MCE 511 Computer-Aided Design Introduction, Scope and Ground Rules

Instructor:
OA Fakinlede www.oafak.com
September 2017

Introduction

Welcome to the final year undergraduate course on Computer Aided Design in mechanical engineering. Many of you are here because this is a required course for the completion of your graduation requirements. There may be other reasons. Some of you have read the course outline as specified in your prospectus. Are you excited about that? Do you really believe that in two hours a week for a term we can cover the scope indicated? Some have not bothered to read but know they have no choice but to go through the motions and obtain the regular graduation grades to get out of this place!

This introductory class is to be used to clarify issues that may prove important, not only for this particular course, but in your attitude to the use of computers in design and your education in general. I will also be making recommendations for your continued education in Computer and the way it can aid design and your career as an engineer. We are going to gauge expectations at the beginning of this course and hopefully, you will still be in the mood to allow your experiences to be gauged at the end.

As the class is composed of aspiring Engineers, it is pertinent to ask,

- Why do Engineers use the computer?
- What does it mean to design?
- What software platforms are most suitable?
- What packages should they learn?
- What are the consequences and results of good programming and software training?
- Do you therefore become programmers as a result? Will you still be engineers?

We can ask these and other questions. I must first draw your attention to the fact that very few people can answer these questions accurately. If that looks negative, the next statement I am going to make is even worse: You will need to have a good background and some understanding to benefit from the right answers! Looks then like an impossible proposition? Not necessarily. That is what we shall seek to clarify in this first week.

Why do Engineers learn to program?

Science is the database of all the knowledge we have about the world, outer space, and ourselves. This database is modified, expanded, and constantly updated by the scientific method. **Engineers** rely largely on the **methods and findings of Science** to create, facilitate, and **orchestrate technology**. Engineers do not necessarily engage themselves in all the quests of science. For a scientific problem to be of engineering significance, it will usually also have some economic importance that allows it to be eventually transformed into **products and services**. The creation of products and services is technology. If your efforts do not result in products and services, what you have done is NOT technology. When such problems are well-defined, computers, computational methods and software packages may often be used to solve them or these can be an essential part of a solution that had its roots in other skills and applications. Engineers can learn to use the computer to solve problems. The most important thing is to focus on the problem to be solved. Very often, there are several different ways computers can be used in solving problems. It is an engineering decision to find out the most relevant or most productive of the variety of approaches to use. Let me make it more practical:

Imagine you are hungry. You need a meal. What choices do you have? Let us for the present forget about costs. Assume you have sufficient means to eat whatever you want. Is there still a problem? Of course, there is the problem of deciding between alternatives! Let us look at how many ways you can arrive at a warm plate of rice, beans, and fried fish stew:

1. You could look for a river; arm yourself with fishing hooks and nets so to catch fishes. Climb a palm tree and take some nuts to extract oil from them or buy some cooking oil and other ingredients. Cook the rice, cook beans and get yourself a good lunch.
2. You might have a packed canned meal which you only need to open in a tin and warm to eat.
3. You could go to “Tantalizers”, Dining Hall or some other less known joint and ask for what you like.
4. Call mommy!

Now, let me ask you, which of these methods is the best way to have your lunch? Simply choosing or arguing that one way is better than another, as we are wont to do, can be misleading. It really depends on who you are. A woman with a family of six may prefer #1 with the modification of getting the fish from the market. A soldier at the battle front has no use for #1 or #3. Only #2 is available and there can be no question of what to do. Yet a Landmark student may elect #3. Children at least know that #4 works! We see here that the kind of person and the circumstances may make on choose one method or another. In this context, we can begin to see that some questions are not so easy to answer!

What does it mean to “design”?

To design is to properly specify, quantitatively the materials, parts and assembly of a product to meet a service. Engineering training equips you with the theoretical knowledge and the knowhow to design products that will serve their intended purpose. A civil Engineers already knows that the structural details of a building are sufficient to make it stand and perform adequately before it is built. A mechanical engineer already knows how a trailer to be pulled by a truck will shake and vibrate on the expected road surface before it is manufactured. It is a mistake to think that your design training is embedded in the “Design” courses that you take. Design is much more than that as we shall see.

I got a WhatsApp message yesterday from a friend who had several plastic buckets in his house that had broken handles. He was wondering why these handles were breaking and wanted to know what the engineering problem was. Is that a fake product? Why is it that buckets brought from other countries don’t break like that? What we call fake products are products that were not properly designed to provide the services demanded of them; they may be properly designed but not properly manufactured. they fail in service and we call them “fake”

Once again, this is may be engineering failure. It may also political failure. A product that has not met requirements ought not to be sold to the public! There are authorities that should not allow products to be in the public space unless they are guaranteed to perform intended services reliably. Where are the people (ir)responsible for ensuring public safety? How did those products get through their oversight systems?

When a course is called “Mechanics of …” you are being given, in that course, the theoretical background to make design decisions! The courses that are named “Design” are expected to harvest that knowledge and show you how you make products with them. If you have zero theoretical knowledge and get “A” in a design course, you have been trained to be a good artisan or welder – not an engineer. The products you will make (if you ever do) are fake products because you did not learn the theoretical bases of product design!

This course necessarily invokes the knowledge you have in all those mechanics courses especially mechanics - of solids, - of fluids, - of heat, vector and tensor analysis, computer programming, graphics etc. The concurrent course on the Mathematical Theory of Elasticity is offered as an option this term. That is another opportunity to get your background right.

What software packages are most suitable?

There are several perspectives or programming paradigms in use. Some of these developed with the history of computing as new challenges led to different ways of approaching the complexities that are involved. There are also many programming languages and software packages with new ones being created each year. Answering questions on which language or paradigm is like shooting at a moving target. Obviously, answer is not easy to get. In this course, one of the things we shall try to do is to explain different programming paradigms. To make an intelligent choice, you need to know the similarities and differences between the paradigms. We will see how drafting tools such as AutoCAD have been excellent at helping your 2-D sketching – Computer Aided Drafting. We will explore 3-D parametric modeling and see how we can reverse the drafting process. We will how we can create a solid model first, generate the two-dimensional views as well as sections at any possible viewing angles from the model. We will get an introduction to symbolic and computer algebra systems. We will see how that can serve as an alternative to the way we look at computer programming. And I hope well trained people will not easily get carried away by unlearned paradigm comparisons as we shall see how the solution of one problem creates another!

The course will revise the concepts of Numerical Analysis you took in earlier courses. We shall be very practical and see that these tools, are designed to make engineering analyses simpler and effective and efficient. You will be introduced to Finite Element Analyses as the most successful numerical tool in engineering design today. You will see that the theoretical background needed to understand such tools are embedded in the courses in solid, fluid and thermal mechanics you have already taken. You will find, for example that the Von Mises and Coulomb stresses you learned about in your "theories of failure" last term are necessary to give proper instructions to your FEA package and to correctly interpret the results. You will see that Bezier Curves, Splines and NURBS and T-Splines in the "sculpt" tool have their history rooted in Numerical Interpolation.

AutoDesk Fusion 360?

After studying your syllabus and examining several possible alternatives, I have elected to use AutoDesk Fusion 360 as the main software package for this course. We selected this package after considering others such as **Solid Works**, Siemens **Solid Edge**, Autodesk **Inventor**, **Pro Engineer**, and several other packages. Fusion 360 is a SaaS and it allows you to eventually have the capability of doing PaaS. (We will look at these acronyms shortly. Fusion 360 environment allows us to do the following:

1. **3-D Parametric Modelling**. We will extend the background you already have with 2-D sketching in AutoCAD and you will be creating 3-D parametric models instinctively. As you imagine your designs in the mind, it will be obvious that working in 3-D is far superior and more efficient that looking at three 2-D views. Parametric models have the added advantage of sensitivity analyses. Parametric modeling means that the model is defined by parameters; a change of dimension values in one place also changes other dimensions to preserve relation of all elements in the design. Properly executed models allow you to change crucial dimensions and properties, get new designs to analyze and optimize thereby saving you a lot of time!

2. **Assembly**. It allows you to create an assemblage of components. In fact, you will learn to create products by breaking them down into components for assembly. The assembly facility in Fusion 360 also allow you to animate possible movements between the parts and components of your design virtually. Such animations may be subject to service constraints that will be imposed in actual implementations of the design. You are also able, not only to import your own components but to actually go out to the market place, select real manufacturer components, test how they work with your design and even place orders from your computer!
3. **Simulation.** You will be able to do Finite Element Simulations on components or your entire assembly. You can do linear and nonlinear stress analyses, Modal Analyses on Vibratory Systems as well as Thermal and Thermal Stress Analyses.
4. **Design Optimization.** From the results of the simulation, we will be able to move, within the package to design analysis and optimization. Areas of negligible stresses can be candidates for material reduction while highly stressed areas will be enhanced to improve safety results for the design in service.

Fusion 360 is also designed for additive manufacturing of Computer Printing as well as subtractive manufacturing. The unavailability of 3-D printers will not allow us to fully explore some of these capabilities. We are also constrained by the limitations in time, It is a fact that each of these four areas could take the entire class time!

The ***Mathematica®*** symbolics system will be used to demonstrate the issues around language Interpreters, Functional Programming and Rule-Based programming. The powerful ***Mathematica®*** system will relate to many of the problems you encounter in your study of engineering here and can be a veritable tool for your other courses. The graphical prowess as well as Numerical, Algebraic and List processing available in such a powerful interpreted environment will make language comparers shudder. And why make a choice? **Tuum est!** It's all yours.

SaaS and PaaS

Fusion 360 is packaged as a **SaaS**. It is acronym for "**Software as a Service**". It is a great benefit to you in many ways and I will seriously urge you to take the opportunity to master this tool very seriously. To understand what we mean, let us go into a bit of history.

Computer Graphics has a history that parallels that of the computer itself. While early efforts began more than 60 years ago, the availability and democratization of computer-Aided Design software became practicable only when the PC became popular in the early 1980s. The first versions of AutoCAD, and early implementation for the PC environment cost nearly three thousand dollars 1982. Lower level and more programmatic approach to computer graphics were available to capable programmers but these had limited penetration as the programming capacity to use them invariably limited them to professional programmers who work at the back end in developing applications for professionals in other fields such as engineering, architecture and Industrial design. Two of the most popular Application Program driven graphics libraries today are OpenGL and DirectX.

3D wireframe features were developed in the beginning of the sixties, and in 1969 MAGI released Syntha Vision, the first commercially available solid modeler program. Solid modeling further enhanced the 3D capabilities of CAD systems. NURBS, mathematical representation of freeform surfaces, appeared in 1989 -- first on Silicon Graphics workstations. In 1993 CAS Berlin developed an interactive NURBS modeler for PCs, called NöRBS. The 3-D parametric modelling tools became viable competitors in the mid 1990s with Solid Works, Sold Edge, Pro-Engineer and Autodesk Inventor as some of the market leaders. In the course of time, these packages added commercial level Finite Element Numerical Analyses tools with the packages. This lead to more useful packages for computer Aided Design. Prior to these development, what were called CAD were essentially glorified drafting tools.

Three Problems

Industry standard CAD packages pose two major problems of acquisition to us. The first is that **cost of purchase** can easily outstrip the laboratory budget of a typical Nigerian Engineering faculty. One or two hundred and fifty thousand dollars may not be sufficient to equip a lab of a hundred students with required software. It was therefore a welcome change that nearly 12 years ago, some multinational graphics software companies started offering academic licenses for these packages free to universities worldwide. In cases like Solid Edge and Autodesk, such licenses were full commercial licenses that were made free for academic use.

This led to the second problem: **Hardware limitations**. A properly configured desktop of today with 8G RAM and a Graphical Processing Unit (GPU) is sufficiently capable of a simple 3-D solid model. If we proceed from there to Finite Element simulation, then we have problems as nothing short of chaining all the computers in a 100-seat lab can truly simulate a real-life engineering design. The hardware requirements were just massive.

A SaaS with free academic license like Fusion 360 solves both problems creating two new ones! A SaaS is usually designed to live in the cloud. As a cloud-based software, you can model your design locally, store them in the cloud and have a choice of doing your simulation locally or in the cloud. As Fusion 360 is completely free for recognized academic usage, the next two worries are Internet connectivity and cloud credits.

A university like Landmark University provides connectivity to its students. Fusion 360 offers, for free, cloud credits that could easily have cost us one thousand dollars per student. For there to be a real competition, any other software must offer us these three: free software, Access to remote computers and cloud credits. These are all needed to make it work!

This is how the SaaS paradigm works for the software package we shall use. The goal of the company is to make the design and manufacturing cycle to easy that manufacturers can so quickly improve designs in response to customer demands and therefore offer the products as a Service – PaaS. In getting on this train, you are probably a first adopter of this new paradigm and it may make a significant impact on your outlook and ability to function as a modern engineer.

Fusion Competition

It is a good idea to plan to make your knowledge of design gained in this course count towards your final year project. This is empowerment to excel. You should further note that you can submit your designs for competition to Autodesk. If they publish your idea, you will earn some cash! Can you beat that? Ideas rule the world. Take this as an opportunity to belong to the class of elite ideas creators in the world. The poverty in our land is essentially a poverty of ideas!

What you will be given

Fusion 360 software is already available for installation on your systems at the CSIS. After installation, you will be instructed on the way to get yourself a personal license directly from Autodesk. The beautiful thing about this arrangement is that this license is for three years! You will start your learning now, continue until you finish your final-year project. This takes you through another year of NYSC and a full year after that to become a master of 3-D parametric modelling! You can therefore take the course as the spark for a three-year learning process by which time a serious student ought to be a master in this art.

The Department promises to provide the following for you:

1. A fully licensed copy of Fusion 360 as we have highlighted;
2. Unfortunately, we were not able to get individual licenses here. There will be 15 to 20 computers in a designated laboratory with the latest version of this symbolics processor. By the time I have demonstrated the usefulness and importance of this software to you, you may be ready to take advantage of a secret way to get your own personal copy. Wait a minute, I am NOT suggesting software piracy here. Wolfram Research, the makers of Mathematica, in a wonderful gesture of commitment to the educationally disadvantaged world, make version 10.2 of this software freely available on the famous Raspberry Pi. This $2,000.00 software is now available to you if you buy the Raspberry Pi. JUMIA and Konga are selling RBP computers and I will greatly encourage every student to get one;
3. **Required documentation and books**. Most of the reference materials you need will be available in PDF form. There are hundreds of videos training in Fusion 360 on the web. Do you still need me? We shall see. The training on the web is essentially for hobbyists, industrial designers and general users. I will encourage you to get all the training you can get on the web. Some are very good while others, like any wiki idea, are made by people who are themselves still learning.
   What you need is far more specific and deeper than what the web teachers offer. We

want you to use Fusion 360 not just to produce pretty models but to do engineering design. The web may fall short of helping you understand the results of your Finite Element design and analysis. The theories you need here will partly be given to you in MCE 511 but you will need MCE 534 and 535 to understand the theoretical basis and implications of the result of the embedded commercial grade NASTRAN FEA package. Most o the web trainers do not have sufficient background to lead you into that. If you do not pay attention to the opportunity you have now, you too will suffer from that same limitation! We will learn sufficient Tensor Analysis and Continuum theory in MCE 534 while the qualitative description of the FEA process will be given in MCE 511. These are the part of the needed background the web teachers will not usually reach. That is the reason you are in a university!

4. Class notes and program will be available on our local Intranet and can be downloaded to your tablets in class for those who have telephones with sufficient memory and Bluetooth.

What will you do with these?

Life may be hard in Nigeria. However, I want to give you this simple challenge: If there is someone in the best university on earth that can design better than you at the end of your studies in Landmark, it is NOT because they went to a better university. It is not because they lived in a better country. It is not because they were taught by better teachers. **IT IS BECAUSE YOU REFUSED TO WORK HARD FROM THE BEGINNING**! If you work consistently hard, and you take the challenges around you, there are sufficiently exciting things to do, to grow your knowledge exponentially and make you competitive with anybody else on earth! The opportunities to excel are here. Don't waste them. The benefits that can accrue to those who work very hard include:

1. Final year project can become more meaningful,
2. Industrial attachment and NYSC period more rewarding just by using the facilities that are provided for you.
3. Doors will open in the future and you will be surprised how a single course can influence the course of your life.

In view of these, I have nothing but pity for the students who miss the opportunity given here to improve and equip themselves for a brighter future.

The following mind attitudes will be useful.

1. **Always define and focus on your problem**. At the beginning, you will learn many jargons, definitions, mechanical ways of getting specific things done. In the end, it is your ability to

relate what you are trying to do to real-life problems that are well defined that really show you are learning anything. Always define the problem that you are trying to solve.

2. **Think First, Do Rough Hand Sketches, Code and Execute Later**. There is nothing so disheartening as a student coming to the practical class with no sheets of paper or note containing a description of what they want to implement. Such a student cannot be described as serious. For every hour you will spend on the computer to write code, spend at least two thinking out what you want to achieve.
3. **CAD is only learned by doing**. You have not learned any concept until ***you have implemented*** it yourself. You will see that I constantly supply you with examples for each concept taught.!

Absence from Class

The full course outline is given to you at the beginning of the term. All the references are available to you. The necessary software environments you need are also given you. Make up your mind. If you are taking the course, read and fire on! A student who has gone over the day's course before the class begins will usually ask better questions and participate more intelligently in class. Such students don't need to spend as much time as bookworm layabouts will need to do later. If you are unavoidably absent make sure you know what was covered in class. The university has rules about attendance. Those who do not meet these rules will be **PREVENTED** from writing the final exams. It is important to know this at the outset and avoid unnecessary problems.

Lateness to Class

It is most discomfiting to have students disturb class by coming late and interrupt the lecturer by pretending to greet! If you want to exchange pleasantries, come ten minutes early and do that to your heart's content. Once lectures begin, you MUST have an extraordinary reason to be admitted. That is if you find the door open! And please don't greet me if you are late to class. Quietly sneak into your seat and I will pretend I did not see you. I will take your greeting if you inflict t on me as an assault!

Interacting with me

You can get my notes and slides from your university website in your normal way. All notes written by me will be available on my personal website. I encourage you to use it. The address www.oafak.com is also at the beginning of this note. My email address is oafak@unilag.edu.ng and fakinlede.omotayo@lmu.edu.ng. Please use the email only as a last resort. My preference with students is that our communications benefit the class. Personal communications should be avoided.

There are two things you should note: I expect you to read the notes and slides for the next class ahead. This should be done even if you are unavoidably absent from that particular class. It is a mark of unseriousness to be asking me questions when you have not availed yourself of the provision and explanations made for you. Reading ahead means you can ask good questions that will benefit the class at class time.

You can also ask me questions on the webpage. If I discover, by the number of questions asked that a class coverage is problematic, I will be willing to do more work on it for the benefit of the class. I am a lecturer and not a grade alterer. If you are hoping for good grades, do your work and you will get good grades. Don't wait for the term to end before you realize you need good grades! At that time, it is too late!

You can also make comments on my webpage. There is room for disagreement but no room for being disagreeable. If everyone in a place are always agreeing, I believe that perhaps only one of them is thinking. I therefore do not mind being told if there are things I do that you do not agree with. I believe that any interaction between adults is a teachable moment if we are all honest with one another!

References

1. **Popular Programming Languages**,(2010) http://www.scriptol.com/programming/choose.php
2. **Comparison of Programming Languages,** (2001) http://en.wikipedia.org/wiki/Comparison_of_programming_languages
3. **Alfred Thompson**,(2007) "Legard's Programming Proverbs", http://blogs.msdn.com/b/alfredth/archive/2007/02/08/programming-proverbs-1-define-the-problem-completely.aspx