



AIAA 2002-0752
CFD General Notation System (CGNS):
Status and Future Directions

Steve M. Legensky and David E. Edwards
Intelligent Light, Lyndhurst, NJ

Robert H. Bush
Pratt & Whitney Aircraft, East Hartford, CT

Diane M. A. Poirier
ICEM CFD Engineering, Berkeley, CA

Christopher L. Rumsey
NASA Langley Research Center, Hampton, VA

Raymond R. Cosner
Boeing Phantom Works, St. Louis, MO

Charles E. Towne
NASA Glenn Research Center, Cleveland, OH



40th AIAA Aerospace Sciences Meeting and Exhibit
14-17 January 2002
Reno, NV

For permission to copy or to republish, contact the copyright owner named on the first page.

CFD General Notation System (CGNS): Status and Future Directions

Steve M. Legensky^I and David E. Edwards^{II}
Intelligent Light, Lyndhurst, NJ

Robert H. Bush^{III}
Pratt & Whitney Aircraft, East Hartford, CT

Diane M. A. Poirier^{IV}
ICEM CFD Engineering, Berkeley, CA

Christopher L. Rumsey^V
NASA Langley Research Center, Hampton, VA

Raymond R. Cosner^{VI}
Boeing Phantom Works, St. Louis, MO

Charles E. Towne^{VII}
NASA Glenn Research Center, Cleveland, OH

A limitation in today's Computational Fluid Dynamics (CFD) simulation systems is the lack of a standardized, robust, efficient data format to be used in the exchange of information between the various stages of a numerical simulation (grid generation, solver and post processing) using either in-house or commercial simulation tools. A committee consisting of participants from NASA, the aerospace industry and commercial software companies was formed to address this problem. The result of that effort is the development of the CFD General Notation System (CGNS) that was first released in May 1998. New functionality is currently being added to the CGNS system. The Standard Interface Data Structures (SIDS) and corresponding Application Programming Interface (API) are being extended to support chemistry and species, arbitrary user data and linked CGNS files. Version control has been added to allow CGNS source development by multiple organizations. An effort is underway for CGNS to be adopted as an ISO standard for the recording of aerodynamic data. This paper describes the current progress in CGNS since January 2000, how it is being utilized today and the future directions for CGNS.

1. Introduction

The CGNS (CFD General Notation System) project originated in 1994 through a series of meetings that addressed improved transfer of NASA technology to industry. A principal impediment in this process was the disparity in I/O formats employed by various flow codes, grid generators, and other utilities, and CGNS was conceived as a means to promote "plug-and-play" CFD. Agreement was reached to develop CGNS at

Boeing, under NASA Contract NAS1-20267, with active participation by a team of CFD researchers from NASA's Langley, Glenn, and Ames Research Centers, McDonnell Douglas Corporation (now part of Boeing), and Boeing Commercial Airplane Group. This team, which was joined by ICEM CFD Engineering Corporation of Berkeley, California in 1997, undertook the core of the development. However, in the spirit of creating a completely open and broadly accepted standard, all interested parties were encouraged to participate; the US Air Force and Arnold Engineering Development Center were notably present. In 1999, commercial software companies such as Fluent, CD ADAPCO, Intelligent Light, and NUMECA began to participate with the CGNS team. The CGNS Steering Committee is now a subcommittee of the AIAA Committee on Standards

From the beginning, the purpose was to develop a

Copyright © 2002 by Intelligent Light. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

^I Manager, Senior Member AIAA

^{II} Manager, Senior Member AIAA

^{III} Engineer, Associate Fellow AIAA

^{IV} Engineer, Member AIAA

^V Scientist, Associate Fellow AIAA

^{VI} Senior Technical Fellow, Associate Fellow AIAA

^{VII} Engineer, Member AIAA

system that could be distributed freely, including all documentation, software and source code. This goal has now been fully realized; further, as described below, control of CGNS has been completely transferred to a public forum known as the CGNS Steering Committee.

The specific purpose of CGNS was to provide a standard for recording and recovering computer data associated with the numerical solution of the equations of fluid dynamics. The intent was to facilitate the exchange of CFD data between sites, between applications codes, and across computing platforms, and to stabilize the archiving of CFD data. The format implemented by this standard was to be (1) general, (2) portable, (3) expandable, and (4) durable.

The resulting system today consists of a collection of conventions, and software implementing those conventions, for the storage and retrieval of CFD data. The system consists of two parts: (1) a standard format for recording the data, and (2) software that reads, writes, and modifies data in that format. The format is a conceptual entity established by the documentation; the software is a physical product supplied to enable developers to access and produce data recorded in that format.

The principal target is the data normally associated with compressible viscous flow (i.e., the Navier-Stokes equations), but the standard is also applicable to subclasses such as Euler and potential flows, and superclasses such as chemically reacting flows. The initial release addressed structured multi-block grids, flow fields, boundary conditions, and block-to-block connection information, as well as a number of auxiliary items, such as non-dimensionalization, reference states, and equation set specifications. This stage of development was summarized in reference 6.

It is worth noting that extensibility is a fundamental design characteristic of the system, which in principle could be used for other disciplines of computational field physics, such as acoustics or electromagnetics, given the willingness of the cognizant scientific community to define the conventions.

The standard format, or paper convention, part of CGNS consists of two fundamental pieces. The first, known as the Standard Interface Data Structures² (SIDS), describes in detail the intellectual content of the information to be stored. It defines, for example, the precise meaning of a "boundary condition". The second, known as the File Mapping⁴, defines the exact location in a CGNS file where the data is to be stored.

The implementation, or software, part of CGNS

likewise consists of two separate entities. CGNS files are read and written by a stand-alone database manager called ADF³ (Advanced Data Format). ADF manages a tree-like data structure, implemented as a binary file. Since the format of this file is completely controlled by ADF, and since ADF is written in ANSI C (FORTRAN wrappers are provided), these files and ADF itself are portable to any environment that supports ANSI C. ADF is available separately and constitutes a useful tool for the storage of large quantities of scientific data.

ADF, however, implements no knowledge of CFD or of the File Mapping. To simplify access to CGNS files, a second layer of software known as the Mid-Level Library⁵ is provided. This layer is in effect an API, or Application Programming Interface for CFD. The API incorporates knowledge of the CFD data structures, their meaning and their location in the file, enabling applications such as flow codes and grid generators to access the data in familiar terms. The API is therefore the piece of the CGNS system most visible to application developers. Like ADF, the API is written in ANSI C; all public API routines have FORTRAN counterparts.

Figure 1 illustrates an example of a typical CGNS file. It is given to provide the reader with an idea of the type of information that goes into a CGNS file and how it is organized. This example is from the structured-grid NASA Langley CFL3D⁹ code. It is for a one-zone grid and does not include grid connectivity information. Each ADF node is represented with its name and label (name/label). However, when the only difference between the name and label is $i_t\hat{i}$, the name has been omitted. The boundary conditions are specified under ZoneBC_t. Note that in this example, the user has also added the descriptor node under ZoneBC_t called CFL3DType. The information recorded under this node is not part of the CGNS standard, but rather was added for convenience in order to tie in the CGNS boundary condition identifier with a type specific to the CFL3D code. Descriptors like this may be sprinkled throughout the CGNS file. They provide an easy way to add descriptive comments to a file for the benefit of present or future readability. Flow solution information is stored in data arrays under the FlowSolution_t data structure. CFL3D gives this information at cell centers; thus, GridLocation_t specifies CellCenter in the example. The node Rind_t indicates the number of rind cells (ghost cells) at which flow solution information is also given. The FlowEquationSet_t node contains information about how the solution was run. The convergence history may be stored for each individual zone or globally under the CGNSBase_t node, as shown in this example. DataClass_t specifies the type of data given. Further information regarding the

reference quantities is given under ReferenceState_t. The nodes VelocitySound, ViscosityKinematic, Density and Length give the reference levels by which all quantities in the file have been non-dimensionalized.

```

ROOT node
+ CGNSLibraryVersion_t
+ Base/CGNSBase_t
  + Zone1/Zone_t
    + ZoneType_t
    + GridCoordinates_t
      + CoordinateX/DataArray_t
      + CoordinateY/DataArray_t
      + CoordinateZ/DataArray_t
    + ZoneBC_t
      + Ilo_Seg1/BC_t
      + PointRange/IndexRange_t
      + CFL3Dtype/Descriptor_t
      + ...(other boundary conditions...)
  + FlowSolution_t
    + GridLocation_t
    + Rind_t
    + Density/DataArray_t
    + VelocityX/DataArray_t
    + ...(other solution arrays)
  + FlowEquationSet_t
    + EquationDimension/"int"
    + GoverningEquations_t
      + DiffusionModel/
        "int[1+...+IndexDimension]"
      + GasModel_t
      + ViscosityModel_t
      + ThermalConductivityModel_t
      + TurbulenceClosure_t
      + TurbulenceModel_t
  + GlobalConvergenceHistory/
    ConvergenceHistory_t
  + RSDMassRMS/DataArray_t
  + CoefLift/DataArray_t
  + ...(other convergence parameters)
+ DataClass_t
+ ReferenceState_t
  + Mach/DataArray_t
  + Reynolds/DataArray_t
  + VelocitySound/DataArray_t
  + ViscosityKinematic/DataArray_t
  + Density/DataArray_t
  + Length/DataArray_t
  + ...(other reference parameters)

```

Figure 1 Example of Implementation

The overall architecture of CGNS is that of shared files accessible by the various software tools common to CFD: solvers, grid generators, field visualizers, and post processors. These applications must be modified,

using the API, to be able to access CGNS data. Each application then serves as an editor of the data, adding to, modifying or interpreting it according to that application's specific role. The user, or a locally developed user interface, remains responsible for executing the required actions and for the disposition of the files.

In 1998, ICEM CFD Engineering released the first version of the API. It supported structured topology, where multi-block connectivity could be either one-to-one abutting, mismatched abutting or overset. It defined standards for the storage of grid coordinates, flow solutions, boundary conditions, convergence history, and reference state. Dimensional units and non-dimensionalization information could be associated with each type of data. Additionally, it provided conventions for archiving the governing equations including the gas, viscosity, thermal conductivity, turbulence and diffusion models. It also provided for the recording of descriptive data throughout the file.

Two subsequent revisions of the CGNS Mid-Level Library were released by ICEM CFD Engineering. In June 1999, support for unstructured topology and geometry-to-mesh association were added to the API. Then in December 2000, rigid and arbitrary grid motion, as well as iterative or time-accurate data were implemented and released. Details of these CGNS extensions and data structures are given in Reference 8.

The CGNS elements are documented individually at the web site www.CGNS.org. All documentation as well as the software, both compiled and source code, is available at this web site. For those new to CGNS, the Overview¹, the CGNS System paper⁶ and the CGNS User's Guide¹⁰ are recommended reading.

2. Enhancements to the CGNS System

The following enhancements are currently being implemented into the CGNS system and scheduled for release in the first quarter of 2002:

1. Extension for Chemistry and Species
2. Extension for User-Defined Arrays
3. Linked CGNS Files
4. CGNS Version Control
5. HTML Documentation and Discussion Group

The enhancements include new functionality for CGNS as well as improvements for the support and development of CGNS. Several groups are working on these tasks. Intelligent Light is implementing the first three enhancements, ICEM CFD Engineering has implemented version control and NASA Glenn has implemented the web documentation and electronic

mailing list.

The following subsections describe these new enhancements, how they are being incorporated into the CGNS system, and how they affect the Standard Interface Data Structures (SIDS).

2.1 Chemistry and Species

The addition of chemistry and species data to the CGNS data file is necessary for many CFD simulations for turbomachinery and hypersonic flow.

The CGNS extension for chemistry and species is being developed with the following requirements:

1. Be backward compatible.
2. Keep the nodes for chemistry and species as simple and open as possible. Chemistry and species can be extremely complicated and specification of a standard to encompass all gas-fluid applications areas is not presently feasible. The nodes to be developed are to be open enough to allow the user to specify their specific models through the utilization of descriptor nodes and UserDefined nodes

This new CGNS extension requires modifying the structure of the GasModel_t node in FlowEquationSet.

Also the following new auxiliary nodes are being added to FlowEquationSet:

- ThermalRelaxationModel_t
- ChemicalKineticsModel_t

New data-name identifiers are being added for FlowSolution quantities. Modifications are being made to CGNS structures and Mid-Level library routines that utilize these nodes.

The extension for chemistry and species is currently in development.

2.2 User-Defined Arrays

The inclusion of user-defined data of arbitrary dimensions to the CGNS data file allows users to efficiently store additional information into the CGNS file. CFD solver developers have requested this extension since CFD solvers have additional data that could not be easily stored in the CGNS file due to its structure. This extension allows solver specific data to be stored into the CGNS file.

This extension introduces UserDefinedData_t as a node type to the CGNS library. This provides a parent node for an arbitrary number of user defined (numerical) data arrays of arbitrary size to be written.

The reading and writing of these arrays are managed with the existing DataArray_t support. There may be any number of DataArray_t nodes for any number of UserDefinedData_t parents. There may be an arbitrary number of indexed UserDefinedData_t nodes under their parents.

UserDefinedData_t is supported under the following nodes:

```
ArbitraryGridMotion_t
BaseIterativeData_t
BC_t
BCDataSet_t
BCData_t
CGNSBase_t
ConvergenceHistory_t
DiscreteData_t
Elements_t
Family_t
FlowEquationSet_t
FlowSolution_t
GasModel_t
GeometryReference_t
GridConnectivity_t
GridConnectivityItoI_t
GridCoordinates_t
GoverningEquations_t
IntegralData_t
OversetHoles_t
ReferenceState_t
RigidGridMotion_t
ThermalConductivityModel_t
TurbulenceModel_t
TurbulenceClosure_t
ViscosityModel_t
Zone_t
ZoneBC_t
ZoneGridConnectivity_t
ZoneIterativeData_t
```

UserDefinedData_t is not supported under the nodes:

```
DataArray_t
DataConversion_t
Descriptor_t
DimensionalExponents_t
DimensionalUnits_t
IndexArray_t
Range_t
UserDefinedData_t
```

It has the following properties:

```
Name: UserDefinedData# (or user defined)
Label: UserDefinedData_t
Data-Type: MT
Cardinality: 0,N
Child Nodes:
  Descriptor_t      (0,N)
  DataClass_t       (0,1)
  DimensionalUnits_t (0,1)
  DataArray_t:
```

Several API calls will manage the UserDefinedData_t nodes. They include:

```
C: ier = cg_user_data_write(char *user_data_name, int
*Index)
F: call cg_user_data_write_f(user_data_name, Index,
ier)
```

Where 'C' indicates a C routine and 'F' indicates a FORTRAN routine.

The routines above write a UserDefinedData_t node under the current location. The UserDefinedData_t node is given a name of up to 32 characters. This name may be from the SIDS, but this is not a requirement. An Index value is returned indicating this is the Nth UserDefinedData_t under the current parent node.

The routines

```
C: ier = cg_nuser_data(int *nuser_data)
F: call cg_nuser_data_f(nuser_data, ier)
```

return the number of UserDefinedData_t children under the current parent node.

Finally the routines

```
C: ier = cg_user_data_read(int Index, char
*user_data_name)
F: call cg_user_data_read_f(Index,
user_data_name, ier)
```

read the indexed UserDefinedData_t name under the current location. The internal name string is copied to the character argument and assumes that this memory has been allocated. The name may be up to 32 characters in length. The data arrays can then be read or written with the existing API functions cg_array_read(_f) and cg_array_write(_f).

The extension for user-defined arrays has been implemented and is currently in testing.

2.3 Links to other CGNS Files

A facility to support links to other CGNS files has been added to the CGNS library. These are wrappers around the ADF library routines to write and read links cast into the CGNS API.

```
C: ier = cg_link_write(char *nodename, char
*filename, char *name_in_file)
```

```
F: call cg_link_write_f(nodename, filename,
name_in_file, ier)
```

where the routines have the following character input arguments:

nodename = name of the link node (e.g., 'GridCoordinates')

filename = filename to use for the link

name_in_file = the name of the node which the link will point to (this can be a simple or a compound name, e.g., 'Base/Zone1/GridCoordinates')

The cg_link_write and cg_link_write_f routines will create a link at the current location. The node linked to does not have to exist when the link is created (but it may exist and that is OK). However, when the link is used, an error will occur if the linked to node does not exist. If filename is blank (null), the link is assumed to be a location within the current file. The link is written in the location previously set by cg_goto or cg_goto_f.

```
C: ier = cg_is_link(int *path_length)
```

```
F: call cg_is_link_f(path_length, ier)
```

where the routines have the following integer output argument:

path_length = 0 if current node is not a link, otherwise the value is the length of path string.

The cg_is_link and cg_is_link_f routines test if a node is a link. If the actual data-type of the node is a link, it returns the link path length. Otherwise, it returns 0. The link information is read from the location previously set by cg_goto or cg_goto_f.

```
C: ier = cg_link_read(char **filename,
                    char **link_path)
```

```
F: call cg_link_read_f(filename, link_path, ier)
```

where the routines have the following character output arguments:

filename = name of linked file

link_path = link path

The `cg_link_read` and `cg_link_read_f` routines get path information from a link at the current location. If the node is a link-node, return the path information. Else, return an error. If the link points to a location in the current file, then a blank filename is returned. The link information is read from the location previously set by either `cg_goto` or `cg_goto_f`. When the C version is called the user is responsible for freeing memory.

This functionality has been implemented and is currently in testing.

2.4 CGNS Version Control

One of the goals of the CGNS system is to provide the capability for multiple organizations to be involved in the direct development of CGNS. To accomplish this, a version control system is required during the software development. CVS was selected for the CGNS version control and has been implemented. Using CVS, you can record the history of your source files. This is crucial since bugs can occur when software is modified, and might not be detected for some time after the modification. With CVS, you can easily retrieve old versions to see exactly which change introduced the error. CVS stores all the versions of a file in a single file such that only differences between saved versions are stored. Every developer works in his/her own directory, and CVS merges the work when each developer is done. CVS is available from the Internet.

Access to the CVS controlled API source is given to developers selected by the CGNS Steering Committee, for implementation of predefined features. For more information see www.cvshome.org.

2.5 HTML Documentation & CGNSTalk

Previously sections of the CGNS documentation were provided as Adobe PDF files (SIDS, Overview, User's Guide, File Mapping Manual and the ADF User's Guide) and the Mid-Level Library manual was provided as a HTML document. NASA Glenn has revised all documents and made them available as either Adobe PDF or HTML files. All documentation is available on the Internet at www.CGNS.org. In

November 2001, the web site recorded 2019 page requests of the CGNS documentation from 430 distinct hosts in 33 countries.

A new feature at www.CGNS.org is the introduction of a CGNSTalk electronic discussion group where CGNS users can post messages to the CGNS team and other users via this email mailing list. Users can register at the Website to be added to the mailing list. As of the first of January 2002, the mailing list has 85 subscribers from 17 countries and 23 different commercial organizations.

3. CGNS is Open Software

The CGNS Steering Committee was formed to ensure the continuation of the CFD General Notation System. The Committee created a charter that lays out the mission, organization and governing principles of the Steering Committee. The charter can be found at www.CGNS.org. The charter defines several responsibilities of the CGNS Steering Committee that support its fundamental mission. They include:

- maintaining the software implementing the Standard, documentation of the Standard and an open web site to disseminate information on the Standard,
- providing mechanisms for the evolution of the Standard,
- promoting the acceptance of the Standard,
- providing user services, and
- determining the means to fund activities in support of the Standard.

The CGNS Steering Committee is organized as a voluntary organization that governs by consensus. The Committee meets at a minimum of once per year, and is represented by a Chairperson. The current Chairperson is Robert Bush of Pratt & Whitney. The Chairperson coordinates activities, facilitates meetings and serves as a focal point for the Committee. The Charter explicitly states that all parties are welcome to bring forward issues and participate in the development of the Standard, whether or not they are members of the Steering Committee. The CGNS Steering Committee is made up of representatives from specific institutions, rather than individuals. The current members of the Steering Committee are:

- Airbus
- Boeing Commercial
- Boeing Phantom Works
- Boeing Space & Communications
- CD ADAPCO Group
- Fluent, Inc.
- ICEM CFD Engineering

- Intelligent Light
- NASA Ames
- NASA Langley
- NASA Glenn
- NUMECA International
- Pointwise, Inc
- Pratt & Whitney Aircraft
- Rolls-Royce Allison
- US Air Force

The CGNS charter also lays out the governing principles that are to be followed. In general, the charter calls on the Committee to adhere to the philosophy used for *free open software*, as defined under the GNU Lesser Public License. In particular, the user can:

- Run the program, for any purpose
- Study how the program works, and adapt it to one's needs
- Redistribute copies so one can help their neighbor
- Improve the program, and release improvements to the public, so that the whole community benefits

This open software standard helps to promote the CGNS Standard by allowing users the freedom to redistribute copies, with or without modifications, either gratis or charging a fee for distribution, to anyone anywhere.

4. Status of the ISO standards effort

The effort to establish an ISO/STEP standard⁷ for fluid dynamics data, based on CGNS, continues to progress. In February 2001, ISO Standing Committee 4 (SC4), *Industrial Data*, accepted the CGNS project as a New Work Item. As a consequence, this project was assigned the designator *AP (Application Protocol) 237 Fluid Dynamics*.

The ISO standards process is based on modular and reusable standards. This concept means, for example, there will only be one ISO standard for mesh-based data, and that standard will apply in any technical discipline that requires representation of mesh-based data. Therefore, any initiative that develops such a standard must do it in such a way to anticipate and facilitate future use of the data definitions and data structures by other disciplines. In the case of the Fluid Dynamics standard, mesh-based data structures are being defined for the first time, and therefore steps must be taken to avoid inadvertently producing a standard that meets the needs of the fluid dynamics community but is too limited to be of value to other technical communities. In this respect, modular and reusable standards, the ISO process has requirements that were not considered in developing the CGNS standard.

The ISO standards process also has strong requirements for integration into a formal product data management system. CGNS did not address such a requirement. To meet this need, we are taking various product data management structures first developed in AP 209 (Finite Element Analysis) and repackaging them in Part 53 as general structures for numerical analysis.

To accommodate these and other constraints, the Fluid Dynamics standard will consist of four parts, each of which will be developed and approved as a stand-alone ISO standard. These four parts are:

- AP 237 Fluid Dynamics. This is the top-level standard, and it defines data types and data structures that will be used throughout the field of fluid dynamics (for example, in both computational and experimental fluid dynamics).
- Part 110 Computational Fluid Dynamics. This part defines data types and structures that are unique to CFD, for example, grid-based discrete data, boundary conditions, temporal and spatial iteration and convergence data.
- Part 52 Mesh-Based Topology. This part defines structured and unstructured grids including topology, connectivity, boundary conditions, cell and vertex numbering schemes, etc.
- Part 53 Numerical Analysis. This part defines links to product data management structures and configuration control for numerical analysis, including linkage between a specific analysis and a specific physical problem, a specific nominal geometry model, and other features of numerical analysis.

It is anticipated that eventually another 100-series part will be created, under AP 237, to address specialized needs for experimental data in fluid dynamics, complementing Part 110 for CFD data. We also anticipate creating a Usage Guide, which is not a formal part of the AP 237 standard. This Usage Guide will consist of tips and examples, and it should be very useful to those seeking to use the standard. Usage Guides are not an ISO requirement, but they have been prepared for several other AP standards.

The AP 237 fluid dynamics standard is linked to AP 203 for geometric data, and AP 209 for finite element structural analysis data. At present, no ISO standards initiatives are in progress in other related areas that would utilize CFD data, for example, thermodynamics data, electromagnetics data, etc. If and when standards initiatives are launched in these areas, we expect they will be able to re-use Parts 52 and 53.

The last meeting of the Engineering Analysis

Committee, which is a part of SC4, occurred in October 2001. At that time, a completion schedule was projected for the four parts discussed above. It is expected that the three lower-level parts (52, 53, and 110) be completed and submitted for final ballot approval as International Standards in the second half of 2003. Preparation of the top-level AP 237 standard is a larger task, and it is projected that final balloting will occur in the second quarter of 2005. Each of these standards will reach iDraft International Standardi status about a year before the final ballot, and at that point the standards should be stable, and very close to their final form.

The intellectual content of the ISO AP 237 standard, and its subsidiary parts (52, 53, and 110), is being developed by the CGNS Steering Committee, which is also chartered as the CGNS subcommittee of the AIAA Committee on Standards for CFD. The mapping of this CGNS intellectual content into the requirements of the ISO standards process is being led by Raymond R. Cosner, of The Boeing Company.

Extensive draft documents are available for all four parts of the Fluid Dynamics standard. Comments are welcomed on any aspect of these drafts. Copies of the draft documents, in Adobe Acrobat format, can be obtained by contacting any member of the AIAA Committee on Standards for CFD and are also posted on the web at www.CGNS.org.

In a related development, the AIAA Committee on Standards for CFD has agreed in June 2001 to publish the CGNS standard as an AIAA Recommended Practice.

5. Utilization of CGNS

The global CFD community has welcomed CGNS since its introduction in 1998. Engineers and scientists from academia, industry, and government organizations are utilizing the CFD General Notation System. At the time of writing, 591 users from over 25 countries have registered at the CGNS web site.

The CGNS Steering Committee is aware of several CFD applications from commercial, industrial and government organizations in United States and Europe that have implemented the CGNS standard successfully: Table I shows the organizations and CFD applications that are using CGNS. The table is organized to show the organizations broken down as either commercial, industrial or government. The table is also ordered to show the stage (grid generation, solver or post processor) of the simulation process that the application addresses. The table also shows if the application reads a CGNS file, writes a CGNS file or

does both. CGNS is being utilized at government labs in United States, France and Germany (NASA, ONERA and DLR) indicating that CGNS is being accepted in the government research community. It is also observed from this table that the majority of the industrial and government organizations are exporting CGNS data files for their solver applications. They use commercial software tools for grid generation and post processing.

A limitation in using CGNS in previous years was that the commercial software tools did not support CGNS. However from Table I it is observed that the number of commercial applications using CGNS is on the same order as the number of industrial/government applications, indicating that the issue of commercial applications not supporting CGNS is starting to be resolved. The table shows over twenty diverse organizations are using CGNS indicating it is becoming a standard.

A survey was made to determine how various organizations use CGNS. Descriptions of CGNS utilization by Boeing, ONERA, Pratt & Whitney and NASA Langley follow in the next subsections.

Boeing

Boeing has identified the CGNS data standard as their common format for CFD data at the level of Euler and Navier-Stokes analysis methods. New tools and processes are being developed to conform to CGNS, and existing tools and processes are being converted to this standard. Boeing sees the primary benefits of using a common data standard, such as CGNS, in the following areas:

- Enabling component tools in the CFD process to be used in any combination according to the needs of the task at hand,
- Avoiding the need for specialized data translation steps, which often is the source of limitations or errors,
- Establishing a standard interface to pass CFD data into downstream processes, such as aerodynamic loads analysis, which is independent of the tools used to produce the CFD data, and,
- Establishing definitions and representations of the fluid dynamics data in a CFD solution, such that the precise meaning of each variable will be understood at any time in the future, without reference to any notes ñ the file plus the CGNS standard documentation defines the meaning of each element of data.

Organization	Application	Grid Generator	Solver	Post Processor	Read CGNS Files	Write CGNS Files
Commercial						
Advanced Visual Systems	AVS/Express			X	X	
AEA	CFX5 CFX-TASCFlow		X X			X X
Aerosoft, Inc	GASP V4		X	X	X	X
Amtec	Tecplot			X	X	
CD ADAPCO	STARCD		X			X
CEI	EnSight EnSight Gold			X X	X X	
Fluent Inc	Fluent		X	X	X	X
ICEM CFD	ICEM CFD HEXA OPTIMESH	X X			X X	X X
ICEM CFD MIT	Visual3			X	X	
Intelligent Light	FIELDVIEW			X	X	
NUMECA	IGG EURANUS CFVIEW	X	X	X	X	X X
Pointwise Inc.	GRIDGEN	X				X
Government						
DLR/MTU (Germany)	TRACE		X			X
NASA Ames	PEGASUS/ OVERFLOW		X			X
NASA Ames	PLOT3D			X	X	
NASA Ames	Cart3D		X			X
NASA Glenn	NCC		X			X
NASA Langley	CFL3D		X		X	X
NASA Langley	UPS		X			X
NPARC Alliance	NPARC WIND		X X		X	X X
ONERA (France)	elsA		X		X	X
Industry						
Boeing-Rocketdyne	APPT		X			X
Boeing - Seattle	TLNS3D		X			X
Boeing ñSt. Louis	CFE		X			X
Rolls-Royce Allison	ADPAC		X			X
Rolls-Royce/ Oxford University	HYDRA		X			X

Table I ñ Organizations and Applications Using CGNS

ONERA

A sub-committee at ONERA¹¹ is discussing object-oriented interfaces (most notably C++ and Python) to CGNS. While a general consensus has not been reached about the most sensible approach to define them, implementation for both languages is underway. Basic functionality is available and the full implementation (at least for C++) is expected to be publicly available in January 2002. The experience of the community with these preliminary efforts will guide the strategy of the CGNS Steering Committee regarding such alternative interfaces.

ONERA has also developed applications with CGNS. The electromagnetism and radar departments use it to store data results. The aerodynamics and aeroacoustics simulation departments have developed Python¹³ tools and translated elsA¹² validation cases database in a CGNS format. These tools are built on the freely available pyCGNS¹⁴ package.

The validation database contains grids, which have different formats for a given configuration (every other point, 2D/3D, multi-zone...), initialization data and solution data. The CGNS tree structure allows consistency between these files, for a given test case. The link support allows the sharing of grids, initialization data and makes it possible to use several reference sessions of test with separate and structured solutions for each session. More than 300 test cases of the software validation have been translated.

The elsA software is expected to be fully CGNS compliant by the end of 2002. During this time, some studies have been planned in order to use CGNS as a basis for software interoperability with aeroelasticity solvers but also as a basis for large amount of data storage and retrieval.

Pratt & Whitney Aircraft

As a world leader in developing gas turbine engines for commercial and military propulsion applications, Pratt & Whitney is on the forefront of enabling technologies and relies heavily on CFD generated data for product design. The broad range of CFD applications requires leveraging of proprietary, commercial, collaborative and government supplied tools to meet the diverse demands with the most up to date capabilities. CGNS will play a key role in bringing these tools together in a streamlined design system. CGNS will meet varied requirements for data archive, communication and process automation by providing a stable, efficient, consistent and industry standard means of representing CFD data.

The first step towards CFD data standardization at Pratt & Whitney has been the development of conversion routines that do a full translation of grid, solution and boundary condition information between CGNS and the existing internal P&W CFD data format. These conversion routines allow us to leverage all existing tools and processes to maintain capabilities while changing processes to utilize CGNS. Anticipated conversion of commercial, collaborative and government tools to CGNS will allow us to streamline our design processes by eliminating data conversion errors and the associated cycle time. Simultaneously, we will convert our proprietary tools and internal data representations to the CGNS standard. Full conversion will facilitate automation of design processes reliant on CFD by providing a consistent source and destination for the volumetric data generated.

NASA Langley

CGNS capability has been added to the NASA Langley Research Center code CFL3D⁹ Version 6.0. Currently, the CGNS file is an optional file that replaces the old grid and restart files. The CGNS file not only contains all the information necessary for a restart run, but it also can be read and used by any utility with CGNS capability, including post processor plotting packages. There are a few limitations and features particular to the current implementation in CFL3D, listed below.

In the current CGNS implementation, when CFL3D is run with mesh sequencing and is stopped at a level lower than the finest grid, there is no easy way to handle this with CGNS since the CGNS file contains the fine grid only. With "standard" CFL3D, the restart file would contain the coarser solution only. With CGNS, it was decided to write out the coarse-grid FlowSolution on the fine grid, with data repeated in neighboring cells. Hence, if you plot the solution, it will be identical to the coarse grid solution, only all the fine-grid gridpoints will be used. A descriptor node called "Information" under "FlowSolution" says something like: Every-2nd-cell solution (1 level down) to indicate that the solution is actually on a coarser level. There is currently no SIDS standard for handling this situation.

Note that in the current release, PATCHING and OVERSET information have not yet been included in the CGNS-part of CFL3D. In other words., the CGNS files are purposefully missing this information. The user can still run patched and overset cases with CGNS, but the patched/overset connectivity information is never put into the CGNS file itself. This is similar to the "normal" mode of running CFL3D. In the "normal"

mode, the user must have available a grid file, restart file, patched-info file, and overset-info file when running with patched/overset. With CGNS, the user must have available a CGNS file, patched-info file, and overset-info file.

Multiple-time flow field information for time-accurate runs is currently not written to the CGNS file. Only the latest (most recent) flow field data, necessary for a clean restart, is given. For the time-being, standard PLOT3D-type files are always still output at the end of a run. This redundant output procedure will continue until CGNS is implemented more widely in post processor utilities.

CFL3D currently writes out solution information to the CGNS file both at cell centers as well as within one rind cell at all boundaries. This is necessary in the long-term so that plotting packages will be able to extract correct boundary data (e.g., velocity=0 on solid surfaces) from the CGNS file. However, note that at the current time the BCs are not updated immediately prior to writing out the CGNS file, so the rind cells are "behind" by one iteration; this can mean imperfect matching at interfaces or slightly non-zero velocities on solid surfaces when plotting from the CGNS file, particularly if the solution is far from convergence.

The CGNS implementation in CFL3D follows the recommendations outlined in Appendix D of Reference 10 to improve the likelihood that other external applications will be able to easily use the resulting CGNS files without a too-extensive array of checks and translators.

6. Future Directions for CGNS

The CGNS charter calls for changes and additions to the CFD General Notation System. CGNS has been developed with the key concepts of flexibility and extensibility in mind. In order to address a perceived need or deficiency in the software, a proposal must be submitted to the CGNS Steering Committee. The proposal is then presented in an open and public forum. The CGNS Steering Committee is responsible for accepting, altering or rejecting the proposal, and determines the timetable for implementation. The primary requirement for any proposal is to maintain code compatibility with the existing CGNS Standard. Currently there are several proposed CGNS extensions still at the discussion level. They include the following features:

- multigrid
- rotating coordinates
- periodic boundary conditions

- wall functions,
- 2D axisymmetry data
- Cartesian data
- object-oriented interfaces
- parallel implementation

Several of these proposed extensions are posted on the web at www.CGNS.org. Proposals for modifications and extensions of CGNS may be submitted by anyone.

To expedite the implementation of the proposed extensions, one of the CGNS Steering Committee's long-term goals is to have multiple organizations involved in the development of CGNS. CGNS is iOpen Software similar to LINUX and is available to the public. Information on CGNS (source code, executables, sample CGNS files, documentation, etc.) can be found at www.CGNS.org.

Another long-term goal is to see CGNS used globally. CGNS is currently not used at a significant level in Japan/Asia. We hope to promote its utilization through publications and the Internet. CGNS will be utilized in all parts of the world as more commercial applications adapt CGNS.

The CGNS standard is also the object of an ISO standardization effort for Fluid Dynamics data. This project has been accepted by the ISO organization and is progressing. Final balloting on the ISO AP 237 standard is anticipated to occur in the second quarter of 2005.

Finally, the AIAA Committee on Standards for CFD has agreed in June 2001 to publish the CGNS standard as an AIAA Recommended Practice.

7. Concluding Remarks

This paper described the status of the CFD General Notation System. CGNS has grown into a data standard sufficient to support a majority of CFD applications while providing for easy data exchange between sites, computing environments and applications. New functionality is currently being added to the CGNS system. The Standard Interface Data Structures (SIDS) and corresponding Application Programming Interface (API) are being extended to support chemistry and species, arbitrary user data and linked CGNS files. Version control has been added to allow CGNS source development by multiple organizations. The support of CGNS has been enhanced through providing CGNS documentation in either HTML or Adobe PDF files and a CGNS electronic discussion group where CGNS users can now post messages to the CGNS team and other

users using this email mailing list. An effort is underway for CGNS to be adopted as an ISO standard for the recording of aerodynamic data. The utilization of CGNS by commercial, government and industrial organizations was reported with a focus on how CGNS is being used at Boeing, ONERA, Pratt & Whitney and NASA Langley. Future directions and goals for CGNS were discussed.

References

1. CGNS Team, "The CGNS System Overview and Entry Level Document", Draft, Version 1.0, May 1998.
2. Allmaras, S., "CGNS Standard Interface Data Structures", Draft, May 1997.
3. CGNS Team, "The ADF User's Guide", May 1997.
4. CGNS Team, "SIDS-to-ADF File Mapping Manual", Version 1.1, June 1999.
5. Poirier, D., "CGNS Mid-Level Library", December 1999.
6. Poirier, D., Allmaras, S., McCarthy, D., Smith, M., Enomoto, F., "The CGNS System", AIAA Paper 98-3007, June 1998.
7. "Guidelines for the Development and Approval of STEP Application Protocols", document number ISO TC 184/SC4N535:1998(E), dated 18 December 1998.
8. Poirier, D., Bush, R., Cosner, R., Rumsey, C., McCarthy, D., "Advances in the CGNS Database Standard for Aerodynamics and CFD", AIAA Paper 2000-0681, January 2000.
9. Krist, S. L., Biedron, R. T., Rumsey, C. L., "CFL3D User's Manual (Version 5.0)", NASA TM-1998-208444, June 1998.
10. Rumsey, C.L., Poirier, D.M.A., Bush, R.H., Towne, C.E., "A User's Guide to CGNS", NASA/TM-2001-211236, October 2001.
11. <http://www.onera.fr>
12. Cambier, L and Gazaix, M., "An Efficient Object Oriented Solution to CFD Complexity", AIAA Paper 2002-0108, Jan. 2002.
13. <http://www.python.org>
14. <http://elsa.onera.fr/CGNS/releases>