

# CFD So Easy Your Teenager Could Do It...NOT!

Have you seen those advertisements on TV for a major office supply chain where a person is faced with a really complex dilemma? Then someone presses the big red EASY button and the problem is instantly solved. Some CFD software providers want you to think doing CFD is about that easy. But what does it really mean for CFD to be easy?



**W**e all love something that is easy to use. In fact, if something is easy to use, we consider it to be a great design; that is, providing it also does what we want it to do. But if it doesn't do what we want it to do, it really doesn't matter how easy it is to use (or how little we paid for it), we won't be satisfied. Many times, when something is made so easy to use that anyone can use it, functionality and flexibility are compromised. If you want to do something outside the standard mode of operation you might be stuck.

If we're resourceful, we may try to make an "easy to use" product work the best it can (since we already purchased it) by adopting some round-about techniques to help us overcome its shortcomings. But this can quickly get complicated and pretty soon the overall process is no longer as easy to do as we initially intended.

---

*There is definitely a difference between "easy to use" and "easy to do".*

---

We can apply this to CFD software. These days, more and more CFD software products are

advertising how easy they are to use. In some sense, they imply that your teenager could do CFD using their software...it's that easy. But, as most of us know, CFD is not a simple subject. While you can usually simplify a complex subject, simplifications almost always lead to some compromise in functionality and/or flexibility. In the ideal case those simplifications won't compromise the functionality you need. If they do, you won't be satisfied. But another important thing to remember is that those simplifications that make the CFD software easy for you to use today may limit your capabilities down the road when your needs expand.

We were working on a CFD project that required modeling a very complicated geometry. Unfortunately, we weren't fully aware of how big a challenge we faced until we were knee deep into the project. The CFD software we leased was advertised as being one of the easiest to use. They had simplified many of the complex decision making processes usually required when doing CFD modeling to a few different choices with intuitive icons and buttons (the big "Easy" button). This was important to us because we planned to transfer the modeling work to our supply base that tended to have a lower level of engineering expertise. The CFD software was easy to use. When the software company demonstrated a very simple simulation of our

## True Confessions of a CFD Software User

problem to us, they showed us how easy it was to use their software. We were impressed and thought we'd be done with this project in no time.

But with the software's simplicity of use came many compromises that we didn't fully realize until we were trying to model the real situation. The first thing we found was that our "easy to use" CFD software had very limited geometry input capability. Our geometry was defined in Nastran format. But our CFD code preferred a CAD input format which was difficult and time consuming for us to acquire. The software advertised being able to read Nastran input formats, but the translator wasn't robust. It usually crashed trying to read our complex geometry. If it didn't crash, the model was read into the software as a single huge entity. We lost the ability to retain and manipulate groups of surfaces which would have enabled us to work with smaller pieces of the large model. As an alternative we tried different round-about methods to convert our geometry data into something our CFD code could read. But our lives got more complicated since we had to acquire and learn to use another software package.

Later we found out that even if we had been able to read our complex model into the CFD software, we would have encountered another show stopper. The complexity of our model resulted in many surface elements that were far from ideal. We had overlapping surfaces, cracks, and intersecting surfaces in our surface mesh. These can be pretty normal occurrences for highly complex geometries. So the next step was to correct these problems. No CFD code can generate a high quality volume mesh for the CFD analysis when these deficiencies in the surface mesh are

present. Our "easy to use" CFD software did not contain any tools for repairing meshes. Most CFD software relegates the task of repairing surface meshes to third party software products. They assume that when you bring the surface mesh into their software, it will be a perfect mesh. So again, we would have to acquire and learn another piece of software.

---

*It was more important for us to use CFD software that enabled us to "do" what we had to do than to be simply "easy to use".*

---

This wasn't the last of the compromises we would have had to make if we stayed with the "easy to use" CFD software. In addition to the complex geometries, our model also had a number of jet flows. Jet flows have large gradients which make it important to be able to locally increase the resolution of the mesh in the regions where the jet expands. Our "easy to use" CFD software did not have adaptive meshing capability. But, more fundamentally, its simplified meshing schemes did not give us precise manual control of the volume mesh.

In the end we switched CFD codes. We selected software that wasn't as easy to use, but offered a great deal more functionality and flexibility that ultimately enabled us to do what we needed to do. It was more important for us to use CFD software that enabled us to "do" what we had to do than to be simply "easy to use".