[redacted]

N95- 28727

1995122306

GRID GENERATION - A VIEW FROM THE TRENCHES

David Ives, Robert Miller, William Siddons, Kevin VanDyke[1]
Pratt & Whitney
East Hartford, CT 06118

ABSTRACT

This paper presents "A view from the trenches" on CFD grid generation from a Pratt & Whitney perspective. We anticipate that other organizations have similar views. We focus on customer expectations and the consequent requirements. We enunciate a vision for grid generation, discuss issues that developers must recognize.

BACKGROUND

The Engine Industry. Our grid generation goals are best understood in the context of the turbine engine industry, which is in the process of re-engineering with the premise that turbomachinery is a commodity in a mature business. This implies that specialists are used only in the core technologies, otherwise generalists will be supplemented by outsourcing of labor and/or purchase of vendor codes to meet business needs. The new paradigm of continual process improvement emphasizes driving costs down and reducing cycle time. The user of a CFD system is not a PhD level CFD specialist but rather a generalist designer. This represents a fundamental change from the last decade. The tight coupling of reduced design cycle time with time-to-market and bottom line profitability in this responsive business environment sets new CFD process requirements.

The Designer

A typical designer must balance theory, experiment, experience, and calculations covering a wide span of disciplines including CAD, aerodynamics, heat transfer, structures, and manufacturing. A designer is usually not a CFD expert. There is simply not time to develop in-depth skills in a specific area because of the wide range of skills and knowledge needed for design and because of the requirements to meet design deadlines.The designer may typically have a four month hiatus between sets of analyses, where he or she closely monitors component tests to achieve the necessary physical "feel" for the component. Thus the design systems must be easy to learn and relearn under tight schedules.

Flow Solver Success. Three dimensional viscous flow predictions have recently become validated in the engine industry as powerful design/analysis tools. Significant component improvements have been demonstrated for CFD-based designs. These recent successes are producing increasing pressure for expanding CFD capabilities to new component application areas. We have more opportunities now in CFD than we have ever had! Consequently, increasing numbers of component design systems are transitioning from correlation based algorithms to 3D viscous CFD analyses "tuned" with experimental test data. The grid generators must handle these new components.

Flow Solver Speed. A few years ago, the state of the art in industry was running jobs in queue on a supercomputer. Users often waited days for results for a single engine component. Then came a revolution: the use of networked workstations which currently produce overnight turnaround with high resolution. For example the flow analysis of a fan or compressor blade row, including details such as shrouds and tip clearances, is routinely done overnight. Network computing and more user friendly systems fueled a large expansion of the CFD user base to include the designers themselves, who

[1] Pratt & Whitney, West Palm Beach, FL

PRECEDING PAGE BLANK NOT FILMED

[redacted]

now run the overwhelming majority of CFD calculations at P&W. Flow calculations with one half million grid points are typical, with some cases employing 1.5 million grid points.

Grid Generation Turnaround. For simple components, such as a fan or compressor stage, specialized grid generators (developed over a ten year period) create the required grids without user intervention or even inspection. For complex new 3D components such as nacelle installations, complete combustor/diffusers, rocket turbopumps, etc., current grid generators may require days to months of elapsed time. In other words, the elapsed flow solver time can be a small fraction of the elapsed grid generator time!

Grid Generation Capability. Although current grid generators are rather general and applicable to a wide range of problems, some problems remain unresolved. For instance current grid generators currently cannot cope with some new applications such as the highly swept, highly wrapped (staggered), close-spaced periodic centrifugal inducer blade elements with multiple sets of embedded splitters as encountered in modern rocket turbopumps.

The Designer is the Customer. The designer ultimately decides which flow solver and grid generator are used, and which ones become "trashcan" CFD codes. Those that best meet the designer's needs get used. Those that do not best meet the designers' needs are discarded, even if such codes are "technically" superior. As a practical matter, the CFD grid generator developer must have a tight communication link with both the designer and the flow solver developer, so that the right needs are answered. The grid generator must integrate perfectly with the flow solver, and this requires constant communication as the target flow solvers and computing systems change regularly The key to this is understanding that the designer is the ultimate customer, and the flow solver/grid generator combination must meet design needs. This situation is like buying a house, where the advice is "location, location, location." Here the advice is "designer, designer, designer;" it is that simple. In practice, it is fully recognized that ,for logistical reasons, there may be "gatekeepers" between the developer or vendor and the designers. This does not alter the fact that the designer is the ultimate customer.

VISION

CFD Process Vision. The requirement for the CFD process in this new market-driven business environment is **one day** turnaround for the everyday aerodynamic design CFD cycle illustrated in the first slide. This cycle is composed of inspection of the preceding nights' run, conceptualization of modifications expected to improve performance, defining the new geometry using a CAD system, CAD-to-grid geometry transfer, grid generation, flow boundary condition specification, flow solution, and post-processing. The flow solver unattended running occupies 16 hours of this cycle. This new design may have a different topology than previous cases. The grid generation occupies **one hour** of this cycle, leaving at least four hours for the designer to conceive and define a new geometry to improve on the current design. This 24 hour design-analysis cycle is in tune with the human circadian rhythm, allows a large number of design cycles, and cuts time to market for new products. Each of the CFD system components must be sufficiently automated to allow the designer to operate them on a sporadic basis with minimal specialized training and support from CFD "experts". The second slide illustrates that others also recognize this one day turnaround requirement.

Grid Generator Requirements. The requirement is reliable **one hour** grid generation turnaround for one-time geometries when run by designers. The system must incorporate CAD to grid links which resolve tolerance issues, and produce grids with quality good enough for the flow solver. The designer must feel that the grid generation process is under control and predictable, since the grid generation process is now viewed as part of an overall CFD process which is expected to produce engineering answers in a day. The one hour grid turnaround requirement was derived by starting with the desired customer-defined goal and working backward.This requirements-driven approach will produce technologies significantly different (and more responsive to designer needs) than will a grid code written in isolation and then "thrown over the wall". We do not envision a new grid generator replacing our specialized grid generators (which usually run in at most a few minutes); in general the geometries we wish to analyze will be complex and significantly different from previous geometries. We also require that the grid generation be more automated with minimal user interaction to decrease variation between users, and to reduce training and retraining costs. Remember, every time the user must provide input, the variation in the grid increases! An automated approach for non-specialist customers is even more important for vendors than for internal developers, since local help and training may not be quite as close.

This **one hour** grid turnaround requirement is reality and must be fully accepted by CFD process providers. This is not an easy thing to do, given the current evolving state of grid generation. The development of grid generators to meet the one hour turnaround requirement may seem daunting taken out of context. Reliable one hour grid turnaround by generalist design users for up to a million cells is believed realistic based on hardware improvement projections and significant software process enablers discussed below. The one hour grid inside the one day CFD analysis will be the routine working environment of tomorrow!

Evolution or Revolution? Looking at how we have selected and deselected grid generation codes over the last decade at P&W, it becomes clear that we tend to hold on as long as possible to a technology that works, and only change codes or vendors when the current codes cannot effectively solve our problems in a timely manner. When we finally made the change to new codes, the technology always was a revolutionary jump, not an evolutionary improvement. For example, we switched from a command-line text driven grid generator to an interactive grid generator. As another example we ceased use of conformal mapping based generators when the coding changes required for flexibility became too time consuming because the basic code was not created with interaction in mind. When we switch codes, we typically evaluate the market leaders and "select" one within a single class of technology. Incremental improvements help postpone the date of transition to a new technology level, but do not change the outcome. We thus expect our next generation grid generators to be revolutionary, not evolutionary. Automation, which the dictionary defines essentially as "without human interaction," will increase and interaction will decrease.

The first generation of grid generators required typically thousands of command-line or script commands for complex cases. The second generation used graphical interaction to simplify the user interface, but still requires thousands of user decisions such as where to place a cursor and click. The interaction accelerated the user decisions and increased the probability of correctness, but did not significantly reduce the number of user decisions. Newer technology aims at reducing the number of decisions by automating tasks within the current process paradigm, such as the provision for more automated blocking or the ability to add components separately using templates. Simultaneously, altogether different gridding paradigms are emerging, such as the advancing front tetrahedral grid generators and the geometric grid generators discussed in Reference 3.

The interrelation between reliability and automation is illustrated in the fourth slide. A skilled user called on to make a moderate number of decisions will usually make the correct decisions, but as the number of user decisions grows, the probability of incorrect decisions grows. Given a **fixed** technology, the more user input "knobs" available, the higher the reliability if the user input is flawless. These trends interact to produce a reliability that at first increases with the number of user decisions, reflecting more user knobs to control special cases, but then decreases as the number of user decisions grows further due to user errors. The idea is to automate the processes to move the left side of this curve dramatically upward and to the left. This movement, labeled "progress", is where the current frontier lies. Specialized grid generators tuned for limited geometries tend to have high reliability and few decisions. This is typical of the specialized cascade grid generators we have developed over the last 20 years. They are not expected to be displaced by new technology since they already fully meet the design needs for their limited scope of application.

Revolution Enablers Recent hardware, software, communications, and flow solver improvements have dramatically altered the computing and coding environment, making the one hour grid turnaround requirement a realistic goal.

Significant hardware advances have occurred in cost, speed, core storage, disk storage, and graphics. Workstations have lowered the cost per computational unit of work by order(s) of magnitude. Development has been accelerated by computing environments where there is no (or only a negligible) chargeback based on use. Developers can afford to experiment with new methods which show promise without having to justify the computing expenses. The impressive speed of modern workstations exceeds that of a single IBM 3090 processor of just 4 years ago. Just ten years ago, it was difficult to get authorization for 2 megabytes of core on a mainframe, now workstations typically have 32 times this amount. With this much core, one can try out new ideas quickly by not being concerned (in the 1st cut) with storage. The disk storage of typically 10 "cylinders" of ten years ago is now replaced by disk storage typically 100 to 200 times as much. This allows room for better documentation and stored verification cases, improving the code quality. The field of 3D graphics is about to explode. While there have been 3D graphics workstations available, the workstation market sales leader has announced that their next CPU chip will contain three subsystems: an integer unit, a floating point unit, and a graphics unit. You evidently will not be able to buy one without powerful 3D graphics capability! Others are developing increasingly inexpensive 3D graphics cards. A personal computer company has announced a 3D version of their "Quick Draw" operating system code which can display 3,000 facets in real time without an accelerator card. Since grids are 3D relationships in space, graphical visualization is an absolute necessity for the development of grid codes. This holds true even when automated grid codes do not need graphics for running; they still need the graphics for code debugging.

Significant software and communications advances include overlapping windows, the "make" facility, code checkers, sorting and searching algorithms, publications, and the Internet. The overlapping windows and concurrent processes capabilities allow development work to proceed on several different fronts at once. The UNIX operating system and the "make" facility simplify the construction of large codes. Typically a flow solver of ten years ago would range from 2,000 to 8,000 lines. Todays' flow solvers and grid generators often range from 50,000 to 150,000 lines of code. Progress on the code quality front has also proceeded on a rapid track. The **ftnchek** FORTRAN code checker of Reference 4 is capable of finding errors (such as uninitialized variables) in code even six levels deep in subroutines. The **lint** checker serves similar functions for "C" code. Computer science has contributed by establishing an "object attitude" relative to modular coding which again has allowed large, complex projects to be successfully completed. The knowledge of rapid sorting and searching algorithms has been developed and published extensively, and much faster codes can result.There are now a number of popular magazines and books devoted to coding techniques, while there were only a handful ten years ago. The Internet can be viewed as the world's biggest, best library as well as a source of technical news, technical papers, and assistance. It also simplifies collaboration between developers on a project.The power of this resource is growing at an exponential rate.These hardware and software advances work together to bootstrap the rate of code development; and this process will continue. We know of developers today who produce at a rate 2 to 8 times higher than ten years ago.

Another enabler is the relative insensitivity of modern finite volume flow solvers to grid element quality (distortion) compared to the finite difference flow solvers of the past.The law of diminishing returns implies that once a grid is "good enough" for the flow solver, further improvements are hard to come by and might not be economically or time justified. This presumes that predictive grid quality measures for the flow solver are known, so that "good enough" is a measurable quantity. The simple grid quality measure "flatness" described in Reference 3 has worked well for our Euler flow solver. By relaxing the grid requirements from "perfect" to "good enough", new grid generation paradigms become feasible. Of course the grid must still be fine enough to resolve significant features so that grid independence is achieved.

USER NEEDS

Needs. We asked developers, designers, and other CFD users at P&W to identify their grid generation needs, and were inundated. The results were merged, categorized, and are presented in the attached slides. They are a "wish" list based on our current experiences, and do not include all possible requirements.

Implications. An examination of the needs listed in the slides leads one to conclude that there are certainly a lot! This implies that the grid generator codes of the future will be large. These large codes will have more links between sections, greatly increasing the importance of coding quality standards and error avoidance procedures. Vendors may have an advantage with large codes since they can spread their costs over many customers, if there are only a few vendors. Another possible scenario is cooperative work between developers in different organizations to each address different needs, with coordination over the Internet. The National Grid Project and the NASA IGES effort are examples of such cooperation. An interesting topic for discussion is whether one dominant vendor will emerge, as in personal computer operating systems, or whether cooperative efforts of smaller groups will be the norm.

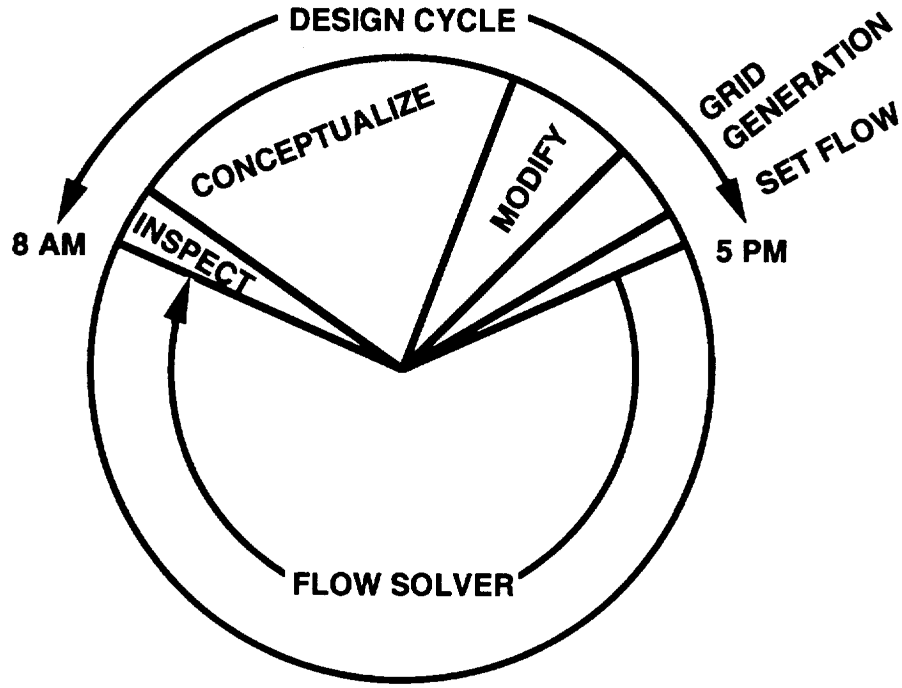
CONCLUSIONS

This presentation concentrated on grid generation requirements from a customer-driven viewpoint. We first identified the designer as the customer, then discussed design requirements and presented perceived designer needs. We assert that the grid generation requirement is **one hour turnaround** for unique complex geometries on a workstation by a designer.We believe that this is an achievable goal, given significant enabling hardware and software developments. In closing, the bottom line is to recognize that the customer is the designer, and to be customer-driven.

REFERENCES

1. Boerstoel, J.W., Spekreijse, S.P., and Vitagliano, P.L., "The Design of a System of Codes for Industrial Calculations of Flows around Aircraft and Other Complex Aircraft Configurations," 10th AIAA Applied Aerodynamics Conference, Paper # AIAA-92-2619-CP, Palo Alto, CA, June 1992.
2. Rubbert, P., "CFD and the Changing World of Airplane Design," AIAA Wright Brothers Lecture, Anaheim CA, Sept 1994.
3. Ives, D. C., "Geometric Grid Generation," Proceedings of Workshop on Surface Modeling, Grid Generation, and Related Issues in CFD Solutions," NASA CP#3291, NASA Lewis Research Center, May, 1995.
4. Monoit, R, "**ftnchek**," available by anonymous ftp from research.att.com (192.20.225.2) in the directory /netlib/fortran or by electronic mail by sending the message "send ftnchek from fortran" to the internet address: netlib@oml.gov

One Day Turnaround CFD



One Day Turnaround CFD

we are not alone

"In aerodynamic design work for commercial aircraft, CFD calculation results for aerodynamic analysis should be produced ... with short turnaround times (for large, complex calculations, 1 day, say)"

Boerstoel, J.W., Spekreijse, S.P., and Vitagliano, P.L., "The Design of a System of Codes for Industrial Calculations of Flows around Aircraft and Other Complex Aircraft Configurations," 10th AIAA Applied Aerodynamics Conference, Paper # AIAA-92-2619-CP, Palo Alto, CA, June 1992.

"The significance of this is that the total elapsed time for redesigning and arriving at a wing which meets all aerodynamic and manufacturability requirements can now be envisioned to approach the order of one day!"

Rubbert, P., "CFD and the Changing World of Airplane Design," AIAA Wright Brothers Lecture, Anaheim CA, Sept 1994

One Hour Grid Generation

The requirement is for reliable one hour grid generation turnaround for one-time geometries when run by designers. The system must include CAD-to-grid links which resolve tolerance issues and produce grids with a quality good enough for the flow solver. The designer has to feel that the grid generation process is under control and is predictable.

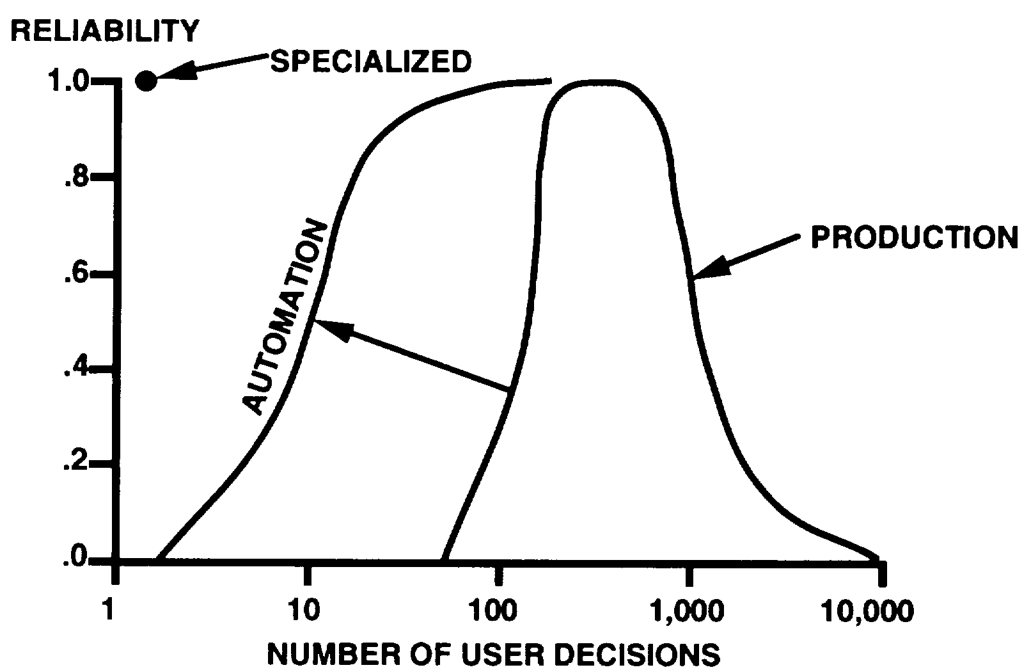
The Designer is the Customer

CFD only counts if it gets into the product

RECOGNIZE WHO THE CUSTOMER IS
RESPOND TO ACTUAL CUSTOMER NEEDS
FOR GRIDS, THE FLOW SOLVER IS ALSO A CUSTOMER
THIS REQUIRES LOTS OF COMMUNICATIONS
NO "OVER THE WALL" APPROACHES

DECISIONS

A MEASURE OF AUTOMATION



Enablers

Workstations
  Speed/Cost
  Large core
  Large disk
  Overlapping windows
Make
Code checkers
Object attitude
How-to publications
Internet
Fast 3D surface
Graphics
  In CPU:
    integer
    floating
    graphics
Robust Flow Solvers

Process Reliability & Stability Needs

sufficient user control to ensure process nearly always works
soft landing after error with meaningful messages and waypoint capability
stability and reliability is usually more important than added features
check all user input for range & reasonableness, allow re-entry of data
reduce user input: every time you require user input, you introduce variation
be predictable enough for just-in-time grid generation
handle often imperfect CAD data with gaps, overlaps, tolerances
handle scales differing by 5 orders of magnitude
prevent bus errors and segmentation (memory) errors

Ease of Use Needs

simplicity, lots of feedback on progress, active on-screen help
automate everything you can , then use graphical interaction for the rest
reduce need for grid generation knowledge & expertise
clear and consistent menus, messages, code appearance, process directions, and documentation
if something fails, suggest alternatives or corrective action on screen
provide visualization to track progress and detect failures
provide automated or simplified block decomposition
support general, non-restrictive script and data format with in-line comments
scripting and journal file for:
- batch running
- template construction by experts for "push-button" specialized uses
- coupling with other automated processes
- simplified re-learning and work organization for sporadic users
- restart from partially executed process waypoints
- reduced variation

Integration Needs

block face points which are conceptually the same must be IDENTICAL
input and output source code to simplify integration into existing systems
read existing user formats, ie: VSAERO
batch running

Standards Needs

write generic visualization files (ie: Wavefront .obj format)
accept standard CAD output formats (IGES, sterolithography, ...)
write Postscript files for design documentation
provide hardware platform independence where possible
all platforms should run identically
support the hardware platforms which designers have on their desks
use terminology and notation familiar to designers, where possible

Developer Communication Needs

rapid response to critical bug reports, tell user if fix is not planned
good code/version control process and results tagging
Internet
- new features
- hints, suggestions
- feedback
- data base of outstanding and resolved bugs

Miscellaneous Needs

common grid generator with international divisions and industrial partners
unstructured output capability, even for structured grids
adaptive tie-in to solver

Implications

Large quantity of needs implies large codes needed

Large codes imply coding quality much more important

Will a dominant vendor emerge?
- ability to spread development costs
- like personal computer operating systems

Will cooperative efforts prevail?
- National Grid Project
- NASA IGES

NASA OVERVIEWS