Innovative Teaching of Aircraft Structural Analysis and Design Courses - Mathematica in an Engineering Education Environment

Gillian N. Saunders-Smits, Zafer Gürdal, Jan Hol,

Aerospace Structures
Faculty of Aerospace Engineering
Delft University of Technology, Delft, The Netherlands

INTRODUCTION

This paper reports on a new course on aircraft structural analysis and design in the second year of the BSc curriculum at the Faculty of Aerospace Engineering at Delft University of Technology in the Netherlands. The course is aimed at improving the understanding of the design drivers in structures as well as increasing the student's motivation in undertaking structural design. It bridges the gap between the basic mechanics knowledge and its application as foundation for advanced mathematical models such as Finite Element codes used in modern design environments. Already established knowledge of elementary mechanics equations of deformable structures as taught in the first year of the BSc curriculum are used to develop discretized equivalent numerical models of components for design configurations of statically determinate and indeterminate structural problems. The engineering tool Mathematica which provides state of the art symbolic and numerical solution techniques with graphical representation facilities embedded in text and equation handling capabilities within an integrated notebook environment, is used as an integral part of the course delivery.

STRUCTURAL DESIGN EDUCATION IN THE BSC AEROSPACE ENGINEERING

Design education in the Faculty of Aerospace Engineering at Delft University of Technology (TU Delft) starts with the first year courses. In their first year, students are required to take a simple structural design project of 2 ECTS (European Credit Transfer System, 1 ECTS = 25-30 hrs) as described in reference 1. This project consists of the design to specification, the building, and the testing of a box-beam for a wing or a satellite. The boxes are made of aluminum sheets and pre-pressed aluminum ribs and L-shaped stiffeners. Students are free to vary the rivet-, rib- and stringer pitch of their design based on their calculations using basic mechanics of materials knowledge and simplified buckling formulae. The satellite box is then used to measure its eigenfrequencies, and both the wing box and the satellite box are then loaded till failure. Prizes are awarded for the best designs.

The first year project is followed in the second year by the 4 ECTS structural analysis and design course, featured in this paper, as well as another 6 ECTS hands on structural design project[1]. All features of aerospace design are then incorporated into the third year 14 ECTS Design Synthesis Exercise, lasting 10 full weeks, that serves as the final project of the BSc degree program.

Why change a good existing course?

The course reported in this paper has been part of the curriculum earlier. The novelty reported here is partly because of the changes in the content and partly because of the way the course material is delivered. In the following we briefly discuss the changes in the content. The main emphasis of this paper is the change in the delivery of the course and the software tool used in the delivery, which is discussed in some more detail.

Course content changes:

The previous Aircraft Stress Analysis and Structural Design course was heavily analysis oriented and mostly limited to aircraft. Standard mechanics equations were provided and used for analysis of aircraft structural components. The new emphasis is on the fundamentals of structural design and the treatment of design of structural systems with multidisciplinary features. While at the same time integrating mathematical and engineering mechanics skills into the design process. In particular, following the modern trends in numerical structural analysis, the standard mechanics of deformable bodies approach is extended to discretized structural components, such as beams with multiple uniform segments, treating the dimensional properties as design variables and implementing stress based failure criteria to size them. Also emphasized is the difference between the statically determinate and indeterminate structural configurations, where design changes in the former case do not alter the internal load distribution, but may have substantial influence on the internal load paths in the latter case, requiring analysis and sizing steps to be repeated.

Changes to course delivery:

The course was also renewed to meet the needs of the changing undergraduate environment. This environment is rapidly changing as a result of:

i) Changes in the students' study and learning habits,

ii) Expectations of the higher level course instructors and the industry employers from the undergraduate students, and

iii) The availability of powerful numerical tools that enable graduating engineers to perform a variety of day-to-day task in their work environment.

In terms of the students' learning and study habits, two main tendencies appear to be the prevalent challenges for the educators. The first one is the shift of students' skills from being mathematical and analytical to becoming more visual. It is a commonly observed and socially accepted fact that the children grow up with more visual input than before[7]. The recent infiltration of computers into more and more homes would only add to the already widespread visual input in the form of television broadcasts at home. The second tendency is much more recent and is based on the widespread use of the World Wide Web, and availability of information at any time and anywhere. Very rapidly, the students are becoming accustomed to acquiring information at their own pace and at a time when they need it, not when it is available at the choosing of their course instructors. Next to that, recent changes in the Dutch high school curriculum from the traditional teaching to a more assignment and tutorial based form of teaching means a different type of student is entering university.

The second issue labeled as change of expectations of the higher level course instructors and employers from the students is not as much a change of the real expectations of the instructors and employers, but on the contrary it is a change based on what the students are able to offer to them. In the last 10 to 15 years, it is not uncommon to hear complaints that senior (last year undergraduate) students or graduating students "do not know how to design". The possible degradation of the students' design skills, mainly in integrating mathematics and mechanics skills into design implementations, may be attributed to an increased number and variety of required courses that the students have to take during their undergraduate curriculum. Not all of these courses are in the disciplinary area that the students are enrolled in, leaving very little room for exercising the fundamental skills that they learn into design implementations. That is, students barely have enough time to master the topics to use them in an analysis environment let alone use them in design.

Finally, the success of commercially available numerical analysis tools, such as Finite Element Analysis, in the past decade or two has been both a blessing and a potential source of need to change our educational system. The capability to solve highly complex engineering analysis problems with relative ease has made these tools to be an indispensable part of many engineering field practices. As a result, often the faculty members are criticized by industry for not teaching students how to use these codes. This, of course, is a criticism that not many faculty members can sympathize with. Most educators do the right thing and make sure that the students learn the basics of the algorithms and the theoretical limitations of the various features of the tools. Of course along the way a few tidbits about what is important in running those codes are provided so that the users would not generate completely nonsensical results—a situation commonly referred to as "garbage in garbage out". However, while teaching the basics, it may be possible to provide to students some basic information and skills to enable them to use black-box numerical analysis codes in the design environment through appropriate classroom experience. Before using highly complex and advanced Finite Element Analysis tools they should get acquainted with entry level mathematics based structural design to developed some understanding of both possibilities and limitations in the troublesome relation between the physical reality and their (Finite Element) models.

Strengthening and supporting our modernization effort is the TU Delft wide "Focus op Onderwijs" project[5], which aims to modernize both the teaching environment and curriculum content. Its final objective is to improve all teaching material to the current state of knowledge and in line with engineering practice. Revision of teaching methods will introduce state of the art campus wide web based course delivery and management as is for instance provided by the introduction of the e-learning environment Blackboard[6] at the TU Delft.

MATHEMATICA FOR THE NEW COURSE DELIVERY

It was decided the course would be a mix between traditional lectures with in-class tutorials and computer based homework tutorials allowing students to experiment with design variables. These tutorials were set-up using "Mathematica[2]". The choice of Mathematica as a primary tool has one major reason. Mathematica offers a suitable intermediate abstraction level between the relative simplicity of the theory of basic mechanics and the complex knowledge needed to fully understand the intricacies of numerical solution techniques such as Finite Element Analysis. Mathematica also offers a unique "notebook" environment in which text, graphics, mathematical equation building, symbolic manipulation, numerical solutions, and programming can be integrated. In the following some of the elements of the notebook environment will be discussed in more detail.

Notebook environment:

There are several engineering software tools in the marketplace such as Matlab, Excel, Macsyma, that can do many of the features of Mathematica, but historically Mathematica is the one that can handle all of them in a unified fashion. By combining the text and equation building features, entire technical manuscripts such as engineering papers and books can be prepared within the "notebook" environment. This feature will allow preparation of highly structured technical documents that can be read by the students electronically on the web or after printing at their own leisure. For example, a sample notebook for the design of a statically determinate truss structure is shown in Figure 1. What is shown in the figure is the outline of the file, with two main sections and subsections of the notebook, which are encapsulated in, what is referred to as, the "cells" of the notebook. Those cells, which can be expanded by double clicking the cell bar on the right side of the notebook, contain text, graphics, typed equations, and symbolic manipulations.

Various elements of the notebook environment will be discussed in the following subsections using the example notebook shown in the figure, which is used for the design of a statically determinate truss. As the main sections indicate, the first part of the notebook evaluates the effect of the changes of the internal geometry of the truss structure on the stresses in its member as well as its structural weight, while keeping the members cross-sectional areas constant. In the second part of the notebook, the cross-sectional areas are redesigned depending on the member stress level using a simple design criterion for effective use of structural material, commonly referred to as the "fully-stressed design criterion"[8,9].

Graphical representation:

In the example shown in the notebook, the first major section assumes the cross-sectional areas of the members to be specified (all the same) and defines the complete geometry of the truss as a symbolic function of the internal angle θ_A, which is used as a design variable for the problem. The variation of the truss geometry can therefore be visualized graphically for a sequence of internal angles θ_A, and be animated successively. In fact the sketch shown at the top of the notebook in Figure 1 is obtained using Mathematica graphics and corresponds to one of the geometries generated during an animation. Incidentally, for the truss mechanism problem specified in this notebook, the vertical distance between the dashed line at the tip, point **D**, and the horizontal line passing through points **A** and **B** is specified to be fixed. Hence, changing the internal angle θ_A causes the length of the members to change affecting the overall weight of the truss, as well as the internal loads of the truss.

In the following subsection in the notebook, the truss weight, which is used as a measure of the efficiency of the design, is also developed as a function of the internal angle θ_A. The variation of the total structural material volume as a function of the internal angle θ_A, is plotted graphically as shown in Figure 2 enabling the student to choose the lightest weight truss configuration.

The graphical features of the notebook should not give the impression that an independent free input form graphical tool exists in Mathematica. In fact, it is not possible to input a free form graphics unless it is first prepared by an external graphics package and imported into the document. What is possible, however, is to use various Mathematica functions for creating graphics objects such as lines, points, and simple shape objects, and show them within the notebook. Therefore, the truss geometry, which is defined symbolically by specifying the length of the members and the location of the nodes, is used for creating line objects and point objects that are combined to show the truss topology. For engineering graphics the standard Mathematica function "Plot" can be used to generate a functional plot such as the one shown in Figure 2.

Symbolic manipulation:

Symbolic manipulation enables the students (and the instructor) to enter basic mechanics concepts and mathematical relations into equations that can be manipulated to produce solutions to engineering problems that are in parametric form rather than single point numeric solutions. This is partially explained in the graphics section above, where certain quantities are developed symbolically. However, the use of symbolic equation solving is more powerful than that. Using symbolic manipulation the entire solution of a parameterized engineering problem may be derived symbolically. These symbolic solutions in combination with relevant graphics enable us to effortlessly study the influence of multiple design parameters on the results.

For example, for the simple truss problem demonstrated in this paper, it is possible to express the equilibrium equations at various nodes in terms of the symbolic geometry variable(s) and internal force variables, and symbolically solve the values of the internal forces from those equations. A portion of such a nodal equilibrium equation is shown in Figure 3, in which the equations of equilibrium at point D of Figure 1 are solved. First, vertical and horizontal equilibrium equations represented by the symbolic names **EQDy** and **EQDx**, respectively, are written as two equations using the symbolic variables $\mathbf{F_{AD}}$, $\mathbf{F_{CD}}$, and θ_A. In a similar fashion the external forces at node D, **PappDy** and **PappDx**, could also be left symbolically, but in this case they are specified numerically. Out of these two equilibrium equations, we can solve for two unknowns. We choose to solve the equations for the values of the internal forces $\mathbf{F_{AD}}$ and $\mathbf{F_{CD}}$ using the "Solve" command of Mathematica indicated by the construct **Solve[{EQDy, EQDx},{$\mathbf{F_{AD}}$, $\mathbf{F_{CD}}$}]** As can be seen in the last cell of the figure, the solution for the forces is symbolic in terms of the internal angle of the truss θ_A. For any given value of the internal angle θ_A we therefore have the force results without solving the equations again and again.

The symbolic solution of the internal forces enables evaluation and plotting of internal member forces as a function of the variable internal angle θ_A. In fact, in this particular problem of a statically determinate truss, every internal member force can be determined symbolically in terms of θ_A, by writing the other nodal equilibrium equations and substituting the already determined symbolic internal forces from the previously solved nodal equilibrium equations. As mentioned earlier, this feature allows solving the entire problem symbolically and thereby determining all the member forces and stresses as well as nodal reactions at the supports in terms of the symbolic variable(s) intentionally left in the problem.

Numerical solutions:

Extensive numerical solution capabilities, such as solution of system of algebraic equations, solution of differential equations, numerical optimization functions, and eigenvalue solvers, etc. enable us to extend the problem solving capabilities to more complex problems in which symbolic solutions may become either prohibitively expensive to compute or even impossible. Especially minimization and maximization of functions as a baseline approach to optimization enable us to generate better designs when the interactions between various design variables are complicated.

Again, as a simple demonstration of the numerical solution capability, we are able to take the derivative of the total weight function, which is graphically represented in Figure 2, with respect to the symbolic variable θ_A, as shown in Figure 4. The resulting expression, assigned to a variable name **eqn** in the figure, is symbolic in terms of the internal angle θ_A. Equating **eqn** to zero and using **FindRoot** to solve for the root numerically, produces the lowest weight (material volume) truss structure. In fact, the numerical minimization algorithms in Mathematica are not only limited to a single variable, and had we expressed the truss volume in terms of more symbolic variables such as member length AC and distance between points A and B, we would have been able to solve for the best values of those variables which would have produced the lowest weight truss structure.

Programming:

Finally, the programming capabilities coupled with the numerical solution algorithms, enable us to combine various steps of symbolic and numeric analyses into numerical components that can be called just like subroutines or black-box software components which can be incorporated into a design environment requiring repetitive analyses. This feature allows the students to work in a computational environment (as they would with off-the-shelf engineering software) in a fashion similar to the current design practices in industry.

Without questioning the background and the validity of this design approach, it can safely be stated that the current industrial design practices are nothing more than adopting an analysis model, keep tweaking the various model variables and performing the analysis again and again until the design is improved to an acceptable level (or the computational or the time resources are completely exhausted). Although this kind of a practice may not appear to have a strong theoretical basis to be taught in classroom, there are various pragmatic techniques the students can be exposed to which they can effectively use in an industrial design environment. For example, in the current course, after building a numerical model, students are shown how to build sensitivity information. Of course this can be easily accomplished using the symbolic capabilities of Mathematica. However, more importantly, students are shown how to build classic finite difference derivatives using small Mathematica programs that encapsulate their numerical solutions. This enables them to put together solution strategies that mimic the use of commercially available engineering software where industry mostly employs black box Finite Element codes. In contrast using Mathematica allows students to learn the intricacies of mathematical design methods without having to learn the finite element theory in this early stage of their educational program. This way, students will quickly learn to build their own design models and obtain results relevant to those designs. They will gain insight in working with complex mathematical tools such as Finite Element Analysis before actually using them.

PRACTICAL ASPECTS: TUTORIALS, QUIZZES, AND EXAMS

All course material such as Mathematica notebooks together with PowerPoint based lectures and other supporting material is made available by way of the e-learning environment Blackboard[6], used throughout the TU Delft. Apart from some notes no printed lecture material was used. Mathematica was made available on the Faculty network allowing all students to run through the notebooks and create their own optimal designs. This gives the students the opportunity to work through the problems presented in class and further their understanding of the design issues in their time on their terms, making the course suit their working habits better.

While the course is running, a number of intermediate randomized quizzes are made available for limited periods of time via the Blackboard server. The quizzes offer students the opportunity to both exercise their skills and generate credits towards the final course grade.

Upon conclusion of the course students currently take a written exam. The final grade for this course is the weighted average of the intermediate quizzes and the result of the written exam. In future the written exam will be replaced by a more conforming electronic hands-on evaluation using randomized problems within the Mathematica and Blackboard environments.

RESULTS

Although this year was the first year this course was run at TU Delft, we are encouraged by the results. Blackboard's user statistics showed that students were interacting with the course material from the start. Our students diligently carried out the optional randomized quizzes giving during the course, which is an exception in the consumer based Dutch student mentality. Mathematica notebooks were downloaded en mass by the students so they could work through the problems again in their own time.

The exam results were encouraging. Considering this was a new course in a totally new setting than what our students were used to, the pass rates do not differ from normal second year pass rates at TU Delft. It is reasonable to expect that the pass rates could even go up once the word has spread how user-friendly this subject is to study using the notebooks.

The long-term effect on our structural design education will not be known for another few years. Using the various quality control systems in place at TU Delft we will of course be closely monitoring the situation and eagerly await the results. It is anticipated that, within a few years, improvements in the quality of the structural design component in the later years of the curriculum can be witnessed.

APPENDIX: FACULTY OF AEROSPACE ENGINEERING AT DELFT UNIVERSITY OF TECHNOLOGY

The degree of Aerospace Engineering[3] at Delft University of Technology (TU Delft)[4] exists since 1940 and Aerospace Engineering has been an independent faculty since 1975. It currently has some 1700 students enrolled in their Bachelor and Masters programs. Students graduate with a Bachelors of Science degree in Aerospace Engineering, which is internationally recognized (ABET), and many continue on to obtain a Master of Science degree in Aerospace Engineering.

BIBLIOGRAPHY

1. Saunders-Smits, G.N. and De Graaff, E., The development of integrated professional skills in aerospace, through problem-based learning in design projects, Proceedings of the 2003 American Society engineering education, Session 2125, June 2003
2. www.wolfram.com - Official Mathematica website
3. www.lr.tudelft.nl - Official Faculty website
4. www.tudelft.nl - Official University website
5. www_en.icto.tudelft.nl - TU Delft ICT in Education website
6. www.blackboard.com - Official Blackboard website
7. Ketzer, Jan W., Audiovisual Education in Primary Schools: A Curriculum Project in the Netherlands, National Inst. for Curriculum Development (SLO), Enschede (Netherlands), May 1987
8. Niu, Michael C.Y., Airframe Structural Design, Hong Kong Conmilit Press Ltd, 1988
9. Bruhn, E., Analysis and Design of Flight Vehicle Structures, Jacobs Publishing, Indianapolis, 1973

GILLIAN SAUNDERS-SMITS

Gillian Saunders-Smits obtained a MSc. in Aerospace Structures and Computational Mechanics from the Faculty of Aerospace Engineering at Delft University of Technology in 1998. After a short period in industry, she returned to the Faculty of Aerospace Engineering in 1999 as an assistant professor. Since 2000 she is the faculty's project education coordinator and teaches Mechanics and is currently doing a PhD in engineering education.

ZAFER GÜRDAL

Zafer Gürdal is a jointly appointed Professor of Aerospace and Ocean Engineering, and Engineering Science and Mechanics Departments at Virginia Tech, where he spent the last 19 years at various faculty ranks. He is currently appointed as the Aerospace Structures chair holder at Delft University of Technology based on a special agreement between the two institutions.

JAN HOL

In 1983 Jan Hol obtained his MSc in Design and Analysis of Aerospace Structures from the Faculty of Aerospace Engineering at Delft University of Technology. After working several years as an application consultant specializing in Finite Element Analysis he returned to work on the development of DISDECO. From 1988 onwards he teaches Finite Element Analysis and does research in collapse behavior of imperfect thin-walled shell structures.

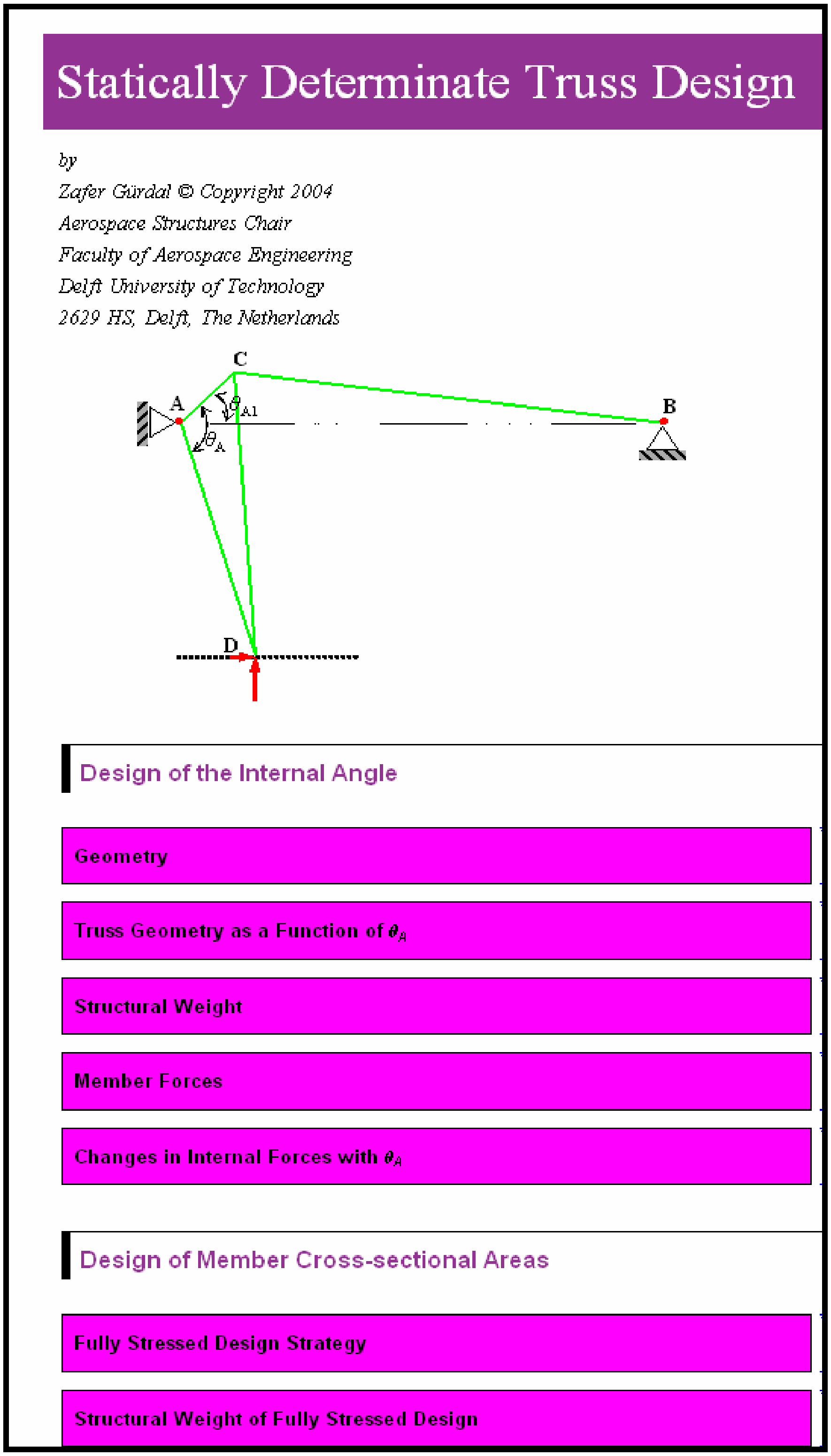

Figure 1: Mathematica notebook for a statically determinate truss design.

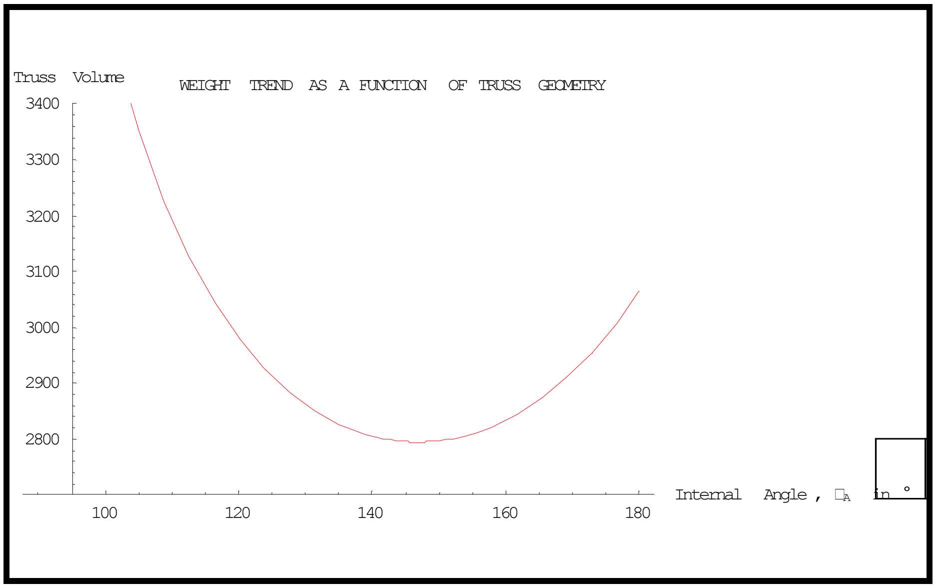

Figure 2: Graphical representation of truss material volume.

■ **Nodal Equilibrium - D**

```
In[46]:= PappDy = 10000.0;
         PappDx = 1000.0;
```

Vertical Equilibrium

```
In[48]:= EQDy = PappDy + F_AD Sin[θA2 Degree] + F_CD Sin[ (θA2 + θD) Degree] == 0

Out[48]= 10000. + Sin[° (-60 + θA)] F_AD + Sin[° (-60 + 57.2958 ArcCsc[
           0.05 √(400. - 3200. Cos[0.0174533 θA] Csc[0.0174533 (-60. + θA)] + 6400. Csc[° (-60. + θA)]^2 Csc[0.0174533 θA]) + θA)] F_CD == 0
```

Horizontal Equilibrium

```
In[50]:= EQDx = PappDx - F_AD Cos[θA2 Degree] - F_CD Cos[ (θA2 + θD) Degree] == 0

Out[50]= 1000. - Cos[° (-60 + θA)] F_AD - Cos[° (-60 + 57.2958 ArcCsc[
           0.05 √(400. - 3200. Cos[0.0174533 θA] Csc[0.0174533 (-60. + θA)] + 6400. Csc[° (-60. + θA)]^2 Csc[0.0174533 θA]) + θA)] F_CD == 0
```

```
In[51]:= sol = Solve[{EQDx, EQDy}, {F_AD , F_CD}]

Out[51]= {{F_AD →
   -10000. Csc[0.0174533 (-60. + θA)] + (Csc[0.0174533 (-60. + θA)] (10000. Cos[0.0174533 (-60. + θA)] + 1000. Sin[0.0174533 (-60. + θA)])
      Sin[0.0174533 (-60. + 57.2958 ArcCsc[0.05 √(400. - 3200. Cos[0.0174533 θA]
                Csc[0.0174533 (-60. + θA)] + 6400. Csc[° (-60. + θA)]^2) Csc[0.0174533 θA]] + θA)]]) /
     (-1. Cos[0.0174533 (-60. + 57.2958 ArcCsc[0.05 √(400. - 3200. Cos[0.0174533 θA] Csc[0.0174533 (-60. + θA)] +
                6400. Csc[° (-60. + θA)]^2) Csc[0.0174533 θA]] + θA)] Sin[0.0174533 (-60. + θA)] +
      1. Cos[0.0174533 (-60. + θA)] Sin[0.0174533 (-60. + 57.2958 ArcCsc[0.05 √(400. - 3200. Cos[0.0174533 θA]
                Csc[0.0174533 (-60. + θA)] + 6400. Csc[° (-60. + θA)]^2) Csc[0.0174533 θA]] + θA)]),
  F_CD → -(1. (10000. Cos[0.0174533 (-60. + θA)] + 1000. Sin[0.0174533 (-60. + θA)])) / (-1. Cos[0.0174533 (-60. + 57.2958 ArcCsc[
            0.05 √(400. - 3200. Cos[0.0174533 θA] Csc[0.0174533 (-60. + θA)] + 6400. Csc[° (-60. + θA)]^2) Csc[0.0174533 θA]] + θA)]
       Sin[0.0174533 (-60. + θA)] + 1. Cos[0.0174533 (-60. + θA)] Sin[0.0174533 (-60. + 57.2958 ArcCsc[
            0.05 √(400. - 3200. Cos[0.0174533 θA] Csc[0.0174533 (-60. + θA)] + 6400. Csc[° (-60. + θA)]^2) Csc[0.0174533 θA]] + θA)])}}
```

Figure 3: Symbolic solution of the equilibrium equations at a node.

An accurate computation of the minimum weight will require taking the derivative of the weight, equating it to zero to solve for the value of the ${}_{A}$.

```
eqn = D[trussVolume[LAC, LBC, LAD, LCD], θA]
```

```
10
 (-80 ° Cot[° (-60 + θA)] Csc[° (-60 + θA)] + (55.8505 Cos[0.0174533 θA]
       Cot[0.0174533 (-60. + θA)] Csc[0.0174533 (-60. + θA)] -
      223.402 Cot[° (-60. + θA)] Csc[° (-60. + θA)]^2 +
      55.8505 Csc[0.0174533 (-60. + θA)] Sin[0.0174533 θA]) /
    (2 √(400. - 3200. Cos[0.0174533 θA] Csc[0.0174533 (-60. + θA)] +
         6400. Csc[° (-60. + θA)]^2)))
```

```
? FindRoot
```

```
FindRoot[lhs==rhs, {x, x0}] searches for a numerical solution to
   the equation lhs==rhs, starting with x=x0. FindRoot[{eqn1,
   eqn2, ... }, {{x, x0}, {y, y0}, ... }] searches for a
   numerical solution to the simultaneous equations eqni. More...
```

```
bestAngleWeight = FindRoot[eqn == 0, {θA, 140}]
```

```
{θA → 146.772}
```

Figure 4: Numerical solution of the minimum weight design.