30 Years of Calculation

Examining the history of software in mechanical engineering

Dr. Stefan Beermann, CEO, KISSsoft AG

The Early Days

Calculating strength has always been part of engineering. However, the introduction and widespread use of powerful computers has brought about significant changes to the character of these calculations. In the beginning, computer-aided sizing primarily involved Finite Element (FE) modeling. One of the very first uses of these newly available tools was to analyze the NC machines used to manufacture the Apollo modules: NASA required proof that the machines were rigid enough to satisfy their stringent requirement for accuracy. This proof was provided by an FE calculation. In the 1970s, FE methods were implemented for gear problems. At that time, these types of calculations usually took 24 hours, running on a VAX 6400 mainframe.

Development of Standards for Calculation Methods

Although machine element calculations are, in themselves, less time-consuming than FE modeling, there was also a developing trend towards using computers to run them. As customers, inspection organizations, or government bodies often required strength verification for parts, it was essential that the strength calculation programs could perform these calculations according to recognized methods. These methods included ISO and DIN standards, VDI guidelines, and the approved technical literature. However, with regard to technical literature, a distinction must be made between well-defined calculation methods such as in Niemann/Winter (Ref. 1) and methods that are better suited for rough calculations such as Roloff/Matek (Ref. 2).

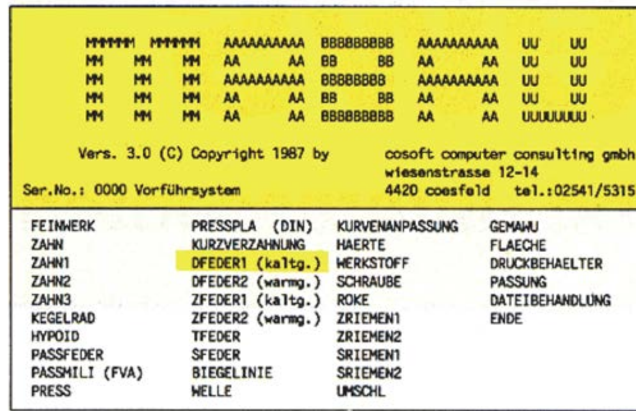

Figure 1 Selection menu from a classic calculation program.

Although DIN standards have a long history (for example, the first edition of DIN 5481 dates from 1940), they addressed geometry exclusively. The first standardized strength calculation was DIN 3990 (December 1970), which described a method for calculating the strength of cylindrical gears. This was the first, crucial step and was followed (albeit very slowly) by other calculation standards. It took another 30 years (DIN 743 in 2000) before the first comprehensive regulations for rotating shafts were issued.

One of the main obstacles preventing the implementation of calculation standards in a program of sufficiently high quality was the problem of how to display the calculation rules. As the guidelines were originally designed for engineers, who worked with slide rules and calculated proofs by hand, most of the data was provided in diagrams (sets of curves) so it could be read easily. The associated formulae were never specified, which meant that programming these curves took a great deal of time and effort — because the data had to be extrapolated from these diagrams and interpolated with higher-order polynomials. The approach taken by simpler programs — in which the user had to read the values from the diagram and then enter them in the program — cannot be considered acceptable.

From 1985 onwards, the standards did become more programming-friendly, because they listed the formulae along with the diagrams. This also made it possible to implement the standards more accurately and reliably in software programs.

DOS Programs

The first calculation programs that worked with the machine element concept were in-house developments by large companies, designed to meet their own requirements. These compa-

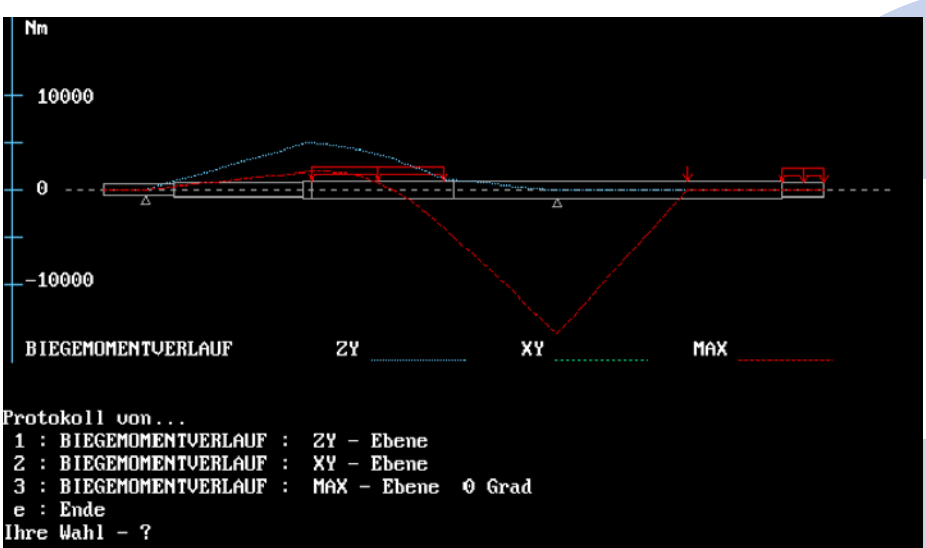

Figure 2 Typical graphics output of the DOS era.

nies generally used the computers and programming languages they had available at the time. Commercially developed programs only really became widely available with the introduction and widespread use of IBM-compatible PCs. These generally ran the DOS operating system, with *BASIC* and *PASCAL* as the most commonly used programming languages. For example, *KISSsoft* was originally created on a Commodore PET, in *BASIC*, for in-house use by the company L. Kissling & Co. AG. However, before the program was released for general sale, it was ported to *Quick-BASIC* running under DOS.

One of the first commercial programs, *Mabau*, from the German company CoSoft, was also initially developed for Commodore computers (Commodore 8000). The first version of the gear calculation program had three parts: presizing, strength, and geometry, because it did not have enough memory to run a program that handled everything. Its user interface (Fig. 1) recalls the days of mainframe computers.

The 1980s saw a great expansion in the number of programs, because the demands made on user interfaces were still relatively low. Many engineers had learned programming during their studies, most of them probably running in batch on mainframes. They were then able to apply this knowledge to programming in the PC development environment (which was almost unbelievably convenient, in comparison to mainframes), and create solutions to a multitude of problems in mechanical engineering.

A feature of these early programs was the sequential querying of input data, followed by the question "Input correct? (Y/N)" which either triggered the calculation or sent you back to the start to input all the data again. Even then, one of the handiest functions was that you could simply press <Return> to confirm your entries. Programming a plausibility check for these entries was relatively easy, because the programmer could tell which entries were available at any stage in the program. However, it was not possible to interrupt a predefined process. This was very inconvenient, because it meant all the additional data required at the end of a query sequence had to be input again, starting from the very beginning.

First Borland and then Microsoft implemented support for window technology under DOS. However, the more important mechanical engineering programs did not implement this technology extensively. Development in this area was stopped by the dawn of the Windows programming era.

In the German-speaking world, the most widely used programs in this period (1985-1990) were *Mabau*, *Hirnware*, *TBK*, *Hexagon* and *KISSsoft*.

Porting to Windows

Porting to Windows was a real milestone in the evolution of every calculation program in use today. Up to the mid-1990s, DOS was the most commonly used platform. As these calculation programs were most often programmed by mechanical engineers who knew something about IT rather than by actual software specialists, there was a degree of reticence when it came to converting to a different operating system. Here Windows provided the option of

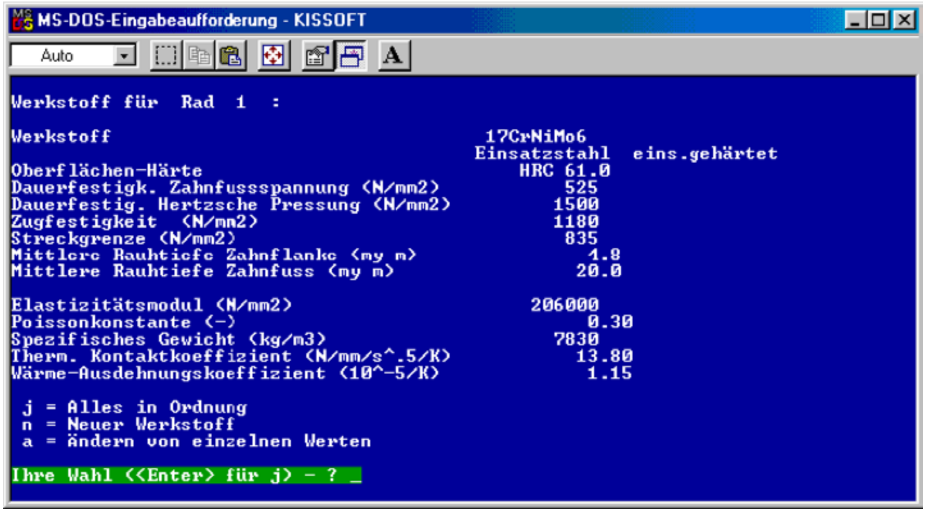

Figure 3 **DOS programs processed keystrokes immediately. Consequently, nearly every screen ended with a prompt asking whether the user's inputs were correct (KISSsoft shaft calculation 1995).**

delegating a whole range of things that would take a great deal of time and effort to program to the operating system: it provided powerful commands for graphics (see the DOS screen display in Fig. 2), enabled the use of a system printer (under DOS you had to program a separate driver for every printer), and greatly simplified file handling. This was all well and good, but only in theory. In practice, the programming effort required to display a line on screen under Windows was much greater than under DOS. As a consequence, every program producer grossly underestimated the effort required to port DOS to Windows: This was the beginning of hard times for providers and customers alike. According to bold announcements, a new version for Windows would soon be available, but was always postponed for some reason or other. Then came the first version — mostly a mish-mash of DOS and Windows which proved to be highly unstable. Some program manufacturers had the idea of keeping the old DOS programs just as they were, and simply putting a user interface on top. Although this provided results very quickly, it soon led to a dead end in development.

Attempts to implement genuine system porting went in two main directions: one direction involved completely restructuring the user interface, whereas the other tried to rescue as much as possible from the old DOS user interface. For existing customers this had the benefit that they did not have to come to grips with an entirely new user interface. Unfortunately, the result was also a program which failed to meet any of the Windows standards.

The most important programs which were able to make the transition to the world of Windows are *Hexagon*, *Mdesign* and *KISSsoft*. These were joined briefly by *Delphi*, which has since disappeared from the market.

Forms or Dialog Technology

Real Windows programs use two main approaches: Form-based programs such as Softwert's *Delphi* or Tedata's *Mdesign* are well suited to running simple calculations. The user interface in these programs looks very like a calculation report (Fig. 3). They also use a top-down workflow, as did the first DOS programs. Every calculation has its own form (key verification, key sizing, etc.). And as long as the entries fit on two screen pages, it is easy to keep the form's content clear in your head. However, the limitations of using forms quickly become clear when you use calculations that require more entries, or where different data variants have to be input.

The vast majority of programs use dialog technology. In other words, you input the parameters required for a calculation in a series of dialog screens. This has the benefit that you can sort the parameters according to group. In principle, you can also input your data in any sequence. In addition, bespoke solutions can be implemented to resolve specific problems. To achieve this, *KISSsoft* has a uniform button, which appears in every screen. This is used to either optimize the values or define them according to specified criteria (Fig. 4).

The Third Dimension

In the last decade CAD programs have increasingly converted to using 3-D. The use of 3-D displays has now become standard in calculation programs, although initially these displays were just an "added extra," because the calculations were performed completely independently of the 3-D display. Although it is debatable whether a three-dimensional image of a cylindrical gear actually provides more information than a 2-D drawing, these images are good to look at, nonetheless.

However, a new interest has come to the forefront in recent years: the direct display of complicated toothing in three-

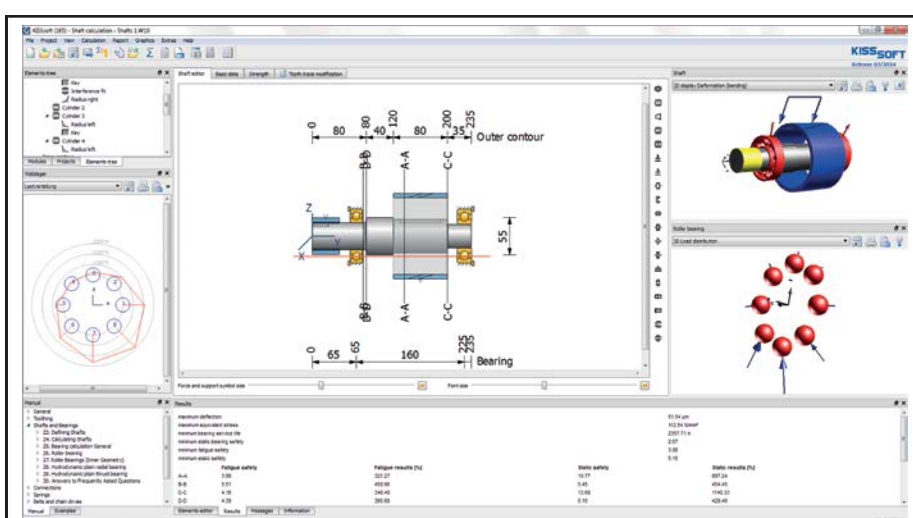

Figure 4 **KISSsoft shaft calculation (2014).**

dimensional models — most critically in bevel gears with spiral teeth. As large bevel gears were in short supply, there was a demand to be able to mill bevel gears directly on five-axis machines. The 3-D models needed to achieve this would then be supplied by the gear calculation programs.

The main problem here was the system performance this would require: i.e., to create realistic gear models you would need not only sufficient computing power, but also the amount of memory required by a mid-range, 3-D CAD system.

Calculating Variants

Since its very earliest days, the mathematical sizing of parts has faced a particular problem. Although the design phase may involve numerous design variants, which are handled in parallel, almost as separate strands, the designer will only select one variant in the end. Nowadays, it is also fairly usual for several variants of a finished design to be processed at the same time, either as standard gear series or as special designs that can be customized to meet specific customer requirements.

The first calculation programs provided absolutely no support for variants. At that time, the limitations on computing power meant that using even one variant in a calculation was quite an achievement. As the hardware has become more powerful, the number of options available in this area has also increased. In particular, the ability to systematically vary parameters to find the best possible solution plays a significant role in calculation programs.

KISSsoft has now implemented variant options at every level of its calculations, starting from individual parameters, such as those used to calculate the profile shift for gears, or find the tolerance pairs for interference fits, through the systematic variation of several parameters at the same time, as required for fine sizing the macro and micro geometry of gears, up to varying the parts in a transmission at system level. It is these additional functionalities that make a calculation program into a modern, efficient system.

Apps

Nowadays, everyone involved in IT is talking about apps. These are relatively small programs which are primarily designed to run on tablets and smart phones. A range of apps, most of which are available free of charge, have already been created for performing mechanical engineering calculations. Each of these apps can handle a small-scale calculation, e.g., determine Hertzian pressure or the L_{10} service life of a roller bearing. However, it is unlikely that the current methods of approach can be used to create apps for more complex problems. This is mainly due to the fact that users are unwilling to input large volumes of data on a 4" screen with a virtual keyboard. And a multilevel calculation would simply demand too much from these devices.

Less of a problem is posed by the relatively weak hardware used by these handheld devices, because most machine element calculations do not require a lot of power. These devices could also be used to find solutions with approaches designed to run the front-end on a handheld device and the back-end on a server over the Internet.

FEM vs. Machine Elements

By the 1980s, computers had become powerful enough to run the FEM calculations described above on a standard designer's workstation. People immediately began forecasting the preeminence of FEM programs, claiming that all the problems encountered in mechanical engineering would now be resolved once and for all. However, in the meantime, this euphoria has been tempered by the sobering realization that, although computers are becoming ever more powerful and sophisticated, the same cannot always be said for their users.

FE models still have to be created by specialists who are capable of defining suitable constraints and interpreting the results. Creating useful and accurate documentation also requires time and effort. But that is not to say that the FEM methods are generally superfluous. They are ideal for handling complicated structures or even only a cylinder head cover. It is just that standardized calculation methods are easier to use for simpler machine elements and provide more meaningful results. In the meantime, the basic calculation methods in the standards have been used to create more sophisticated enhancements. These can now be used to process reliable cases, which previously required a part to be investigated using FEM. These include, for example, the method according to Obsieger (Ref.3) for the detailed investigation of tooth root stresses in a gear. This method has proven to be so reliable that it is even used in aero-engineering.

There is now even a certain trend towards combining FEM with the classic definition of machine elements as the next step in the evolution of calculation programs. This is still based on the idea of using just a few parameters to describe a machine element. The program then creates an FE model in the background to determine the stresses in the part. These stresses are then used as the basis on which strength is verified as specified in the standard. Sheet 2 of VDI 2737 for the proof of bolted joints is one of the first published methods that specifically applies this approach. However, the alluring simplicity of this approach conceals such a multitude of traps in its implementation that all previous advances made in this direction are reduced to side issues.

Transition to a System

Classic machine element calculations only involve a single machine element; it is hard to keep track of much more data at once. Nowadays, computers are powerful enough to process an entire system of machine elements, at least "almost" simultaneously. The principles involved here mean that machine elements which depend on each other can only be processed successively, because the results for one element may affect the next. If a ring is closed (A depends on B, B depends on C, and C depends on A), iteration is the only approach. But this is just what computers do best: they can process huge volumes of calculations at such a speed that you have the feeling they could do everything simultaneously. In our opinion, this is why system programs such as *KISSsys* are the way forward. Once a system has been defined, the engineer can optimize data at a particular point and also constantly keep an overview of the entire system.

CAD Integration

CAD integration has been an ongoing topic of debate ever since the first calculation programs were designed. The first thing many interested parties want to know is whether a calculation program is compatible with their CAD system. Unfortunately, there is not a great deal of data that can sensibly be changed between CAD programs and calculation programs. The difference between machine element calculation and FEM is very clear: the former runs using only a small number of parameters whereas the latter requires the exact geometry. And because the parameters required for a machine element calculation are only rarely directly available in a CAD program, it is usually quicker and easier to input them manually rather than use an interface.

The limited number of options available is primarily due to the way CAD programs are currently structured. Efficient integration will only be possible if all the values involved in the calculation (e.g., speed, torque, and also material data) can also be managed and transferred to the (in-house or external) calculation program.

Full integration also means that designers who use the CAD system must be extremely disciplined and work exclusively with parameters. To explain this, a shaft calculation is used here as an example: The shaft contour is transferred to the calculation program, which identifies that the shaft diameter is not sufficiently dimensioned. When the modified diameter is returned to the CAD program, it immediately causes a number of problems because the changed diameter affects the design. For example, a bearing with a larger internal diameter must now be used. If the design has not been thoroughly parameterized, this type of change will require a lot of data to be input manually.

Summary and Outlook

In summary, our evaluation is that the greatest challenges we face in the coming years involve the implementation of efficient variant calculation processes for optimizing data at every level, up to system level, at the design phase, and the integration of machine elements and FEM.

References

1. Niemann, G. *Maschinenelemente*, Vols I-III, 1981, Springer Verlag.
2. Roloff, Matek. *Maschinenelemente*, Vieweg Verlag.
3. Obsieger, B. "Zahnformfaktoren von Aussen- und Innenverzahnungen," *Konstruktion* 32 (1980), pp. 443-447.

For more information:
KISSsoft AG
Phone: +(41) 55254 2050
www.kisssoft.ag