector \( n \), normal to \( t \), now can be found by:

\[
\vec{n}(x) = \frac{\vec{d}t}{\| \vec{d}t \|} \quad ; \quad |\vec{n}(x)| = 1
\]  

(30)

Working this out yields:

\[
\vec{n}(x) = \frac{1}{\sqrt{1+y_t^2}} (-y_t \vec{e}_1 + \vec{e}_2)
\]  

(31)

Since the tibial centrode already is calculated, the vector \( r_f \) is known. Now vector \( r_t \) and \( l \) have to be found, such that:

\[
\vec{r}_t + l\vec{n} = \vec{r}_f
\]  

(32)

where:

\[
\vec{r}_t = x_t \vec{e}_1 + y_t(x_t) \vec{e}_2
\]  

(33)

and:

\[
\vec{r}_f = x_f \vec{e}_1 + y_f \vec{e}_2
\]  

(34)

Substitution of (31), (33) and (34) in (32) yields:

\[
x_t \vec{e}_1 + y_t(x_t) \vec{e}_2 + \frac{l}{\sqrt{1+y_t^2}} (-y_t \vec{e}_1 + \vec{e}_2) = x_f \vec{e}_1 + y_f \vec{e}_2
\]  

(35)

Now (35) can be divided into two components:

\[
\vec{e}_1: \quad x_t - \frac{ly_t}{\sqrt{1+y_t^2}} = x_f
\]  

(36)

\[
\vec{e}_2: \quad y_t(x_t) + \frac{l}{\sqrt{1+y_t^2}} = y_f
\]  

(37)
In these equations \( y_i' \) can be expressed in terms of \( x_i \). Herewith there are two equations with two unknowns \((x_i, l)\). So the variable \( l \) can be found. These results now will be used for different types of tibial contact profiles.

### 8.2.1.1 Tibial contact profile is a straight line:

A straight line is described as in the next equation:

\[
y_i = ax_i + h
\]  

(38)

Differentiation of (38) with respect to \( x \) yields:

\[
y'_i = a
\]

(39)

Substitution of (38) and (39) in (36) and (37) results in:

\[
x_i = \frac{-la}{\sqrt{1+a^2}} = x_f
\]

(40)

and

\[
a x_i + h + \frac{l}{\sqrt{1+a^2}} = y_f
\]

(41)

The next step is to solve \( l \) and \( x_i \) from (40) and (41). This yields:

\[
x_i = \frac{ay_f - ah + x_f}{1+a^2} \quad l = \frac{y_f - ax_f - h}{\sqrt{1+a^2}}
\]

(42)

A special case is a straight horizontal line \((a=0)\). Then the results are:

\[
x_i = x_f \quad l = y_f - h
\]

(43)
8.2.1.2 Tibial contact profile is a parabola:

A parabola is a second order polynomial and can be described by:

\[ y_i = a(x_i - x_0)^2 + h \]  

(44)

\((x_0, h)\) is the displacement from the origin \(O(0,0)\), and \(a\) determines whether the curve is convex or concave, and also the convexity or concavity. Differentiation of (44) with respect to \(x\) yields:

\[ y'_i = 2a(x_i - x_0) \]  

(45)

In order to be able to solve this problem, first \(l\) has to be eliminated from (38) and (39). This yields:

\[ x_i - y_i(y_f - y_i(x_i)) = x_f \]  

(46)

Substitution of (44) and (45) in (46) yields:

\[ 2a^2x_i^3 - 6a^2x_0x_i + (1 - 2ay_f + 2ah + 6a^2x_0^2)x_i - 2ahx_0 + 2ax_0y_f - 2a^2x_0^3 - x_f = 0 \]  

(47)

The same equation can be represented as:

\[ A_0x_i^3 + A_1x_i^2 + A_2x_i + A_3 = 0 \]  

(48)

with:

\[ A_3 = 2ax_0y_f - 2ahx_0 - 2a^2x_0^3 - x_f \quad A_2 = 1 - 2ay_f + 2ah + 6a^2x_0^2 \]  

\[ A_1 = -6a^2x_0 \quad A_0 = 2a^2 \]  

(49)

(50)

This is a cubic equation which has to be solved numerically. How this will be done, will be discussed in section 9.3.3. Once \(x_i\) is known, \(l\) can be calculated with:

\[ l = (y_f - a(x_i - x_0)^2 - h)\sqrt{4a^2(x_i - x_0)^2 + 1} \]  

(51)
8.2.2 The angle $\chi_i$

At this point of the mathematical analysis it is necessary to switch over to a discretized formulation. So from here on we will also use the index $i$ at some of the variables. These variables however are subject to the same relations as before!!

Before $\chi_i$, which is the angle between the vector $n_{ui}$ and the tangent of the tibial centrode, can be calculated, the vector $c_{ti}$ tangent to the tibial centrode has to be found. It is possible to find an analytical solution for this problem, but this solution will be very complex and will take a lot of calculation time. So a rather simple but satisfactory approximation is chosen (see also figure 11):

*Figure 11. Approximation of vector $c_{ti}$ tangent to tibial centrode.*

A number of points on the tibial centrode are known, and since the curve from the tibial centrode is rather smooth, a good approximation is found by using the tangent of the line through the point $TC_{i-1}$ and the point $TC_{i+1}$.

This results in:

$$\vec{e}_{ti}=\vec{T}_{fi+1}-\vec{T}_{fi-1}=(TC_{xi+1}-TC_{xi-1})\vec{e}_1+(TC_{yi+1}-TC_{yi-1})\vec{e}_2$$

(52)

Now $\chi_i$ is found by taking the dot product of (28) and (48):

$$\vec{e}_{ti}\cdot\vec{n}_{vs}=|\vec{e}_{ti}||\vec{n}_{vs}||\cos(\pi - \chi_i)$$

(53)

This results in:
8.2.3 The construction of the femoral contact profile

Before the femoral profile can be constructed, an approximation of the unit vector tangent to the femoral centrode, $c_{fi}$, has to be made. Hereto the same approximation as in equation (52) is used. This results in:

$$\chi_i = \pi - \arccos \frac{(\vec{e}_{ti}, \vec{e}_{ci})}{|\vec{e}_{ti}| |\vec{e}_{ci}|}$$  \hspace{1cm} (54)

A point $F_i$ from the femoral contact profile now can be found by rotating the unit vector $c_{fi}$ over an angle $\chi_i$, multiplication of the found vector with $l$, and finally addition of vector $FC_i$. The rotation is done by multiplying the vector with a rotation matrix $R_i$. See figure 12.

$$\bar{e}_{fi} = \frac{1}{S_i} (FC_{i+1} - FC_{i-1}) = \frac{1}{S_i} ((FC_{x,i+1} - FC_{x,i-1})\vec{e}_1 + (FC_{y,i+1} - FC_{y,i-1})\vec{e}_2); |\bar{e}_{fi}| = 1$$  \hspace{1cm} (55)

Where:

$$S_i = \sqrt{(FC_{x,i+1} - FC_{x,i-1})^2 + (FC_{y,i+1} - FC_{y,i-1})^2}$$  \hspace{1cm} (56)

Figure 12. Determination of a point of the femoral contact profile by rotation and multiplication.
This then results in:

\[ \tilde{F}_1 = \tilde{F}_{C_1} + l_i R_i \tilde{e}_{fi} = F_{x1} \tilde{e}_1 + F_{y1} \tilde{e}_2 \]  

(57)

When this procedure of calculating the coordinates of a point of the femoral contact profile is done for a sufficient number of points, a good indication of the contact profile is given by drawing straight lines between the points. Curve fitting is also possible, but may cost a lot of calculation time. This "visualisation of results" is one of the aspects, that will be discussed in the following section.
9. The programming

9.1 Program structure

Using the mathematical model a software program has been written. The program consists of six main parts. These are:

1. Input
2. Calculation
3. Output
4. Parameter variation
5. Slip ratio
6. Movie

Each part of this structure is written as a subroutine, which is called from the main program. All subroutines may use other subroutines, for example from a library. Figure 13 shows this.

Figure 13. The structure of the software program: main program and subroutines.
For further information on the software used, see Chapter 12, Literature.

Now the routines used will be specified. It is recommended here to keep the program text (see Appendices A to I) aside. This is because the program text contains a lot of comment lines, which can be very useful in understanding the structure of the program.
9.2 The main program

The main program calls the subroutines needed to perform the operations requested by the user. These operations are:

1. Input of a mechanism (after input the centroids and contact profiles are calculated automatically).
2. Plotting of results.
3. Parameter variation.
4. Slip ratio.
5. Movie.
6. Quit the program.

9.3 The subroutines

As mentioned, the program consists of a number of subroutines. Each of them performs a specified task, and is called from the main program. In this way the program is well structured, and easy to understand.

From each subroutine now a short description will be given.

9.3.1 Subroutine INPUT

This subroutine supplies the program with the input parameters. These are:

1. Is the mechanism crossed or uncrossed?
2. The coordinates of A and B.
3. Type of tibial contact profile (straight line or parabola).
4. Parameters for the tibial contact profile.
5. The number of increments.
6. The choice for the dimensions of a perspex model which is available (no further input has to be given). This model will also be used for the validation of the program.
7. The choice for realistic anatomic dimensions which will be specified later on in the user's manual.
9.3.2 Subroutine CENTRO

In this subroutine the tibial and the femoral centrode are calculated. First the lengths of the links have to be calculated. This is done by:

\[ l_1 = \sqrt{x_A^2 + y_A^2} \]

\[ l_2 = \sqrt{(x_B - x_P)^2 + y_B^2} \]

\[ l_3 = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \]

\[ l_4 = x_P \]

Now the lengths of the links are known, \( \varphi \) and \( \Theta \) as a function of \( \alpha \) can be calculated, by using equations (12) and (15) (section 8.1.1). As mentioned before, for \( \alpha \) a range from 50° to -90° (for the uncrossed version; for the crossed version the range will be 230° to 90°) will be chosen. This range has to be divided into the chosen number of increments. This is done with the following equation:

\[ \alpha = \frac{50\pi}{180} + R - \frac{I - 1}{N - 1} \frac{140\pi}{180} \]

In this equation \( R \) is equal to zero if the mechanism is an uncrossed one, and equal to \( \pi \) when the mechanism is a crossed one. \( I \) is the \( I^{th} \) increments and \( N \) is the total number of increments.

Next the vector \( TC \), which represents the tibial centrode and vector \( FC \), which represents the femoral centrode, are determined. It may very well be possible that the mechanism cannot make the full range of 140° of flexion. This is caused by the fact that the mechanism may change from the crossed version into the uncrossed version or the other way around. Then a so called Tail appears (see Figure 14). The curves then will be of such a form that moving the mechanism becomes impossible. When this happens the program will give a warning, and it will also give the range that can be covered.
Figure 14: The problem that appears when the mechanism changes from the crossed to the uncrossed version. The program will detect the beginning of the "Tail" and stop further calculations of the centrode.

9.3.3 Subroutine FEMORA

This subroutine calculates the femoral contact profile. It consists of two parts. One of them calculates the femoral contact profile in case the tibial contact profile is a straight line and the other in case the tibial contact profile is a parabole:

when the tibial contact profile is a straight line the mathematical analysis (section 8) together with the program text (see Appendices) give enough information. When the tibial contact profile is a parabola however, some more explanation is necessary.

In section 8.2.1.2 it was found that a cubic equation had to be solved. To be able to solve this equation a routine from the NAG-library is used. This routine is called C02AGF. This routine will give three solutions. For this problem only one of these solutions is of any interest. This particular solution has to be selected.

In the first place it is checked whether or not there are any complex solutions. If so, there is only one real solution, and this has to be the right solution. But when there are three real solutions (two real solutions is impossible since the complex solutions always appear in conjugates) the right solution has to be selected. The right solution then is the one which is nearest to the previous solution. This means that for the first point already a solution has to be known. It is known that when \( x = x_0 \) the line searched is a vertical one. So then the solution of the polynome also has to be \( x_0 \). In the program that point of the array of points of the tibial centrode is selected from which the x-coordinate is the closest to \( x_0 \). For this "starting point" the polynome is solved, and the solution which is nearest to \( x_0 \) is chosen. For the next points the same procedure can be repeated. The consequence of this method is that the femoral contact profile has to be calculated in two parts. One part is calculated from the starting point back-
The programming

wards until \( I=1 \), and the other part is calculated from the starting point forwards, until \( I=N \).

The solution however gives some problems. When a mechanism is chosen it may happen that there are points with one real solution for the cubic equation, and also points with three real solutions. The method used here then may have some difficulties in finding the right solution. Here also a warning is given. This effect can be eliminated by changing either the convexity/concavity \( a \) of the parabola (the parabola will have to be flatter, this means \( a \) has to be closer to zero) or the vertical displacement \( h \) (\( h \) will have to be larger).

This whole solution of the cubic equation has been worked out in the subroutine CALC, which is not mentioned separately, since it is only a part of subroutine FEMORA.

Subroutine FEMORA now will calculate a point of the femoral contact profile by rotation and multiplication of the vector tangent to the femoral centrode.

9.3.4 Subroutine OUTPUT

This subroutine visualises the results from the calculations. It will draw the mechanism, according to the given coordinates of points A and B. Next it will draw the centrodes and contact profiles. In case of parameter variation the centrodes and contact profiles are drawn three times, once for the initial parameter values, once with the chosen parameter larger and once with the chosen parameter smaller (values are mentioned in the user's manual). The three cases will be drawn in different linestyles. The curves consist of straight lines between the calculated points. Curve fitting was not chosen, because this would take a lot of extra calculation time. Furthermore, the program will show that when a sufficient number of increments is chosen, the curves appear to be smooth enough.

9.3.5 Subroutine PARA

In this last part of the program the opportunity is given to change some variables. The following changes are possible:

1. A change in one of the coordinates of points A and B (x- or y-coordinate). The program then will calculate the effects on the femoral contact profile when the chosen coordinate is made a certain value smaller and larger. This value is mentioned in the user's manual.
The programming

2. A change in the curve of the tibial contact profile. The effects of a little more and a little less concave or convex curve on the femoral contact profile will be shown. A vertical or horizontal translation of the curve also is possible (again see the user's manual for the exact values).

3. A third possibility is to make one or more ligaments slightly elastic. At this point it has to be concluded that there is no time left for this part of the problem.

After the parameter to be varied has been chosen, all calculations are done twice again, once with the chosen parameter is a certain value smaller and once with the chosen parameter is a certain value larger (see user's manual for exact values).

9.3.6 Subroutine SLIP

This subroutine calculates the slip ratio, which is defined as the ratio of the length-segments of the two contact profiles over an interval of the flexion angle $\gamma$.

The slip ratio can be defined in a more mathematical way as follows:

A segment of the tibial contact profile is defined by:

$$dl_{tibial} = \sqrt{(T_{x,i+1} - T_{x,i})^2 + (T_{y,i+1} - T_{y,i})^2}$$

The length of the corresponding segment of the femoral contact profile is defined by:

$$dl_{femoral} = \sqrt{(F_{x,i+1} - F_{x,i})^2 + (F_{y,i+1} - F_{y,i})^2}$$

The slip ratio is now calculated by:

$$Slip = \frac{dl_{femoral}}{dl_{tibial}}$$

Also the average slip ratio is determined by calculating the ratio of the total lengths of both the contact profiles. The total lengths of the profiles are calculated by summation of $dl_{tibial}$ and $dl_{femoral}$.

The results of these calculations are represented in a diagram where the ratio is set out against the flexion angle $\gamma$. 

9.3.7 Subroutine MOVIE

This last subroutine plots the mechanism and the four curves in several different positions. This routine gives a clear view of the whole process. It will play an important role in the model validation that is described in the next section.
10. Model validation

After the program text was entered and compiled (for the used equipment and software, see Lit. [6] and [7]) the program works satisfactorily. This fact however does not guarantee that the results are valid. Therefore some tests are performed. These tests will be described in this section.

10.1 Conditions resulting from the model

There must be one contact point between the tibial centrode and the femoral centrode. This contact point has to be at the intersection point of the Anterior Cruciate Ligament (ACL) and the Posterior Cruciate Ligament (PCL) of the mechanism in the position as given in the input. In the same way there must be one contact point between the two contact profiles. The tangent line of the contact point of the contact profiles has to intersect the contact point of the centrodes.

When the tibial contact profile is chosen as a straight line, it is easy to approximate the angle between the vector \( n \) and the tibial centrode. This angle has to be found again at the femoral centrode and the femoral contact profile.

When the tibial contact profile is a horizontal line, the vector \( n \) only has a vertical component, so each point of the femoral contact profile lies straight above the tibial contact profile. (In practice these last two conditions only can be validated for the first and the last point of the curves).

10.2 Symmetrical Mechanism

When a symmetrical mechanism is chosen, the calculated centrodes have to be symmetrical as well. In the same way the calculated femoral contact profile has to be symmetrical as the chosen tibial contact profile (and the mechanism) is symmetrical. Symmetrical in this case means symmetrical about the line \( x = 50 \) (the x-coordinate of \( P \) was chosen to be 100).

10.3 The Perspex Model

In this case the dimensions of a realisation in perspex of the model were used as input parameters for the program. This perspex exhibited all the kinematical characteristics belonging to its particular configuration. The program calculated the tibial and femoral centrodes and the femoral...
contact profile, which have to be in agreement with the perspex model.

10.4 Subroutine Movie

The most powerful possibility to validate the model however is the in section 9.3.6, named subroutine MOVIE. When the mechanism is drawn in several positions, a smooth movement of the curves should appear, where the curves form a rolling contact.

When the tests mentioned are performed on several mechanisms, it can be concluded that the model programmed is correct.
11. Conclusions and recommendations

The following conclusions and recommendations can be made.

1. The model used in the program calculates the correct centroids and contact profile.

2. It appeared to be impossible to satisfy all the demands given in the problem definition in section 4. The reason for this is that there was too little time. The program calculates the centroids and the femoral contact profile, and it is also possible to calculate the effects of a variation in one of the parameters.

3. The program certainly is not written in the most efficient way. First, many comment lines were used to make the program text better understandable. Second, some arrays are larger than necessary, so the program will use a rather large amount of memory. Third, the input needs to be correct. When, for example, a string is entered where a number is expected, the program is terminated, instead of asking the same question again. Finally, not all the equations are represented in the most accurate algebraic form. This however is of no significant meaning, since the calculations are accurate enough for the purpose which has to be served. All these restrictions are due to a lack of programming experience and knowledge.

4. The program runs faster than expected on a 80386 PC/AT with a 80387 coprocessor. The most complex calculations only take a few seconds, even when the maximum number of increments is chosen. Tests on other PC's showed that a 20286 with coprocessor is sufficient also, but a PC/XT is not usable, since it is too slow. Because the speed is high enough, one may consider to calculate the exact tangents of the centroids, where now an approximation is used. Validation of the model however showed that the results need not be more accurate than they already are.

5. The program is easy to use, because of the shell that is built around the calculations.

6. When a mechanism is used which has realistic dimensions it appears that when the links are rigid it is not possible to reach 140° flexion. This may be an indication that both the links and the contact profiles are in fact not rigid.

7. Instead of using a manual it may be useful to implement a help function which can be called at any place in the program.
Conclusions

8. The use of defaults may be of some help, when only some small changes are to be entered.

9. This report does not mention the worth of this program. When used in a practical some background information is needed for a better understanding.

10. The program can be extended easily because of the structure of a main program and subroutines.

11. It may be possible that some of the problems mentioned will be worked out after this report is completed. These changes may cause some difference in the program text (see Appendices). These changes will appear mainly in the main program, since extra functions will be written in the form of subroutines that are called from the main program.

Some of the problems that were mentioned in these conclusions may be joined in a new practical. This practical may also include the parts we were not able to solve. In particular the part where the links are not rigid anymore seems interesting. This problem however demands a different, somewhat more mechanical approach.
12. Literature

[1] Ian A.F. Stokes (Editor)  
Mechanical factors and the skeleton  
John Libbey & Company Limited, London  
ISBN 0-86196-006-8

Knee Ligaments; Structure, Function, Injury and Repair  
Raven Press, New York  

Kinematic Synthesis of Linkages  
McGraw-Hill series in MECHANICAL ENGINEERING  
McGraw-Hill Book company, 1964

Kinematica  
Scheltema & Holkema, Amsterdam, 1970  
ISBN 90-6060-502-0

A Model of the Human Knee, Derived from Kinematic Principles and Its Relevance for Endoprosthesis Design.  

[6] Software used:  
Wordperfect 5.1 NL  
Drawperfect 1.1 NL  
MS-Fortran 5.0 Compiler  
NAG Workstation Library  
NAG PC Graphics library  
Norton Classic Editor, version 1.5  

All software used was retrieved from the Calculation Center (Rekencentrum) of the Eindhoven University of Technology.

[7] Hardware used:  
Rembrant 386 AT with 80387 coprocessor  
VGA card with colour monitor  

This hardware was retrieved from the Department of Fundamental Research, Faculty of Mechanical Engineering of the Eindhoven University of Technology.
Appendices
Report number 92.013 B
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix A: Main Program

********************************************************************************
** THIS PROGRAM IS WRITTEN BY NIEK VAN NUNEN **
** AND FRANK PETERS **
********************************************************************************

PROGRAM KNEE
IMPLICIT NONE

********************************************************************************
** THIS IS THE MAIN PROGRAM FROM WHICH THE NECESSARY **
** SUBROUTINES WILL BE CALLED. **
********************************************************************************

********************************************************************************
** DECLARATION OF THE IDENTIFIERS **
********************************************************************************

INTEGER I,II,III,N(3), NMAX, J, ERROR, IFAIL, NOUT, INP
INTEGER NIN
PARAMETER (NMAX=200)
DOUBLE PRECISION R, SA(2,3), SB(2,3), A(3), H(3), XO(3)
DOUBLE PRECISION L1, L2, L3, L4
DOUBLE PRECISION TC(2, NMAX, 3), FC(2, NMAX, 3)
DOUBLE PRECISION T(2, NMAX, 3), F(2, NMAX, 3), L(NMAX,3)
DOUBLE PRECISION PHI(NMAX, 3)
LOGICAL CROSS, TIBIAL, VARIAT, MECH
PARAMETER (L4=100, NOUT=6)

I,II,III: COUNTER
N(J): NUMBER OF ITERATIONS
NMAX: MAXIMUM NUMBER OF ITERATIONS
J: COUNTER
ERROR: ERROR INDICATOR
IFAIL: ERROR INDICATOR
NOUT: NUMBER OF OUTPUT DEVICE
INP: INPUT VARIABLE
NIN: NUMBER OF INCREMENTS AT INPUT
R: R = 0: UNCROSSED SYSTEM
   R = PI: CROSSED SYSTEM
SA(2,J): X AND Y COORDINATE OF POINT A
SB(2,J): X AND Y COORDINATE OF POINT B
A(3): PARAMETER OF TIBIAL PROFILE
H(3): PARAMETER OF TIBIAL PROFILE
XO(3): PARAMETER OF PARABOLIC TIBIAL PROFILE
L1: LENGTH OF ANTERIOR CRUCIATE LIGAMENT
L2: LENGTH OF POSTERIOR CRUCIATE LIGAMENT
L3: LENGTH OF FEMORAL LINK
L4: LENGTH OF TIBIAL LINK
TC(K,L,M): COORDINATES OF TIBIAL CENTRODE
K: X AND Y COORDINATE.

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix A: Main Program

PRINT 2000, 'This program is written for students
PRINT 2000, ' of the Eindhoven University of Technology.
PRINT 2000, ' It's a software program for the simulation
PRINT 2000, ' of a two-dimensional model of the knee joint.
PRINT 2000, ' It's written to show students the effects
PRINT 2000, ' of parameter variation in a simulation program.
PRINT 2000, '
PRINT 2000, ' DO NOT USE STRINGS,
PRINT 2000, ' use only the numerical part of your keyboard

L: I-TH POINT
M: COUNTER FOR PARAMETER VARIATIONS
   M=1: ORIGINAL VALUES
   M=2,3: VARIED VALUES
FC(K,L,M): COORDINATES OF FEMORAL CENTRODE
T(K,L,M): COORDINATES OF TIBIAL CONTACT PROFILE
F(K,L,M): COORDINATES OF FEMORAL CONTACT PROFILE
L(L,M): DISTANCE CENTRODE CONTACT PROFILE
CROSS: IF CROSS IS TRUE: CROSSED SYSTEM
       IF CROSS IS FALSE:UNCROSSED SYSTEM
TIBIAL: IF TIBIAL IS TRUE: TIBIAL PROFILE IS A STRAIGHT
        LINE
       IF TIBIAL IS FALSE: TIBIAL PROFILE IS A PARABOLE
VARIAT: IF VARIAT IS TRUE: PARAMETER VARIATION
       IF VARIAT IS FALSE: NO PARAMETER VARIATION
MECH: IF MECH IS FALSE: NO MECHANISM CHOSEN YET
      IF MECH IS TRUE: MECHANISM CHOSEN YET

***************************************************************
** INITIALISATION OF THE GRAPHICAL MODE **
***************************************************************

CALL J06VAF(1, NOUT)
CALL XXXXXX
OPEN (NOUT, FILE='KNEE.ERR')
CALL J06WAF
CALL J06WBF(-25.0DO, 150.0DO, -25.0DO, 150.0DO ,1)
CALL J06YLF(5.0DO,0.0DO)
CALL J06YKF(5.0DO,7.0DO)

PRINT 1000
PRINT 2000, 'The design of a software program for the simulation of a 2 dimensional model of the knee joint.

3
Appendix A: Main Program

PRINT 1000
PRINT *, 'Press ENTER to continue'
CALL J06WDF

*******************************************************************************
** INITIALISATION OF VARIABLE **
*******************************************************************************
MECH = .FALSE.

*******************************************************************************
** MAIN MENU **
*******************************************************************************

CALL CLEAR

PRINT *, 'Enter number of your choice'
READ *, INP
CALL CLEAR
IF (INP.LT.1.OR.INP.GT.6) THEN
  CALL CLEAR
  GOTO 5
ENDIF
IF (INP.EQ.1) THEN
  CALL FOR SUBROUTINE INPUT
  ..................
  (RE)SET THE VARIABLES
  ........................................
  MECH = .TRUE.
  VARIAT = .FALSE.
  DO 4 I = 1,3
     DO 3 II = 1,NMAX
        DO 2 III = 1,2
           TC(III,II,I) = 0.0D0
           FC(III,II,I) = 0.0D0
           T(III,II,I) = 0.0D0

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
ELSE

**QUIT THE PROGRAM**

PRINT *, 'Press ENTER'
CALL JO6WZF
STOP 'END OF THE PROGRAM'
ENDIF

C 1000 FORMAT (/,,/,,,/,),
2000 FORMAT (15X, 51A)
3000 FORMAT (/,,/,,,/)
4000 FORMAT (20X, 40A)
5000 FORMAT (/,,/,,,/,,,/,)
6000 FORMAT (25X, 38A)
C END

C

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix B: Subroutine INPUT

SUBROUTINE INPUT(N, NMAX, J, R, SA, SB, A, H, XO, CROSS, TIBIAL)

************************************************************************************
** THIS SUBROUTINE SUPPLIES THE OTHER ROUTINES WITH THE **
** INPUT DATA AND GIVES THE POSSIBILITY TO QUIT **
************************************************************************************

************************************************************************************
** DECLARATION OF THE GLOBAL IDENTIFIERS **
************************************************************************************

INTEGER N(3), NMAX, J
DOUBLE PRECISION R, SA(2,3), SB(2,3), A(3), H(3), XO(3)
LOGICAL CROSS, TIBIAL

************************************************************************************
** DECLARATION OF THE LOCAL IDENTIFIERS **
************************************************************************************

INTEGER UU

UU: INPUT VARIABLE

************************************************************************************
** CHOICE FOR GIVEN OR SELF MADE MECHANISM **
************************************************************************************

1) Anatomic (= realistic) dimensions
2) Perspex model
3) Self made mechanism

PRINT *, ' ', TYPE OF MECHANISM ',
PRINT *, ',
PRINT *, ',
PRINT *, ',
PRINT *, ',
PRINT *, ',
PRINT *, ',
PRINT *, ',
PRINT *, ', Enter number of your choice'
READ *, UU
IF(UU.LT.1.0.OR.UU.GT.3) THEN
   CALL CLEAR
   GOTO 1
ENDIF
IF(UU.EQ.1) THEN
   CROSS = .TRUE.
   R = 3.1415925
   SA(1,J) = 61.0D0
   SA(2,J) = 77.0D0
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix B: Subroutine INPUT

SB(1,J) = 20.0D0
SB(2,J) = 68.0D0
TIBIAL = .FALSE.
A(J) = -0.005D0
X0(J) = 50.0D0
H(J) = 7.0D0
N(1) = 200
GOTO 5

ELSEIF (UU.EQ.2) THEN

** PARAMETERS FOR PERSPEX MODEL **

CROSS = .TRUE.
R = 3.1415925
SA(1,J) = 91.0D0
SA(2,J) = 78.0D0
SB(1,J) = 42.0D0
SB(2,J) = 78.0D0
TIBIAL = .TRUE.
A(J) = 0.0D0
H(J) = 7.0D0
N(1) = 200
GOTO 5

ENDIF

** SELF MADE MECHANISM **

CALL CLEAR
PRINT *, 'IS THE MECHANISM:
PRINT *, 1) A crossed one
PRINT *, 2) An uncrossed one
PRINT *, 'Enter number of your choice'
READ *, UU
IF(UU.LT.1.OR.UU.GT.2) GOTO 2
CROSS = .FALSE.
IF(UU.EQ.1) CROSS = .TRUE.

CALL CLEAR
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
CALL J06YAF(0.0D0, 0.0D0)
CALL J06YCF(0.0D0, 20.0D0)
CALL J06YCF(2.0D0, 17.0D0)
CALL J06YCF(-2.0D0, 17.0D0)
CALL J06YAF(-10.0D0, 20.0D0)
CALL J06YHF('Y', 1)

******************************************************************************
** INPUT OF COORDINATES OF A AND B **
******************************************************************************

PRINT *,' Give the coordinates of hinges A and B'
PRINT *,' X-coordinate of A:'
READ *, SA(1,J)
PRINT *,' Y-coordinate of A:'
READ *, SA(2,J)
PRINT *,' X-coordinate of B:'
READ *, SB(1,J)
PRINT *,' Y-coordinate of B:'
READ *, SB(2,J)

******************************************************************************
** CHOICE AND INPUT OF TIBIAL CONTACT PROFILE **
******************************************************************************

CALL CLEAR
PRINT *,' THE TIBIAL PROFILE
PRINT *,' 1) is a straight line
PRINT *,' 2) is a parabola
PRINT *,' Enter number of your choice'
READ *, UU
IF(UU.LT.1.OR.UU.GT.2) GOTO 3
TIBIAL = .FALSE.
IF(UU.EQ.1) TIBIAL = .TRUE.

CALL CLEAR
IF (TIBIAL) THEN
PRINT *,' A straight line is described by
PRINT *,' \( Y = A*X + H \)
READ *, A(J)
PRINT *,' GIVE H (-50 LE H LE 50):'
READ *, H(J)
ELSE

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix B: Subroutine INPUT

```
PRINT *, 'A parabole is described by
" Y = A(X-X0)^2 + H "
PRINT *
PRINT *, 'GIVE A (-0.01 ≤ A ≤ 0.01):
READ *, A(J)
PRINT *, 'GIVE X0 (0 ≤ X0 ≤ 100):
READ *, X0(J)
PRINT *, 'GIVE H (-50 ≤ H ≤ 50):
READ *, H(J)
ENDIF
CALL CLEAR

******************************************************************************
** INPUT OF NUMBER OF INCREMENTS (= ACCURACY) **
******************************************************************************
4 PRINT *, 'Give the number of increments (10 - 200)
READ *, N(J)
IF(N(J).GT.NMAX) GOTO 4

END
```

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix C: Subroutine CENTRO

SUBROUTINE CENTRO(N, NMAX, J, R, SA, SB, *
    L1, L2, L3, L4, TC, FC, PHI)

******************************************************************************
** THIS SUBROUTINE CALCULATES THE TIBIAL AND FEMORAL CENTRODE. **
** THE X AND Y COORDINATES ARE STORED IN THE ARRAYS TC AND FC. **
******************************************************************************

******************************************************************************
** DECLARATION OF THE GLOBAL PARAMETERS **
******************************************************************************

INTEGER N(3), NMAX, J
DOUBLE PRECISION SA(2,3), SB(2,3), R
DOUBLE PRECISION L1, L2, L3, L4, TC(2,NMAX,3), FC(2,NMAX,3)
DOUBLE PRECISION PHI(NMAX,3)

******************************************************************************
** DECLARATION OF THE LOCAL PARAMETERS **
******************************************************************************

INTEGER I, NNMAX
PARAMETER (NNMAX = 200)
DOUBLE PRECISION ALPHA(NNMAX), THETA(NNMAX), SALPHA
DOUBLE PRECISION CAPHI, CBPHI, CCPHI, TEK, CATHET, GAMMA
DOUBLE PRECISION CBTHET, CCTHET, C1, BETA, LIAA, DELTA, PI
PARAMETER (PI=3.14159265)

I: COUNTER
NNMAX: MAXIMAL NUMBER OF INCREMENTS
ALPHA(NNMAX): ANGLE BETWEEN TIBIAL LINK AND FEMORAL LINK
SALPHA: ANGLE ALPHA AT THE GIVEN POSITION (RADIANS)
PHI(NNMAX): ANGLE BETWEEN TIBIAL LINK AND O-A
THETA(NNMAX): ANGLE BETWEEN TIBIAL LINK AND P-B
CAPHI: VARIABLE
CBPHI: VARIABLE
CCPHI: VARIABLE
TEK: VARIABLE
CATHET: VARIABLE
GAMMA: VARIABLE
CBTHET: VARIABLE
CCTHET: VARIABLE
C1: DISTANCE POINT O TO TIBIAL CENTRODE
BETA: ANGLE ALPHA - SALPHA
LIAA: DISTANCE FROM A TO FC
DELTA: ANGLE PHI - BETA
PI: 3.14159265

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
** CALCULATION OF THE ANGLE SALPHA **

\[
SALPHA = R + \tan \left( \frac{(SB(2,J)-SA(2,J))/(SB(1,J)-SA(1,J))}{1} \right)
\]

** CALCULATION OF THE LENGTHS OF THE LINKS **

\[
L1 = \sqrt{SA(1,J)\times SA(1,J) + SA(2,J)\times SA(2,J)}
\]

\[
L2 = \sqrt{(L4-SB(1,J))\times (L4-SB(1,J)) + SB(2,J)\times SB(2,J)}
\]

\[
L3 = \sqrt{(SA(1,J)-SB(1,J))\times (SA(1,J)-SB(1,J)) + (SB(2,J)-SB(2,J))\times (SA(2,J)-SB(2,J))}
\]

\[
I = 0
\]

I = I + 1

** CALCULATION OF THE ANGLE ALPHA (IN RADIANS) **

\[
\text{ALPHA}(I) = (((50.\times 180)\times \pi) + R - ((1 - 1)/(N(J) - 1))\times (140.\times 180)\times \pi)
\]

** CALCULATION OF THE ANGLE PHI **

\[
\text{CAPHI} = \sin(\text{ALPHA}(I))
\]

\[
\text{CBPHI} = \cos(\text{ALPHA}(I)) - L4/L3
\]

\[
\text{CCPHI} = \cos(\text{ALPHA}(I)) + (-L1\times L1 + L2\times L2 - L3\times L3 - L4\times L4)/(2\times L1\times L3)
\]

\[
\text{PHI}(I,J) = 2\times \arctan((\text{CAPHI} - \sqrt{\text{CAPHI}\times \text{CAPHI} + CBPHI\times CBPHI - CCPHI\times CCPHI}) / (CBPHI + CCPHI))
\]

IF(I.GT.2) THEN
  TEK = (PHI(I,J) - PHI(I-1,J))\times (PHI(I-1,J) - PHI(I-2,J))
IF(TEK.LT.0) THEN
  N(J) = I-1
  CALL CLEAR
  GAMMA = -((\text{ALPHA}(I-1) - R)\times 180.0D0/\pi - 50)
PRINT 2000
PRINT 1000, "It is not possible to calculate the full range of the flexion angle (gamma) for this system."
PRINT 1000, "The program will calculate the centrodes and the femoral contact profile for:"
PRINT 3000, GAMMA

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix C: Subroutine CENTRO

PRINT 1000, "It is not possible to calculate the full range of the flexion angle (gamma) for this system."
PRINT 1000, "The program will calculate the centrodes and the femoral contact profile for:
PRINT 1000, "The mechanism will be drawn in WHITE"
ELSEIF(J.EQ.2) THEN
PRINT 1000, "This mechanism will be drawn in RED"
ELSE
PRINT 1000, "This mechanism will be drawn in GREEN"
ENDIF
PRINT 1000,
PRINT 2000
PRINT *, 'Press ENTER to continue'
CALL J06WDF
GOTO 2
ENDIF
ENDDIF
GOTO 2

******************************************************************************
** CALCULATION OF THE ANGLE THETA **
******************************************************************************

CATHET = SIN(ALPHA(I))
CCTHET = (-L4/L2) * COS(ALPHA(I))

THETA(I) = 2*ATAN((CATHET - SQRT(CATHET*CATHET + CTHET*CBTHET) - CCTHET*CCTHET)) /

IF(I.GT.2) THEN
TEK = (THETA(I)-THETA(I-1))*(THETA(I-1)-THETA(I-2))
IF(TEK.LT.0) THEN
N(J) = I-1
ENDIF
ENDIF
ENDIF
************e***********************

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix D: Subroutine FEMORA

SUBROUTINE FEMORA(N, NMAX, J, A, H, X0, TC, FC, T, F, L, TIBIAL, IFAIL, ERROR)
C
C******************************************************************************
C** THIS SUBROUTINE CALCULATES THE POINTS OF THE TIBIAL CONTACT PROFILE. THE COORDINATES ARE STORED IN THE VARIABLES T AND F. THE RELEVANT DISTANCE FROM A POINT ON THE CENTRODE TO A POINT ON THE CONTACT PROFILE IS STORED IN L.**
C******************************************************************************

******************************************************************************
** DECLARATION OF THE GLOBAL IDENTIFIERS **
******************************************************************************
INTEGER N(3), NMAX, J, IFAIL, ERROR
DOUBLE PRECISION A(3), H(3), X0(3), TC(2,NMAX,3), FC(2,NMAX,3)
DOUBLE PRECISION T(2,NMAX,3), F(2,NMAX,3), L(NMAX,3)
LOGICAL TIBIAL

******************************************************************************
** DECLARATION OF THE LOCAL IDENTIFIERS **
******************************************************************************
INTEGER I, SP, U, WW
DOUBLE PRECISION NT(2), CT(2), CF(2), CHI, S
DOUBLE PRECISION E, DD, PI
PARAMETER (PI = 3.141592653)

I: COUNTER
SP: STARTING VALUE OF I
U: U = 1 OR -1
WW: WW = 1 OR -1
NT(2): THE COMPONENTS OF THE VECTOR Nt
CT(2): THE COMPONENTS OF THE VECTOR Ct
CF(2): THE COMPONENTS OF THE VECTOR Cf
CHI: THE ANGLE BETWEEN Nt AND Ct
S: THE LENGTHS OF THE VECTOR Cf
E: INTER VARIABLE
DD: INTER VARIABLE
PI: PI = 3.141592653

ERROR = 0

******************************************************************************
** CALCULATION OF FEMORAL CONTACT PROFILE WITH TIBIAL CONTACT PROFILE IS A STRAIGHT LINE **
******************************************************************************

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix D: Subroutine FEMORA

C
** CALCULATION OF FEMORAL CONTACT PROFILE **
** FOR I = SP TO 2 (BACKWARD) **

C
3 T(1,SP+1,J) = X0(J)
U = 1
I = SP
IF (I.GE.2) THEN
   CALL CALC(NMAX, J, I, U, A, H, X0, TC, FC,
   T, F, L, IFAIL, ERROR)
* IF (IFAIL.NE.0) THEN
   PRINT *,,' ERROR SUBROUTINE C02AEF, IFAIL= ',IFAIL
   GOTO 6
ENDIF
I = I - 1
GOTO 4
ENDIF

C
** CALCULATION OF FEMORAL CONTACT PROFILE **
** FOR I = SP TO N-1 (FORWARD) **

C
I = SP + 1
U = -1
5 IF (I.LE.N(J)-1) THEN
   CALL CALC(NMAX, J, I, U, A, H, X0, TC, FC,
   * T, F, L, IFAIL, ERROR)
* IF (IFAIL.NE.0) THEN
   PRINT *,,' ERROR SUBROUTINE C02AEF, IFAIL= ',IFAIL
   GOTO 6
ENDIF
I = I + 1
GOTO 5
ENDIF
ENDIF
END

C
SUBROUTINE CALC(NMAX, J, I, U, A, H, X0, TC, FC,
* T, F, L, IFAIL, ERROR)

C
** SUBROUTINE FOR CALCULATION OF POINT OF FEMORAL CENTRODE **

C
** DECLARATIONS OF THE GLOBAL IDENTIFIERS **

C
INTEGER NMAX, J, I, U, IFAIL, ERROR
DOUBLE PRECISION A(3), H(3), X0(3), TC(2,NMAX,3), FC(2,NMAX,3)

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
DOUBLE PRECISION T(2,NMAX,3), F(2,NMAX,3), L(NMAX,3)

** DECLARATION OF THE LOCAL IDENTIFIERS **

INTEGER Q, K, D, TT, WW
DOUBLE PRECISION AA(4), Z(2,3), XX(3), NT(2), CT(2), CF(2)
DOUBLE PRECISION CHI, S, ZERO, WOR, DIF(3), W, ZZ(3), ZZZ(3), PI
PARAMETER (ZERO = 0.0D0, Q = 4, W=1.0 D -8)
PARAMETER (PI = 3.141592653)

Q: NUMBER OF COEFFICIENTS (Q=4)
K: NUMBER OF REAL ROOTS OF THIRD ORDER POLYNOME
D: COUNTER
TT: COUNTER
WW: WW = 1 OR -1
AA(4): PARAMETERS OF THIRD ORDER POLYNOME
Z(2,3): ROOTS OF THIRD ORDER POLYNOME
XX(3): INTERIM VARIABLE (REAL ROOTS OF THIRD ORDER POLYNOME)
NT(2): THE COMPONENTS OF THE VECTOR Nt
CT(2): THE COMPONENTS OF THE VECTOR Ct
CF(2): THE COMPONENTS OF THE VECTOR Cf
CHI: THE ANGLE BETWEEN Nt AND Ct
S: THE LENGTHS OF THE VECTOR Cf
ZERO: = 0.0 D 0
WOR: INTERIM VARIABLE
DIF(3): INTERIM VARIABLE
W: TOLERANCE FOR SUBROUTINE C02AEF
ZZ(3): REAL PART OF SOLUTION THIRD ORDER POLYNOME
ZZZ(3): IMAGINARY PART OF SOLUTION THIRD ORDER POLYNOME
PI: 3.141592653

** CALCULATIONS FOR PARAMETERS OF POLYNOME **

AA(1) = 2*A(J)*A(J)
AA(2) = -6*A(J)*A(J)*X0(J)
AA(3) = 1-2*A(J)*TC(2,I,J)+2*A(J)*H(J)+6*A(J)*A(J)*X0(J)*X0(J)
AA(4) = 2*A(J)*X0(J)*TC(2,I,J)-2*A(J)*H(J)*X0(J)-
* 2*A(J)*A(J)*X0(J)*X0(J)*X0(J) - TC(1,I,J)

** CALCULATION OF ROOTS OF THIRD ORDER POLYNOME **

ZZ(1) = 1.0D0
ZZZ(1)= 1.0D0
IFAIL = 0

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
CALL C02AEF (AA,Q,ZZ,ZZZ,W,IFAIL)
DO 1 TT=1,3
   Z(1,TT)=ZZ(TT)
   Z(2,TT)=ZZZ(TT)
1 CONTINUE
IF (IFAIL.NE.0) THEN
   PRINT *, ' ERROR SUBROUTINE C02AEF, IFAIL= ',IFAIL
   GOTO 3
ENDIF

******************************************************************************
** SELECTION OF REAL ROOTS **
******************************************************************************

K=0
DO 2 D=1,3
   IF (Z(2,D).EQ.ZERO) THEN
      K=K + 1
      XX(K) = Z(1,D)
   ENDIF
2 CONTINUE
IF(K.GE.2) ERROR = 1

******************************************************************************
** SELECTION OF RELEVANT SOLUTION **
******************************************************************************

IF (K.EQ.1) THEN
   T(1,I,J) = XX(1)
ELSEIF (K.EQ.2) THEN
   DIF(1) = ABS(XX(1) - T(1,I+U,J))
   DIF(2) = ABS(XX(2) - T(1,I+U,J))
   IF (DIF(1).LE.DIF(2)) THEN
      T(1,I,J) = XX(1)
   ELSE
      T(1,I,J) = XX(2)
   ENDIF
ELSE
   DIF(1) = ABS(XX(1) - T(1,I+U,J))
   DIF(2) = ABS(XX(2) - T(1,I+U,J))
   DIF(3) = ABS(XX(3) - T(1,I+U,J))
   IF (DIF(1).LE.DIF(2).AND.DIF(1).LE.DIF(3)) THEN
      T(1,I,J) = XX(1)
   ELSEIF (DIF(2).LE.DIF(3).AND.DIF(2).LE.DIF(1)) THEN
      T(1,I,J) = XX(2)
   ELSE
      T(1,I,J) = XX(3)
   ENDIF
ENDIF

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix E: Subroutine OUTPUT

SUBROUTINE OUTPUT(N, NMAX, J, SA, SB, L4, L, TC, FC, T, F, VARIAT)

******************************************************************************
** THIS SUBROUTINE VISUALISES THE RESULTS OF THE CALCULATIONS **
******************************************************************************

******************************************************************************
** DECLARATION OF THE GLOBAL IDENTIFIERS **
******************************************************************************

INTEGER N(3), NMAX, J
DOUBLE PRECISION SA(2,3), SB(2,3), L4, L(NMAX, 3), TC(2, NMAX, 3)
DOUBLE PRECISION FC(2, NMAX, 3), T(2, NMAX, 3), F(2, NMAX, 3)
LOGICAL VARIAT

******************************************************************************
** DECLARATION OF THE LOCAL IDENTIFIERS **
******************************************************************************

INTEGER I, DAT, II
DOUBLE PRECISION ZERO
PARAMETER (ZERO = 0.0D0, DAT = 7)

I: COUNTER
II: COUNTER
DAT: NUMBER OF OUTPUT DEVICE
ZERO: = 0.0D0

******************************************************************************
** SAVING DATA TO FILE KNEE.DAT **
******************************************************************************

OPEN (DAT, FILE='KNEE.DAT', STATUS='UNKNOWN')
DO 2 II = 1, J
   WRITE (DAT, 1000) SA(1,J), SA(2,J), SB(1,J), SB(2,J)
   WRITE (DAT, 3000)
   DO 1 I = 2, N(J)-1
      WRITE (DAT, 2000) I, TC(1,I,J), FC(1,I,J), T(1,I,J),
                       F(1,I,J), L(I,J), TC(2,I,J), FC(2,I,J), T(2,I,J),
                       F(2,I,J)
   1 CONTINUE
2 CONTINUE

******************************************************************************
** SETTING OF CURRENT LINESTYLE AND COLOUR **
******************************************************************************

J06YQF: SETS THE CURRENT COLOUR
J06YRF: SETS THE CURRENT LINESTYLE

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix E: Subroutine OUTPUT

The design of a software program for the simulation of a 2 dimensional model of the knee joint.

```fortran
C
C  J = 1
C
3  CALL J06YQF(J)
   CALL J06YRF(J)

*****************************************************************************
** DRAWING OF TIBIAL CENTRODE **
*****************************************************************************
J06YAF: MOVES PEN TO A GIVEN POINT
J06YCF: DRAWS LINE BETWEEN TWO POINTS

CALL J06YAF(TC(1,1,J),TC(2,1,J))
DO 4 I=2,N(J)
   CALL J06YCF(TC(1,I,J),TC(2,I,J))
   CONTINUE

*****************************************************************************
** DRAWING OF FEMORAL CENTRODE **
*****************************************************************************

IF(.NOT.VARIAT) CALL J06YQF(2)

CALL J06YAF(FC(1,1,J),FC(2,1,J))
DO 5 I=2,N(J)
   CALL J06YCF(FC(1,I,J),FC(2,I,J))
   CONTINUE

*****************************************************************************
** DRAWING OF TIBIAL CONTACT PROFILE **
*****************************************************************************

IF(.NOT.VARIAT) CALL J06YQF(3)

CALL J06YAF(T(1,2,J),T(2,2,J))
DO 6 I=3,N(J)-1
   CALL J06YCF(T(1,I,J),T(2,I,J))
   CONTINUE

*****************************************************************************
** DRAWING OF FEMORAL CONTACT PROFILE **
*****************************************************************************

IF(.NOT.VARIAT) CALL J06YQF(4)

CALL J06YAF(F(1,2,J),F(2,2,J))
DO 7 I=3,N(J)-1
   CALL J06YCF(F(1,I,J),F(2,I,J))
   CONTINUE
```
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix F: Subroutine PARA

SUBROUTINE PARA(N, NMAX, J, ERROR, R, SA, SB, A, H, X0, *
* L1, L2, L3, L4, L, TC, FC, T, F, TIBIAL, *
* PHI)

******************************************************************************
** THIS SUBROUTINE CALCULATES THE EFFECTS OF VARIATIONS **
** OF SEVERAL PARAMETERS. THIS IS DONE BY MAKING THEM **
** A CERTAIN PERCENTAGE C SMALLER OR GREATER **
******************************************************************************

******************************************************************************
** DECLARATION OF THE GLOBAL IDENTIFIERS **
******************************************************************************

INTEGER N(3), NMAX, J, ERROR
DOUBLE PRECISION R, SA(2,3), SB(2,3), A(3), H(3), X0(3)
DOUBLE PRECISION L1, L2, L3, L4, L(NMAX,3)
DOUBLE PRECISION TC(2, NMAX, 3), FC(2, NMAX, 3)
DOUBLE PRECISION T(2, NMAX, 3), F(2, NMAX, 3)
DOUBLE PRECISION PHI(NMAX,3)
LOGICAL TIBIAL

******************************************************************************
** DECLARATION OF THE LOCAL IDENTIFIERS **
******************************************************************************

INTEGER I, CHAR, IFAIL, UU
DOUBLE PRECISION C(2), CC(2), CCC(2)

I: COUNTER
CHAR: INPUT VARIABLE
IFAIL: ERROR INDICATOR
UU: INPUT VARIABLE
C(2): VALUE FOR PARAMETER VARIATION
CC(2): VALUE FOR PARAMETER VARIATION
CCC(2): VALUE FOR PARAMETER VARIATION

C(1) = -0.1D0
C(2) = 0.1D0
CC(1) = -10.0D0
CC(2) = 10.0D0
CCC(1)= -0.0025
CCC(2)= 0.0025

******************************************************************************
** CHOICE OF PARAMETER TO BE VARIATED **
******************************************************************************

1 PRINT *, ' PARAMETER VARIATION '
PRINT *, ' ' PARAMETER VARIATION '

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix F: Subroutine PARA

```
PRINT *, '1) Elements are rigid
PRINT *, '2) Elements are flexible
PRINT *, 'Enter number of your choice'
READ *, UU
CALL CLEAR
IF(UU.LT.1.OR.UU.GT.2) THEN
  GOTO 1
ENDIF

IF(UU.EQ.2) THEN
  PRINT 1000
  PRINT 2000,
  PRINT 2000,
  PRINT 2000,
  PRINT 2000,
  PRINT 1000
  PRINT *, 'Press ENTER to continue'
  CALL J06WDF
  GOTO 1
ENDIF
CALL J06WBF(-25.0D0, 250.0D0, -25.0D0, 250.0D0, 1)

IF(R.GT.1) THEN

  *******************************
  ** PICTURE OF CROSSED MECHANISM **
  *******************************

  CALL J06YAF(0.0D0, 0.0D0)
  CALL J06YCF(100.0D0, 0.0D0)
  CALL J06YCF(35.0D0, 70.0D0)
  CALL J06YCF(75.0D0, 85.0D0)
  CALL J06YCF(0.0D0, 0.0D0)
  CALL J06YAF(80.0D0, 90.0D0)
  CALL J06YHF('A',1)
  CALL J06YAF(25.0D0, 75.0D0)
  CALL J06YHF('B',1)
  CALL J06YAF(10.0D0,-30.0D0)
  CALL J06YHF('CROSSED MECHANISM',17)

ELSE

  *******************************
  ** PICTURE OF UNCROSSED MECHANISM **
  *******************************

  CALL J06YAF(0.0D0, 0.0D0)
  CALL J06YCF(100.0D0, 0.0D0)
  CALL J06YCF(75.0D0, 85.0D0)
```

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
CALL J06YCF(35.0D0, 70.0D0)
CALL J06YCF(0.0D0, 0.0D0)
CALL J06YAF(80.0D0, 90.0D0)
CALL J06YHF('B',1)
CALL J06YAF(25.0D0, 75.0D0)
CALL J06YHF('A',1)
CALL J06YAF(5.0D0,-30.0D0)
CALL J06YHF('UNCROSSED MECHANISM',19)

ENDIF

CALL J06YAF(-10.0D0, -10.0D0)
CALL J06YHF('O=(0, 0)',8)
CALL J06YAF(100.0D0, -10.0D0)
CALL J06YHF('P=(100, 0)',10)
CALL J06YAF(20.0D0, 0.0D0)
CALL J06YCF(17.0D0, 2.0D0)
CALL J06YCF(17.0D0, -2.0D0)
CALL J06YCF(20.0D0, 0.0D0)
CALL J06YAF(20.0D0, 2.0D0)
CALL J06YHF('X',1)
CALL J06YAF(0.0D0, 0.0D0)
CALL J06YCF(0.0D0, 20.0D0)
CALL J06YCF(2.0D0, 17.0D0)
CALL J06YCF(-2.0D0, 17.0D0)
CALL J06YCF(0.0D0, 20.0D0)
CALL J06YAF(-10.0D0, 20.0D0)
CALL J06YHF('Y',1)

IF (TIBIAL) THEN
  PRINT *,
  PRINT *, WHICH PARAMETER WOULD YOU LIKE TO VARIATE ?
  PRINT *, | 1) X-coordinate of point A |
  PRINT *, | 2) Y-coordinate of point A |
  PRINT *, | 3) X-coordinate of point B |
  PRINT *, | 4) Y-coordinate of point B |
  PRINT *, | 5) Slope of straight line |
  PRINT *, | 6) Vertical displacement of straight line |
ELSE
  PRINT *,
  PRINT *, WHICH PARAMETER WOULD YOU LIKE TO VARIATE ?
  PRINT *, | 1) X-coordinate of point A |
  PRINT *, | 2) Y-coordinate of point A |
  PRINT *, | 3) X-coordinate of point B |
  PRINT *, | 4) Y-coordinate of point B |
  PRINT *, | 5) Convexity/concavity of parabole |
  PRINT *, | 6) Vertical displacement of parabole top |
  PRINT *, | 7) Horizontal displacement of parabole top |
ENDIF

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
** CALCULATION OF EFFECTS OF VARIATION **

```
PRINT *, 'Calculating centrodes'
CALL CENTRO(N, NMAX, J, R, SA, SB, L1, L2, L3, L4,
        *
        TC, FC, PHI)
PRINT *, 'Calculating femoral contact profile'
CALL FEMORA(N, NMAX, J, A, H, XO, TC, FC, T,
        *
        F, L, TIBIAL, IFAIL, ERROR)
IF (IFAIL.NE.0) GOTO 4
```

```
IF (ERROR.EQ.1) THEN
    PRINT 3000
    PRINT 4000,
    PRINT 4000,
    PRINT 4000,
    PRINT 4000,
    PRINT 4000,
    PRINT 4000,
    PRINT 4000,
    PRINT 4000,
    PRINT 4000,
    PRINT 4000,
    PRINT 4000,
    PRINT 3000
    PRINT *, 'Press ENTER to continue'
    CALL J06WDF
ENDIF
```

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix G: Subroutine SLIP

SUBROUTINE SLIP(N, NMAX, NIN, T, F, VARIAT)

C
C ** THIS ROUTINE CALCULATES THE SLIP RATIO **
C ** AND PLOTS THE RESULTS **
C
C ** DECLARATION OF THE GLOBAL IDENTIFIERS **
C
INTEGER N(3), NMAX, NIN
DOUBLE PRECISION T(2,NMAX,3), F(2,NMAX,3)
LOGICAL VARIAT

C
C ** DECLARATION OF THE LOCAL IDENTIFIERS **
C
INTEGER NNNMAX, J
PARAMETER (NNNMAX = 200)
DOUBLE PRECISION NN(NNNMAX,3)
DOUBLE PRECISION TL(NNNMAX,3), TDF(NNNMAX,3)
DOUBLE PRECISION NMAX, FL(NNNMAX,3), STL(3), SFL(3)
DOUBLE PRECISION TDFMAX
DOUBLE PRECISION STDF(3)

C
C NNNMAX : MAXIMAL NUMBER OF ITERATION
C J : COUNTER
C NN(NNNMAX,3) : FLEXION ANGLE IN DEGREES
C TL(NNNMAX,3) : LENGTH OF PART OF TIBIAL CONTACT PROFILE
C FDF(NNNMAX,3) : SLIP RATIO
C NMAX :
C FL(NNNMAX,3) : LENGTH OF PART OF FEMORAL CONTACT PROFILE
C STL(3) : SUM OF TL
C SFL(3) : SUM OF FL
C TDFMAX : MAXIMAL SLIP RATIO
C STDF : AVERAGE SLIP RATIO

C
C ** INITIALISATION OF VARIABLES **
C
IF(VARIAT) THEN
   NMAX = MAX(N(1), N(2), N(3))
ELSE
   NMAX = N(1)
ENDIF

C J = 0

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix G: Subroutine SLIP

TDFMAX = 0

*******************************************************************************
** CALCULATION OF SLIP RATIO **
*******************************************************************************

J = J + 1
STL(J) = 0
SFL(J) = 0
DO 2 I = 2, N(J)-2
   TL(I,J) = SQRT((T(1,I+1,J)-T(1,I,J))*(T(1,I+1,J)-T(1,I,J)) +
   (T(2,I+1,J)-T(2,I,J))*(T(2,I+1,J)-T(2,I,J)))
   * FL(I,J) = SQRT((F(1,I+1,J)-F(1,I,J))*(F(1,I+1,J)-F(1,I,J)) +
   (F(2,I+1,J)-F(2,I,J))*(F(2,I+1,J)-F(2,I,J)))
   TDF(I,J) = FL(I,J)/TL(I,J)
   TDFMAX = MAX (TDFMAX, TDF(I,J))
   STL(J) = STL(J) + TL(I,J)
   SFL(J) = SFL(J) + FL(I,J)
   NN(I,J) = 140.0D0*(I-1)/(NIN-1)
2 CONTINUE

*******************************************************************************
** CALCULATION OF AVERAGE SLIP RATIO **
*******************************************************************************

STDF(J) = SFL(J) / STL(J)
IF(J.LT.3.AND.VARIAT) GOTO 1

*******************************************************************************
** PLOTTING OF RESULTS **
*******************************************************************************

CALL CLEAR
CALL J06WBF(0.0D0, 140.0D0, 0.0D0, TDFMAX, 1)
CALL J06WCF(0.0D0, 1.0D0, 0.0D0, 1.0D0)

J = 0
J = J+1

*******************************************************************************
** SETTING OF CURRENT LINESTYLE AND COLOUR **
*******************************************************************************

J06YQF: SETS THE CURRENT COLOUR
J06YRF: SETS THE CURRENT LINESTYLE

CALL J06YQF(J)

*******************************************************************************
** DRAWING OF LINES **
*******************************************************************************

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
CALL JO6YAF(NN(2,J),TDF(2,J))
DO 4 I = 3, N(J)-3
CALL JO6YCF(NN(I,J),TDF(I,J))
CONTINUE

................................... ** DRAWING OF AVERAGE SLIP RATIO ** ...................................
CALL JO6YRF(2)
CALL JO6YAF(NN(2,J),STDF(J))
CALL JO6YCF(NN(N(J)-3,J),STDF(J))
CALL JO6YRF(1)
IF(J.LT.3.AND.VARIAT) GOTO 3

................................ ** DRAWING OF AXES AND TITLES ** ................................
CALL JO6YQF(1)
CALL JO6YAF(0.0D0,0.0D0)
CALL JO6YQF(4)
CALL JO6AJF(1,'FLEXION ANGLE')
CALL JO6AJF(2,'SLIP RATIO')
CALL JO6AHF('SLIP RATIO')
CALL JO6YQF(1)
CALL JO6AAF
CALL JO6WDF
CALL JO6WBF(-25.0D0, 150.0D0, -25.0D0, 150.0D0, 1)
CALL JO6YL(5.0D0,0.0D0)
CALL JO6YKF(5.0D0,7.0D0)

END

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix H: Subroutine MOVIE

SUBROUTINE MOVIE(N, NMAX, SA, SB, TC, FC, T, F, VARIAT, 
                  PHI, R, NIN, L4)

******************************************************************************
** THIS SUBROUTINE CALCULATES THE NEW COORDINATES OF **
** SA, SB, FC AND F, WHEN WE ROTATE L3. **
******************************************************************************

******************************************************************************
** DECLARATION OF THE GLOBAL IDENTIFIERS **
******************************************************************************

INTEGER N(3), NMAX, NIN
DOUBLE PRECISION SA(2,3), SB(2,3), TC(2,NMAX,3), FC(2,NMAX,3)
DOUBLE PRECISION T(2,NMAX,3), F(2,NMAX,3), PHI(NMAX,3)
DOUBLE PRECISION R, L4
LOGICAL VARIAT

******************************************************************************
** DECLARATION OF THE LOCAL IDENTIFIERS **
******************************************************************************

INTEGER I, II, III, J, NNMAX, NUM, CHAR
PARAMETER (NNMAX = 200, NUM = 7)
DOUBLE PRECISION FCLO(2,NNMAX,3), FLO(2,NNMAX,3), SBLO(2,3)
DOUBLE PRECISION ROT, TRANS(2), NESA(2,3,NUM), NESB(2,3,NUM)
DOUBLE PRECISION NEFC(2,NNMAX,3,NUM), NEF(2,NNMAX,3,NUM), PI
DOUBLE PRECISION SALPHA, L1
PARAMETER (PI = 3.141592653)

I : COUNTER
II : COUNTER
III : COUNTER
J : COUNTER
NNMAX : MAXIMAL NUMBER OF ITERATIONS
NUM : COUNTER (NUMBER OF MOVING STEPS)
CHAR : INPUT VARIABLE
FCLO(2,NNMAX,3) : LOCAL COORDINATES OF FEMORAL CENTRODE (FC)
FLO(2,NNMAX,3) : LOCAL COORDINATES OF FEMORAL CONTACT PROFILE (F)
SBLO(2,3) : LOCAL COORDINATES OF B
ROT : ROTATION ANGLE
TRANS(2) : TRANSLATION VECTOR
NESA(2,3,NUM) : COORDINATES OF A AFTER TRANSLATION/ROTATION
NESB(2,3,NUM) : COORDINATES OF B AFTER TRANSLATION/ROTATION
NEFC(2,NNMAX,3,NUM) : COORDINATES OF FC AFTER TRANSLATION/ROTATION
NEF(2,NNMAX,3,NUM) : COORDINATES OF F AFTER TRANSLATION/ROTATION
PI : PI = 3.141592653
SALPHA : ALPHA AT THE GIVEN POSITION
L1 : LENGHTS OF LINK OA

The design of a sofware program for the simulation of a 2 dimensional model of the knee joint.
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix H: Subroutine MOVIE

*                               \[ (((\text{III}-1.0D0)/(\text{NIN}-1.0D0))*\]
*                               (140.0D0/180.0D0)*\Pi) - \text{SALPHA}

\[
\text{NES}(1,J,II) = L1 \times \cos(\text{PHI}(\text{III},J)) \\
\text{NES}(2,J,II) = L1 \times \sin(\text{PHI}(\text{III},J)) \\
\text{TRANS}(1) = \text{NES}(1,J,II) - \text{SA}(1,J) \\
\text{TRANS}(2) = \text{NES}(2,J,II) - \text{SA}(2,J)
\]

**************************
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
*                        **
**************************

** CALCULATION OF THE NEW COORDINATES OF B,  **
** AND THE FEMORAL CENTRODE AND CONTACT PROFILE **

\[
\text{NESB}(1,J,II) = \text{SA}(1,J) + \text{TRANS}(1) \\
+ \text{SBLO}(1,J)\times\cos(\text{ROT}) \\
- \text{SBLO}(2,J)\times\sin(\text{ROT})
\]

\[
\text{NESB}(2,J,II) = \text{SA}(2,J) + \text{TRANS}(2) \\
+ \text{SBLO}(1,J)\times\sin(\text{ROT}) \\
+ \text{SBLO}(2,J)\times\cos(\text{ROT})
\]

DO 4 I = 2, N(J)-1

\[
\text{NEFC}(1,I,J,II) = \text{SA}(1,J) + \text{TRANS}(1) \\
+ \text{FCLO}(1,I,J)\times\cos(\text{ROT}) \\
- \text{FCLO}(2,I,J)\times\sin(\text{ROT})
\]

\[
\text{NEFC}(2,I,J,II) = \text{SA}(2,J) + \text{TRANS}(2) \\
+ \text{FCLO}(1,I,J)\times\sin(\text{ROT}) \\
+ \text{FCLO}(2,I,J)\times\cos(\text{ROT})
\]

\[
\text{NEF}(1,I,J,II) = \text{SA}(1,J) + \text{TRANS}(1) \\
+ \text{FLO}(1,I,J)\times\cos(\text{ROT}) \\
- \text{FLO}(2,I,J)\times\sin(\text{ROT})
\]

\[
\text{NEF}(2,I,J,II) = \text{SA}(2,J) + \text{TRANS}(2) \\
+ \text{FLO}(1,I,J)\times\sin(\text{ROT}) \\
+ \text{FLO}(2,I,J)\times\cos(\text{ROT})
\]

CONTINUE

CONTINUE

PRINT *, ' WHICH TYPE OF PLOT ?'
PRINT *, 1) Several positions in one plot'
PRINT *, 2) More positions one by one'
PRINT *, 3) Back to the main menu'
PRINT*, ' Enter number of your choice'
READ *, CHA
CALL CLEAR
IF(CHAR.EQ.1) THEN
  GOTO 18
ELSEIF(CHAR.EQ.2) THEN
  GOTO 19
ELSEIF(CHAR.EQ.3) THEN
  GOTO 16
ELSE

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix H: Subroutine MOVIE

CONTINUE

**************************************************************************
** DRAWING THE FOUR LINKS **
**************************************************************************

CALL J06YQF(5)
CALL J06YAF(0.0D0, 0.0D0)
CALL J06YCF(L4, 0.0D0)
CALL J06YCF(NESB(1,J,II), NESB(2,J,II))
CALL J06YCF(NESA(1,J,II), NESA(2,J,II))
CALL J06YCF(0.0D0, 0.0D0)

CONTINUE

PRINT 1000
PRINT *, 'Press enter to continue'

CALL J06WDF
CALL J06YQF(1)
IF(VARIAT) GOTO 15
GOTO 17

**************************************************************************
** **** DRAWING NUM MECHANISMS ONE BY ONE ****
**************************************************************************

DO 10 II=1, NUM

**************************************************************************
** ** DRAWING OF TIBIAL CENTRODE **
**************************************************************************

J06YAF: MOVES PEN TO A GIVEN POINT  
J06YCF: DRAWS LINE BETWEEN TWO POINTS

CALL J06YQF(1)
CALL J06YAF(TC(1,2,J),TC(2,2,J))
DO 11 I=3, N(J)-1
   CALL J06YCF(TC(1,I,J),TC(2,I,J))
CONTINUE

**************************************************************************
** ** DRAWING OF TIBIAL CONTACT PROFILE **
**************************************************************************

CALL J06YQF(3)
CALL J06YAF(T(1,2,J),T(2,2,J))
DO 12 I=3,N(J)-1
   CALL J06YCF(T(1,I,J),T(2,I,J))
CONTINUE

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
Appendix H: Subroutine MOVIE

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
USER'S MANUAL
Report Number 92.013 C

Frank Peters
Niek van Nunen

Eindhoven University of Technology, January 1992
1. Contents

1. Contents ................................................................. 1
2. Introduction ............................................................. 2
3. The program ............................................................. 3
   3.1 Starting the program ............................................. 3
   3.2 The main menu ..................................................... 3
   3.2.1 Input of a mechanism ......................................... 4
   3.2.1.1 Anatomic dimensions ...................................... 5
   3.2.1.2 Perspex model .............................................. 5
   3.2.1.3 Self made mechanism ...................................... 6
   3.2.2 Plot results ..................................................... 6
   3.2.3 Parameter variation ........................................... 7
   3.2.4 Slip ratio ....................................................... 8
   3.2.5 Movie ............................................................ 9
   3.2.6 Quit ............................................................. 10
   3.3 The calculations .................................................. 10
   3.4 Termination of the program ..................................... 12
4. Finally ................................................................. 12
5. Literature ............................................................ 13

The design of a software program for the simulation of a 2 dimensional model of the knee joint.
2. Introduction

The bones that meet a joint, as well as the ligaments that hold the bones together, can be analyzed as a mechanical linkage. Since the movements allowed to the bones at the human knee occur mainly in the sagittal plane, much can be learned by treating the knee as a two-dimensional single-degree-of-freedom linkage, moving in a single plane. Figure 1 shows a human knee from which the lateral femoral condyle has been excised, exposing the cruciate ligaments. The ligaments, together with the two bones, form the cruciate linkage. \( AB \) and \( PB \) represent the anterior (ACL) and posterior (PCL) cruciate ligaments, respectively.

![Diagram of knee joint with ligaments labeled](image)

The program knee simulates a 2 dimensional kinematic model of a knee joint. It calculates, given a mechanical linkage and tibial contact profile, the corresponding tibial and femoral centrodes and the femoral contact profile. A normal knee joint can move from full extension to about 140° flexion. These values will also be used in the program. The user has the opportunity to select a straight line as tibial contact profile or a parabole. He can also choose between a crossed or an uncrossed linkage. After the program has calculated the centrodes and contact profiles, the user can variate one of the parameters and examine the effects of the parameter variation. The slip ratio between the tibial and the femoral contact profile can be plotted in a diagram. The program also has the possibility to plot the mechanism with the centrodes and contact profiles in several positions for different flexion angles. All the calculations will be made for a mechanism with rigid elements. It's the intention to extend the program with the opportunity to use flexible elements, but at this moment the program does not have this opportunity yet.
3. The program

3.1 Starting the program

It’s easy to start the program. When the disk is in the disk drive, the user only has to type:

\[
A: [\text{ENTER}] \\
\text{knee} [\text{ENTER}] \\
\]

As soon the program has started it is not allowed to use strings anymore (the easiest way to avoid that this will happen anyhow, is to use only the numerical part of the keyboard), otherwise the program will be terminated. First the program will ask which graphical device is to be used. **Always choose: 1. PC SCREEN.** This switches the terminal to the graphical mode.

3.2 The main menu

Next the main menu will appear (figure 2).

```
<table>
<thead>
<tr>
<th>MAIN MENU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Input of a mechanism</td>
</tr>
<tr>
<td>2) Plot results</td>
</tr>
<tr>
<td>3) Parameter variation</td>
</tr>
<tr>
<td>4) Slip ratio</td>
</tr>
<tr>
<td>5) Movie</td>
</tr>
<tr>
<td>6) Quit</td>
</tr>
</tbody>
</table>
```

Enter number of your choice

*figure 2: The main menu*

When no mechanism has been chosen yet, the user can only choose: 1) Input of a mechanism. Otherwise a message "No mechanism chosen yet" will be given. As soon as a mechanism has been chosen, also one of the other opportunities can be chosen.
3.2.1 Input of a mechanism

The program needs the following parameters (figure 3):

- Whether the mechanism is a crossed or an uncrossed one.
- The coordinates of the points A and B (the coordinates of the points O and P are not variable: O = (0, 0) and P = (100, 0).
- Whether the tibial contact profile will be described by a straight line or a parabole.
- The parameters of the straight line or the parabole:
  - The straight line will be described by:
    "\( Y = AX + H \)"
    with \( A \) the slope of the straight line and.
    \( H \) the height of the line at \( X = 0 \).
  - The parabole will be described by:
    "\( Y = A(X-X_0)^2 + H \)"
    with \( A \) the convexity/concavity of the parabole.
    \( X_0 \) the x-coordinate of the top of the parabole.
    \( H \) the height of the top of the parabole.
  - The number of increments (N).

Figure 3: The two types of mechanisms and some of the parameters to be entered.

When the user chooses option 1 at the main menu, the choice out of three kinds of mechanisms will be given (see figure 4).
3.2.1.1 Anatomic dimensions

The parameters for a mechanism with anatomic (realistic) dimensions are stored in the program. These parameters are:

* The mechanism is a crossed one
* $A = (61, 77)$
* $B = (20, 68)$
* The tibial profile is a parabola
* $A = -0.005$
* $X_0 = 50$
* $H = 7$
* $N = 200$

3.2.1.2 Perspex model

The faculty of Medical Engineering Technology of the Eindhoven University of Technology has the disposal of a 2 dimensional Perspex model of the knee joint. The dimensions of this model are also stored in the program. These values for the parameters then are:

* The mechanism is a crossed one
* $A = (91, 78)$
* $B = (42, 78)$
* The tibial profile is a straight line
* $A = 0$
* $H = 7$
* $N = 200$
3.2.1.3 Self made mechanism

In this case the user has to enter the parameters him/herself. The program will ask for them one by one. After every value the user gives, the ENTER-button has to be pressed. The program will also give a recommended range for some of the parameters.

3.2.2 Plot results

This option draws a plot of the mechanism in the given position. It is possible that there will be no contact between the tibial contact profile and the femoral contact profile. In this case the given position has a flexion angle which is less than 0° or larger than 140°. The program only calculates for a range between 0° and 140°.

In the case that the user didn't use the opportunity of parameter variation yet, there will be drawn only one mechanism:

- The yellow line is the input situation of the linkage.
- The white line is the tibial centrode.
- The red line is the femoral centrode.
- The green line is the tibial contact profile.
- The blue line is the femoral contact profile.

In the other case (the option parameter variation has been chosen yet) there will be drawn three mechanisms:
- The white mechanism is the one without parameter variation.
- The red mechanism is the one with negative parameter variation.
- The green mechanism is the one with positive parameter variation.
(see also 3.2.3 Parameter variation)

### 3.2.3 Parameter variation

When the user chooses the option "3) parameter variation", the next menu will appear (figure 5):

<table>
<thead>
<tr>
<th>PARAMETER VARIATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Elements are rigid</td>
</tr>
<tr>
<td>2) Elements are flexible</td>
</tr>
</tbody>
</table>

Enter number of your choice

**figure 5** The parameter variation menu. The choice for flexible elements is not yet possible.

The user has to choose for opportunity 1, while opportunity 2 is not available yet. In the situation where the tibial contact profile is a straight line, the following menu will appear (figure 6):

<table>
<thead>
<tr>
<th>WHICH PARAMETER WOULD YOU LIKE TO VARIATE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) X-coordinate of point A</td>
</tr>
<tr>
<td>2) Y-coordinate of point A</td>
</tr>
<tr>
<td>3) X-coordinate of point B</td>
</tr>
<tr>
<td>4) Y-coordinate of point B</td>
</tr>
<tr>
<td>5) Slope of straight line</td>
</tr>
<tr>
<td>6) Vertical displacement of straight line</td>
</tr>
</tbody>
</table>

Enter number of parameter to be varied

**Figure 6**: Menu for parameter variation with tibial contact profile is a straight line.

When the user chooses for options 1, 2, 3, 4 or 6 the chosen parameter will set on:
- Its old value - 10
- Its old value + 10

When the user chooses for option 5 the slope of the line will be changed in:
The program will calculate the centrodes and the contact profiles for both cases.

If a parabole is chosen for the tibial contact profile the user will get the next menu (figure 7):

<table>
<thead>
<tr>
<th>WHICH PARAMETER WOULD YOU LIKE TO VARIATE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) X-coordinate of point A</td>
</tr>
<tr>
<td>2) Y-coordinate of point A</td>
</tr>
<tr>
<td>3) X-coordinate of point B</td>
</tr>
<tr>
<td>4) Y-coordinate of point B</td>
</tr>
<tr>
<td>5) Convexity/concavity of parabole</td>
</tr>
<tr>
<td>6) Vertical displacement of parabole top</td>
</tr>
<tr>
<td>7) Horizontal displacement of parabole top</td>
</tr>
</tbody>
</table>

Enter number of parameter to be variated

Figure 7: Menu for parameter variation with femoral contact profile is a parabole.

When the user chooses for options 1, 2, 3, 4, 6 or 7 the chosen parameter will be changed in:

- Its old value -10
- Its old value +10

When option 5 is chosen the convexity/concavity of the parabole will be changed in:

- Its old value -0.001
- Its old value +0.001

3.2.4 Slip ratio

This option calculates the slip ratio and plots it against the flexion angle. When the slip ratio is equal to 1 this means there is no slip. In the diagram the average slip is also plotted. When the option parameter variation has been chosen three lines will be drawn:

- The white one for the mechanism without variation.
The red one for the mechanism with neg. variation.
The green one for the mechanism with pos. variation.

3.2.5 Movie

The program will calculate a number of different positions of the femur with respect to the tibia. If the option parameter variation has been chosen, the next menu will appear (figure 8):

<table>
<thead>
<tr>
<th>WHICH MECHANISM WOULD YOU LIKE TO MOVE ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) The mechanism without parameter variation</td>
</tr>
<tr>
<td>2) The mechanism with neg. parameter variation</td>
</tr>
<tr>
<td>3) The mechanism with pos. parameter variation</td>
</tr>
<tr>
<td>4) Back to the main menu</td>
</tr>
</tbody>
</table>

Enter number of your choice

*Figure 8:* Option menu for routine MOVIE (this menu only appears when the option parameter variation has been chosen yet.

In this way each of the three possible mechanisms can be shown. After the user has chosen a mechanism, and if the option parameter variation has not been chosen yet, the following menu will appear (see figure 9):

<table>
<thead>
<tr>
<th>WHICH TYPE OF PLOT ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Several positions in one plot</td>
</tr>
<tr>
<td>2) More positions one by one</td>
</tr>
<tr>
<td>3) Back to the main menu</td>
</tr>
</tbody>
</table>

Enter number of your choice

*Figure 9:* The option MOVIE

The user can choose between several positions of the mechanism in one and the same plot, or he can look at several calculated positions of the mechanisms one by one. In this case more plots will be drawn than in the first case.
3.2.6 Quit

This option will quit the program. It also resets the terminal to the normal (VGA/EGA/....) video mode.

3.3 The calculations

When the user has given a mechanism, or when he has varied one of the parameters, the program will start the calculations automatically. The centrodes and the profiles will be calculated from full extension (flexion angle is 0°) to full flexion (flexion angle is 140°). This agrees with a range of -90° to 50° for the angle α of the femoral link (see figure 10).

In some cases the program will give a warning:

1) When the program rotates the tibial link from -90° to 50° it is possible that the tibial link becomes in a direct line with one of the cruciate ligaments. When this happens, the femoral centrode will get what is called a "tail" (see figure 11).
Demonstration of the problem that appears when the mechanism changes from crossed to uncrossed. The program will detect the beginning of the "Tail" and stop further calculations of the centrode.

This causes a discontinuity of the femoral contact profile. To avoid this problem the program checks if this is happening and stops the calculations at that moment. Then the user will get the following warning (figure 12):

\[
\text{It is not possible to calculate the full range of the flexion angle (gamma) for this system. The program will calculate the centrodes and the femoral contact profile for:}
\]
\[
0^\circ \leq \gamma \leq \ldots \ldots ^\circ
\]
\[
\text{The mechanism will be drawn in \ldots \ldots .}
\]

Figure 12: The warning given when it is not possible to simulate the whole flexion range.

The user can read in the warning for which range of the flexion angle the program could calculate the centrodes. The warning gives also the colour of the plot this mechanism will be drawn in. This is only of interest when the option parameter variation has been chosen.

2) It is possible that the combination of the chosen linkage and the tibial contact profile does not fit. In that case the next warning will be given (figure 13):
WARNING: It is not possible to find a femoral contact profile which fits. Change the vertical displacement H (larger) or the convexity A (smaller absolute value). (First look at the results.)

Figure 13: Warning when it is not possible to calculate a combination of contact profiles that fit. Also some suggestions to solve the problem are given.

The program doesn’t stop the calculations, so it’s possible to take a look to what happens in this case. Sometimes it is possible to find parts of the profiles which could fit but it is impossible to find a femoral contact profile which fits for the whole flexion range.

3.4 Termination of the program

When the program is terminated, for example when a string is entered where a number was expected, the monitor will not be reset to the normal (VGA/EGA/...) video mode. The easiest way to get back to this mode is the restart the program (by typing knee [ENTER]). Now the user can continue the program, or if desired quit the program from the main menu.

4. Finally

This program has been written for a simulation practical that is a part of the compulsory part of the curriculum of the faculty of Medical Engineering Technology of the Eindhoven University of Technology. More information about the modelling of the knee by a two dimensional model, is given during one of the courses, Functional anatomy. More information about the model can also be derived from Lit. [1], [2] and [5] (see 5. Literature).
5. Literature

[1] Ian A.F. Stokes (Editor)
Mechanical factors and the skeleton
John Libbey & Company Limited, London
ISBN 0-86196-006-8

Knee Ligaments; Structure, Function, Injury and Repair
Raven Press, New York

Kinematic Synthesis of Linkages
McGraw-Hill series in MECHANICAL ENGINEERING
McGraw-Hill Book company, 1964

Kinematica
Scheltema & Holkema, Amsterdam, 1970
ISBN 90-6060-502-0

A Model of the Human Knee, Derived from Kinematic Principles and Its Relevance for Endoprothesis Design.

[6] Software used:
Wordperfect 5.1 NL
Drawperfect 1.1 NL
MS-Fortran 5.0 Compiler
NAG Workstation Library
NAG PC Graphics library
Norton Classic Editor, version 1.5

All software used was retrieved from the Calculation Center (Rekencentrum) of the Eindhoven University of Technology.

[7] Hardware used:
Rembrant 386 AT with 80387 coprocessor
VGA card with colour monitor

This hardware was retrieved from the Department of Fundamental Research, Faculty of Mechanical Engineering of the Eindhoven University of Technology.

The design of a software program for the simulation of a 2 dimensional model of the knee joint.