On the Use of Parametric-CAD Systems and Cartesian Methods for Aerodynamic Design

Marian Nemec[1], Michael J. Aftosmis[2], and Thomas H. Pulliam[3]

[1] NASA Ames Research Center, Moffett Field, CA, USA *nemec@nas.nasa.gov*
[2] *aftosmis@nas.nasa.gov*
[3] *tpulliam@mail.arc.nasa.gov*

1 Introduction

Automated, high-fidelity tools for aerodynamic design face critical issues in attempting to optimize real-life geometry and in permitting radical design changes. Success in these areas promises not only significantly shorter design-cycle times, but also superior and unconventional designs. To address these issues, we investigate the use of a parametric-CAD system in conjunction with an embedded-boundary Cartesian method. Our goal is to combine the modeling capabilities of feature-based CAD with the robustness and flexibility of component-based Cartesian volume-mesh generation for complex geometry problems. We present the development of an automated optimization framework with a focus on the deployment of such a CAD-based design approach in a heterogeneous parallel computing environment.

2 Problem Formulation

The aerodynamic optimization problem consists of determining values of design variables such that a given objective function is minimized, while satisfying the governing flow equations and any other side constraints. Examples of objectives and constraints include performance functionals, such as lift and drag, and geometric quantities, such as volumes and thicknesses. The flow equations are the three-dimensional Euler equations of a perfect gas.

A modular framework is constructed to solve the optimization problem based on two optimizers: 1) a genetic algorithm (GA), and 2) a BFGS quasi-Newton algorithm, where the objective function gradient is evaluated via central-differences. At the core of the framework is the analysis module that consists of a CAD-system interface and the *Cart3D* flow-analysis package. The primary components of *Cart3D* include a Cartesian grid generator [1], and a flow solver [2]. Below, we provide a description of the CAD interface, followed by the optimization framework. See [3] for additional details.

3 CAD Interface: CAPRI

The CAD interface controls the regeneration of a CAD model in response to a design change and provides a corresponding watertight surface triangulation. This is accomplished by the use of the Computational Analysis PRogramming Interface (CAPRI) [4, 5]. CAPRI exposes the master-model feature tree of the CAD model and allows direct modification of parameters within that tree. Most design variables are associated directly with values exposed in the feature tree. Depending on the CAD-face topology of the model, CAPRI triangulates the faces by using either a quality triangulation algorithm [6, 4], or by decomposing quadrilaterals generated with an auto-blocking algorithm [7].

4 Optimization Framework

The synthesis of CAPRI with the optimization process is shown in Fig. 1. At each iteration of the optimization, CAD geometry requests are generated for different design-variable values and these are placed in a central repository. Independent of the optimization runs, a distributed geometry server is initiated that consists of multiple CAD nodes. The nodes process the geometry requests by retrieving the required parts and assemblies, regenerating the CAD models, and providing surface triangulations for the optimization processes. Since the geometry requests are independent, the geometry server achieves nearly linear scalability.

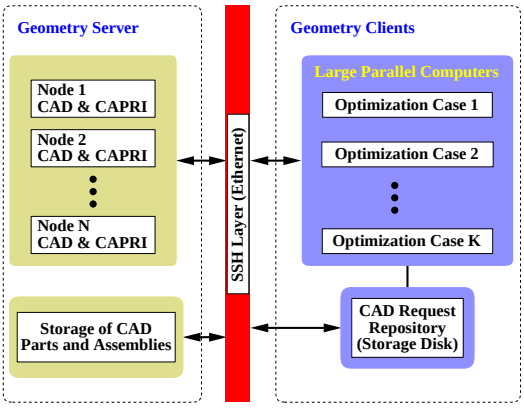


Fig. 1. Layout of the interface between optimization processes and geometry server

The number of CAD nodes is limited by the number of available CAD licenses, as each node consumes one license. An immediate concern is that the CAD nodes may become the bottleneck of the optimization process, idling the processors of the compute engines. To avoid such bottlenecks, we dynamically allocate the processors of the optimization to the number of completed surface triangulations. Figure 2 illustrates this on an example with 64 processors. At the start of each design iteration, all processors are dedicated to the solution of the first returned surface triangulation from the CAD nodes. This is the base state of the gradient method and the first chromosome of the GA, denoted as

"Geometry 1" in Fig. 2. Upon completion of the first geometry analysis, we check the number of completed surface triangulations. These are processed by the CAD nodes while the analysis of the first geometry is performed on the compute engine, denoted as "Geometries $2 \ldots K$" in Fig. 2. The number of processors is distributed among the completed surface triangulations and multiple analysis modules are executed on subsets of the available processors. This dynamic, coarse-grained parallelism provides not only concurrent execution of serial tasks, but also ensures high parallel efficiency of the flow solver by limiting the number of processors available to each analysis module.

The worst case scenario occurs when the wall-clock time required for the processing of a geometry request exceeds the time for completion of the flow solution. If only one CAD node is available, then this CAD node would not be able to feed the compute engine with geometries without processor idle time. This situation is unlikely, since CAD model regeneration and tessellation tasks have computational complexity of $\mathcal{O}(N^2)$, while volume mesh generation and flow solution tasks are $\mathcal{O}(N^3)$.

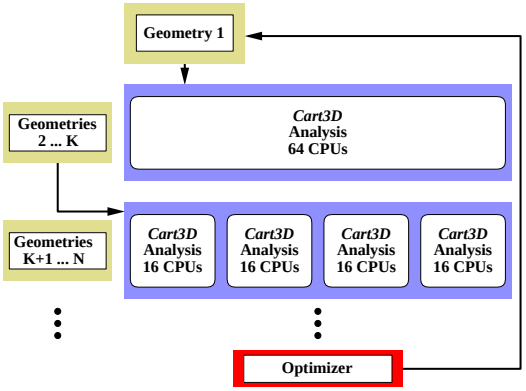


Fig. 2. Dynamic allocation of processors to mask the latency of CAD geometry processing

5 Design Example

We investigate the performance of the optimization framework for a design example based on the configuration shown in Fig. 3. This generic model is a CAD assembly of five parts, where the wing and canard are "attached" to the fuselage via two parameters, their horizontal and vertical locations, respectively. These parameters are constrained to intersect the projection of the fuselage on the symmetry plane within the CAD system. This simple construct avoids non-physical configurations, for example wings that detach from the fuselage during the

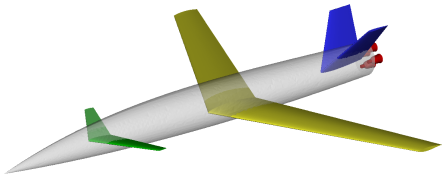


Fig. 3. CAD model configuration (before component intersection)

optimization, even if the fuselage shape and dimensions change. The CAD model is constructed using the Pro/ENGINEER CAD system.

Before presenting optimization results, we characterize the performance of the optimization framework. Table 1 presents average CPU timing results for the CAD model regeneration and surface triangulation using CAPRI. It is clear from Table 1 that CAD-model regeneration times are not a significant expense even for problems with many design variables. While the time required for surface triangulation is not prohibitive, it is important to avoid all unnecessary re-triangulations during the optimization. This is accomplished by caching an associated baseline triangulation for each part prior to the optimization.

Table 1. Average CPU time for CAD-model regeneration and tessellation (600 MHz R14000 SGI Octane Workstation)

Part	CAD-Model Regeneration (s)	Tessellation (s)	Number of Triangles	Tessellation Algorithm
Fuselage	2.0[a]	93.3	$\approx 41,000$	Quality-based
Wing	3.0[b]	16.5	$\approx 50,000$	Right-triangle

[a] No shape-section change, only global parameter modifications
[b] Shape-section change and planform parameter modifications

Table 2 presents average timing results for individual components within the analysis module. The volume mesh contains roughly 1.5 million cells for a half-span model of the configuration. The time for the mesh-solution transfer algorithm used to "warm-start" finite-difference gradient computations is also shown. Comparison of Tables 1 and 2 indicates that the time required to

Table 2. Wallclock times for individual components of the analysis module (600 MHz R14000 SGI Origin 3000)

Component	Time (s)	Algorithm
Mesh Generation[a]	132.0	Serial
Flow Solution (flowCart 1.2)	455.0	Parallel (64 processors)
Mesh Solution Transfer	26.0	Serial

[a] Also includes component intersection, domain decomposition, and multigrid coarse-mesh generation

complete a CAD-model regeneration and surface triangulation is a factor of six smaller than the time required for a flow solution. This means that by the time the analysis module completes the flow solution of the first chromosome of the

GA or the base-state of the gradient method, six new surface triangulations are ready for analysis, ensuring good CPU efficiency.

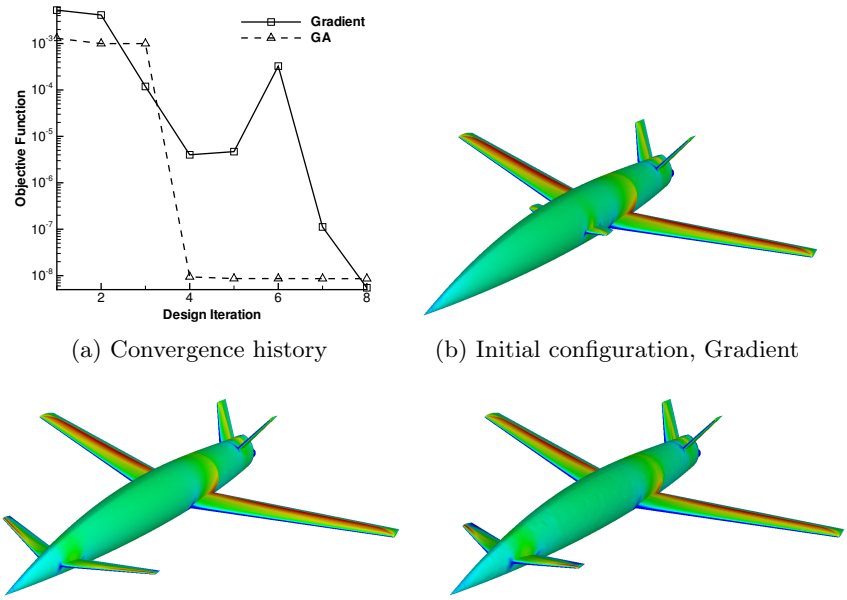
We consider the optimization problem of attaining a nearly zero pitching moment coefficient by optimizing the canard/tail control surface. The design variables are the control surface aspect ratio (AR), twist (tip relative to root section), and axial position along the center line of the fuselage. The problem has two local optima, the tail or canard configuration, with the canard configuration as the global optimum due to an aft location of the center of gravity. For optimization using the GA, the canard area is also a design variable. This introduces the possibility of a topology change in the design space. We use 16 chromosomes for each generation of the GA. For the quasi-Newton algorithm, the control surface area is kept constant (60.0). The freestream Mach number is 0.85, and the angle of attack is 1.0 deg.

The objective function uses a quadratic-penalty formulation, with a target lift coefficient of 0.222 and a target pitching moment coefficient of 0.001. The initial pitching moment is -0.0714. Figure 4(a) shows the convergence history of the objective function. The label "Design Iteration" refers to the number of generations evaluated by the GA, and the number of objective function and gradient evaluations by the quasi-Newton algorithm. The GA converges within 5 design iterations, requiring only 80 function evaluations. The quasi-Newton algorithm requires 56 function evaluations to trim the configuration.

Figures 4(b) and 4(c) show the initial and final designs for the quasi-Newton algorithm. The control surface converged to the minimum allowable forward location on the fuselage (8% of fuselage length), a twist angle of 2.98 deg., and AR of 6.03. Figure 4(d) shows the final design using the GA. For this case, the optimization converged to the upper bound of the control surface area, which is 60.0, a forward location of 8.2% of fuselage length, a twist angle of 3.41 deg., and AR of 4.36. The small differences in the two designs indicate that there may be many control surfaces that trim this configuration and further constraints are required to define a unique problem.

6 Conclusions

A modular framework for the aerodynamic optimization of complex geometries has been developed. The framework includes a direct interface to a parametric-CAD system that allowed an efficient manipulation and surface tessellation of generic-CAD models. Furthermore, the use of a component-based Cartesian method reduced the demands on the CAD system by reusing cached component triangulations, and improved the robustness of the framework due to the decoupling of the surface mesh form the volume mesh. Parallel efficiency of the framework was maintained even when subject to limited CAD resources by dynamically re-allocating the processors of the flow solver, thereby using the scalability of the solver to mask the latency of the geometry server.



(a) Convergence history (b) Initial configuration, Gradient

(c) Final configuration, Gradient (d) Final configuration, GA

Fig. 4. Design example summary (Contour plots denote surface Mach number)

7 Acknowledgments

The authors gratefully acknowledge Robert Haimes (MIT). This work was performed while the first author held an NRC Research Associateship Award.

References

1. Aftosmis, M. J., Berger, M. J., and Melton, J. E., "Robust and Efficient Cartesian Mesh Generation for Componenet-Based Geometry," *AIAA J.*, Vol. 36-6, 1998.
2. Aftosmis, M. J., Berger, M. J., and Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," AIAA Paper 2000–0808, Reno, NV, Jan. 2000.
3. Nemec, M., Aftosmis, M. J., and Pulliam, T. H., "CAD-Based Aerodynamic Design of Complex Configurations Using a Cartesian Method," AIAA Paper 2004–0113, Reno, NV, Jan. 2004.
4. Haimes, R. and Aftosmis, M. J., "On Generating High Quality "Water-tight" Triangulations Directly from CAD," ISGG, June 2002.
5. Haimes, R. and Crawford, C., "Unified Geometry Access for Analysis and Design," 12th International Meshing Roundtable, Sept. 2003.
6. Aftosmis, M. J., "On the Use of CAD-Native Predicates and Geometry in Surface Meshing," NASA TM 1999–208782, Aug. 1999.
7. Haimes, R. and Aftosmis, M. J., "Watertight Anisotropic Surface Meshing Using Quadrilateral Patches," 13th International Meshing Roundtable, Sept. 2004.