Table of Contents

Edited by 22nd ITTC QS Group 1999	Approved
ITTC 1996 21st pp 485 - 487	21st ITTC 1996
Date	Date

CFD User's Guide

1. PURPOSE OF PROCEDURE

„The provision of the necessary information allowing the user to perform CFD calculations of known Quality Characteristics as defined in ISO 9126 "Description of quality characteristics of CFD Software packages".

2. USER'S GUIDE

2.1 Introduction

This section summarises and documents the quality characteristics of the CFD software package. The quality of a CFD prediction depends on the pre-processor, the core model and the post-processor. It must be clear which of these components the USERS'S GUIDE addresses. It is recommended that all 3 components must be documented in the user's guide. The quality characteristics can be found in ISO 9126 :

i) Functionality
The software should provide information using parameters and nomenclature which are compatible with establish design practice. Parameters of interest include the following

ii) Reliability
CFD software must be capable of perform calculation within a specified range and consistent level of level accuracy.

iii) Usability
The software should be geometry independent, catering for the complicated 3D shapes formed by the hull/appendage combination. Output should be easily accessible. As well as providing data for dedicated post processing devices, it should be possible to output the result in a standard format so that it may be easily transferred to other systems. Where it is not possible for the user to establish values for input variables, the code must be able to generate starting values to allow the application to proceed.

iv) Efficiency
-The ability to generate and manipulate data with the minimum of effort.
-Elapsed run times should be minimised so that they are compatible with normal working cycles.
-A user must be able to perform a CFD analysis on workstation.

v) Maintainability
Maintenance of the code should not be the responsibility of the user. The software developer must be able to respond to the needs of the user by providing upgrades to reflect the correction of errors and the incorporation of new features.

vi) Portability
As far as possible the software should be machine independent.

2.2 Scientific Documentation Summary

The Proposed Standard

The Standard is presented in form of a table of contents including comments and description of what the section should contain in more detail. Items in italic are document heading.

Functionality
This section describes the functionality of the software broken into sub-sections. Wherever applicable claims about assumptions, functionality, etc. and substantiation of these claims should be stated.

Physical model
This section describes the physical system or systems being modelled such as geometric configuration, phenomena, etc.

Conceptual model
This section describes the assumptions made when deriving the conceptual model based on the physical model. Particular emphasis should in this section be made on assumptions like turbulence closure models etc.

Numerical model
The numerical model is a discrete representation of the conceptual model suitable for implementation as a computer code. This section defines the schemes adopted for this purpose such as spacial discretization, time integration, solution procedures for algebraic equations, etc.

Implemented model
This section describes technical aspects of the implementation that are relevant to the quality of the results that the software produces. These aspects can include coding of algorithms, use of standard mathematical software, software development tools, etc. A flow chart is recommended as additional information to be included in this section. Also the Quality Assurance procedures adopted during development and maintenance should briefly be described.

Reliability
A very important aspect of the reliability of a software product is the uncertainty of the results that the product produces. The standard discussed and proposed in chapter 4 shall be the frame work. The standard is in preparation by ASME.

Usability
The usability for a CFD software package of importance to the end-user are: user interfaces, data interfaces to CAD software and other software. Claims and substantiations of the range of practical geometries and conditions for which the software package will be useful must be stated.

Efficiency
Based on example runs, typical CPU times and memory requirements should be stated. Simple graphs showing the relationship between CPU time and grid cell number would be useful as well as graphs depicting other relationships of similar importance to the end-user.

Maintainability
A user will certainly be interested in how a software package will and can be maintained and upgraded in the future. It should be stated if possible what effort is expected when new numerical procedures, conceptual models, etc. are implemented in the software package.

Portability
The portability, including the scalability of the code is important to the end-user with respect to protecting his investment in training, data generation, etc. by being able to use the same product in the future. Items to be addressed are: porting the code the computers having different architectures (vector, parallel, massive parallel machine), range of machines on which the present product runs with out modifications (compatibility), etc.

2.3 User's Instructions

Having read, hopefully before purchasing the software package, the documentation summary provided in section 2 the user shall be able of achieving the results of the same quality characteristics by him self by following the instructions set forth in this section.

Getting Started
In a very brief form the typical and recommended implementation procedures when using the software product should be described. This includes: installation, how to reproduce sample results as shown in section 3.3 etc.

Input/Output Reference
All input variables that the user requires should be documented. Typical input variables can be group into:

Geometry, Conditions and Solution control parameters. The range of validity and guidelines for setting optimum values shall be included. This is very important as many data can not be kept constant independent of the actual condition. Examples of this are: how many grid cells in each dimension should be used as function of geometry, speed etc., relaxation factors can in some situations be difficult to set in order to get converted and stable results, etc. A format for documenting each input variable can be derived when elaborating the User's Guide Standard.

Output data should be clearly defined, either self-contained or in the User's Guide.

Sample Applications
Sample applications shall be provided in as many variations as necessary. Sample applications shall contain input as well as output data. Claims and substantiation justifying the choice of each important and non-obvious input variable shall be given. The output shall be presented in a format that allows the user to identify the quality characteristics according to ISO 9126 that varies as function of input data setting. This includes compliance with the quality assurance procedures as recommended in section 3.3. If proper documented the Sample Applications section will also serve as a file for keeping and documenting benchmark and validation data calculations.

Quality Assurance Procedures
In this section, recommendations on Quality Assurance procedures that must or should be made on a routine basis or for selected cases to allow the user to feel confident about his capabilities of running a particular code should be clearly defined as an easy to follow recipe. This could include:

1) grid independence checks,
2) model parameter independence checks,
3) numerical parameter independence checks.