NONLINEAR FREE-SURFACE FLOW SOLVER

Users' Guide

D.C. Scullen and E.O. Tuck

Scullen & Tuck Pty Ltd
PO Box 237 Rundle Mall, Adelaide, South Australia 5000

May 1, 2002

Abstract

The program `NFSFS` can be used to solve for flow about three-dimensional bodies of arbitrary shape moving beneath an otherwise-calm free surface. This document describes its operation.

1 Introduction

Scullen and Tuck have developed a program `NFSFS` ("Nonlinear Free-Surface Flow Solver") which solves for steady irrotational flow of an inviscid incompressible fluid about a given body beneath a free surface under gravity. This program generates the disturbance due to the body by a finite but large set of discrete point-source singularities located externally to the fluid domain, namely inside the body and above the free surface, whose strengths are determined. The body and free surface are represented by collocation nodes, and the Neumann boundary condition is enforced on the body boundary, while the kinematic and fully-nonlinear dynamic boundary conditions are satisfied at the free-surface nodes. Since the free-surface location is unknown a priori, a Newton-like iterative procedure is adopted, whereby an approximation to the free surface is refined successively until convergence is achieved at a quadratic rate.

For specifics of the problem description and its solution procedure within `NFSFS`, the reader should refer to [1] which is available via the internet.

2 Operation, Inputs and Outputs

The source code consists of the file `nfsfs.f` and a series of FORTRAN include files. (The High-Performance FORTRAN file `nfsfs.hpf` is supplied as an alternative to `nfsfs.f`. The difference between the two is that `nfsfs.hpf` makes use of the faster `forall` construct which is unavailable in standard FORTRAN 90 implementations.) The code is of fixed format with lines of length greater than 72 characters, and it may be necessary to supply flags to indicate this when compiling. The main program `nfsfs.f` contains a call to the `NAG` library's linear-system solver `F04ARF`. If the `NAG` library is unavailable to the user, then the code contains instructions on how to replace the call to `F04ARF` with a call to the `LAPACK` linear-system solver `DGESV`. One can obtain a copy of the public-domain `LAPACK` library from the internet. An executable is produced by compiling `nfsfs.f`, linked with the linear-system solver's library.

Almost all program input is achieved through files, with the only interactive input being a response to the question: "`Do you have a converged solution as input?  (n/y):`". The normal response of "`n`" indicates that the free-surface data contained within the input files corresponds to an initial guess at the solution, and sets the program on its course of determining an accurate, converged solution. The alternative response "`y`" instructs the program that the input files contain data from a previously determined converged solution, and the program then moves directly to the post-processing stage. In this initial release of `NFSFS`, the only post-processing performed is determination of the fluid velocity at a given set of points.

Input File Manifest

The following list contains the files used for input, and a description of their purpose and format. Example input files have been supplied with the source code.

- `constants.dat`
  Used for setting the values of the free-stream speed U and the acceleration due to gravity g. The file contains two lines. The first entry of the first line is the value of U, the first entry of the second line is the value of g. Subsequent entries on any line are ignored and may be used for the purpose of descriptive comments. If this file is not present, then both quantities are assumed to be unity.

- `body.dat`
  Describes the geometry of the body. The body is represented by a distribution of points (body nodes) and surface normal vectors. This file contains the coordinates of the body nodes and the components of the normals. The first line contains the hash character "`#`" followed after a space by the number of points used to represent the body. Each subsequent line contains the x, y, and z coordinates of a body node followed by the x, y, and z components of the unit normal to the body at that body node. There is one line for each body node.

- `guess_surface.dat` or `converged_surface.dat`
  These files describe the free surface which is represented by a distribution of points (surface nodes) at which both the dynamic and kinematic free-surface boundary conditions are to be enforced. They contain the coordinates of those surface nodes. The program may be executed in order to determine the free surface, or it may be executed when the converged free surface is already known in order to compute some other quantity in the post-processing stage, and it takes one or other of these files as input, depending on the answer to the question: "`Do you have a converged solution as input?  (n/y):`". In the first case it takes as input the file `guess_surface.dat`. In the second case it takes as input the file `converged_surface.dat`. Both files are of the same format. The first line contains the hash character "`#`" followed after a space by the number of points used to represent the free surface. Each subsequent line contains the x, y, and z coordinates of a free-surface node. There is one line for each free-surface node.

- `radiation_pts.dat`
  Describes the locations at which the radiation condition is to be enforced. It is a property of steady free-surface flow problems that the

solution is not unique. In general, the desired solution is the one that does not have waves ahead of the body, and that particular solution is selected by enforcing a radiation condition. This file contains the index of the free-surface nodes at which the radiation condition is enforced. The first line contains the hash character "`#`" followed after a space by the number of nodes at which the radiation condition is to be enforced. Each subsequent line contains the index of a free-surface node at which the radiation condition is to be enforced. The index is an integer which identifies the corresponding surface node, based upon the order in which the surface nodes are supplied in the files `guess_surface.dat` and `converged_surface.dat`. There is one line for each such free-surface node.

- `guess_singularities.dat` or `converged_singularities.dat`
  These files describe the potential which is represented by a distribution of point singularities located externally to the fluid domain, both inside the body and above the free surface. Each singularity may be either a point source, i.e. of type `three_d_source`, or a point source with an image reflected in the plane $y = 0$ and of identical strength, i.e. of type `three_d_source_with_image`. This file contains the types, coordinates and strengths of the singularities. The program takes as input one or other of these files, depending on the answer to the question "`Do you have a converged solution as input?  (n/y):`". If the program is executed in order to determine the free surface, then `guess_singularities.dat` is supplied as input. If the program is executed when the converged free surface is already known, in order to compute some other quantity in the post-processing stage, then `converged_ singularities.dat` is supplied as input. Both files are of the same format. The first line contains the hash character "`#`" followed after a space by the number of singularities used to represent the potential. Each subsequent line contains the type, the x, y, and z coordinates and the strength of a singularity. There is one line for each singularity.

- `relaxation_factors.dat`
  Controls the extent to which the surface nodes and singularity strengths change between iterations. Because the solution process is iterative, with both the free-surface location and potential being successively re-

fined, there is the possibility of divergence from the desired solution if too large a change is made, at least in the earlier iteration stages. This file contains a factor by which this change can be reduced if desired. No change corresponds to a relaxation factor of zero. No relaxation corresponds to a relaxation factor of one, which is the most desirable situation and normally leads to a quadratic convergence rate. Any relaxation factor between zero and one should give convergence, but at a slower rate. The first line contains the hash character "`#`" followed after a space by the number of relaxation factors. Each subsequent line contains the value of the relaxation factor to be used for the corresponding iteration. Once all relaxation factors have been used, relaxation is not used in subsequent iterations.

- `partition_size.dat`
  Determines the amount of memory to allocate to working arrays. The program has been written in FORTRAN 90 and is intended for use on super-computers. Significant effort has been made to allow for true parallel processing on multiprocessor machines. A consequence is that large arrays are used to store interim values, whereas for a single-processor serial machine, interim values could have been stored in single variables. This has the effect of dramatically increasing the memory requirements of the program. To ease this burden, a parameter has been introduced that allows variation in the size of these arrays and hence also the extent of parallelisation. Specifically, this value determines the extent of one dimension of the large working arrays. A small value reduces the memory requirements, but also degrades the performance towards the level of a serial-processor computer. A large value allows for significant parallel processing, but increases the memory requirements. This file contains a single integer with the value of this parameter; a value of 32 is recommended in almost all cases, and only needs to be decreased if there are severe memory constraints.

In instances where input files are not supplied, assumptions are made about the intention. For files `body.dat`, `guess_surface.dat`, `converged_surface.dat` and `radiation_pts.dat`, the assumption is that the number of entries corresponding to that file is zero. For example, for flow about a body in an infinite fluid, as may be the case for a deeply submerged body or an aircraft, there is no free surface. For the file `constants.dat`, the assumption is that both the free-stream speed and acceleration due to gravity are unity. For the file `relaxation_factors.dat`, the assumption is that there is no relaxation in any iteration, so that the relaxation factor is always unity. For the file `partition_size.dat`, the assumption is that the partition size is 32. One of the files `guess_singularities.dat` and `converged_singularities.dat` must be supplied as input, and the program ceases execution if it detects that this is not the case.

Output File Manifest

- `converged_surface.dat`
  Describes the solution free surface. See the entry "`guess_surface.dat` or `converged_surface.dat`" above for a description. This file is produced as output if the response to the initial question is "`n`", i.e. if the user does not have a converged solution for input.

- `converged_singularities.dat`
  Describes the solution singularities. See the entry "`guess_singularities.dat` or `converged_singularities.dat`" above for a description. This file is produced as output if the response to the initial question is "`n`", i.e. if the user does not have a converged solution for input.

- `surface_pressures.dat`
  Contains a record of the extent of dissatisfaction of the dynamic and combined kinematic-dynamic free-surface boundary conditions at each iteration. The aim of the iterative procedure is to determine the surface node locations so that the appropriate free-surface boundary conditions are satisfied. This file contains, for each iteration, the x, y and z coordinates of each surface node, and the value of the pressure p and its material derivative $\frac{Dp}{Dt}$ there, both of which should be zero for a converged free surface. It is these quantities that are minimised by the iterative procedure.

- `singularity_strengths.dat`
  This file contains, for each iteration, the type, the x, y and z coordinates and the strength of each singularity.

- `current_surface.dat`
  Used to determine the shape of the free surface in the instance when

the program terminates without convergence (and before outputting the associated values of p and $\frac{Dp}{Dt}$). This file contains, for the final iteration before termination, the x, y and z coordinates of each free-surface node.

- `error_message.dat`
  If the program terminates prematurely, then an associated error message is written to this file.

Post-Processing Input File Manifest

Post-processing may be performed once the solution free surface and potential are known. These may have been determined from an initial guess through the iterative solution process, or may have been supplied as converged inputs. In the current version of the program, the only post-processing performed is the evaluation of the component of fluid velocity in a specified direction at specified points.

This option can be used with fully-converged solutions to determine the extent of satisfaction of the Neumann boundary condition at locations on the body surface other than those nodal points at which the boundary condition has been enforced. By doing so, one may determine useful information on the accuracy of the solution for a given finite but large number of nodal points. Large flows normal to the body surface between nodal points would indicate that the overall accuracy of the solution is likely to be poor with this many nodal points, and more nodal points may be needed.

- `velocity_components.dat`
  This file contains coordinates of points together with directions, such that the velocity in that direction is to be computed at that point. For example, the points can be on the body and the directions normal to the body. Such points are then typically taken from a representation of the body with a finer resolution than that for which the flow was solved, in order to provide a measure of the accuracy of the solution. The first line contains the hash character "`#`" followed after a space by the number of evaluation points. Each subsequent line contains the x, y, and z coordinates of an evaluation node followed by the x, y, and z components of the unit vector in the direction for which the velocity component is desired at that evaluation node. There is one line for each evaluation node.

Post-Processing Output File Manifest

The only output file created by the post-processing stage of the program is `velocity_residuals.dat`.

- `velocity_residuals.dat`
  The component of velocity at the locations and in the directions as supplied in the file `velocity_components.dat`. Typically the component of velocity normal to the body surface.

3 Notes on Program Operation

The simplest case to run and solve is one which determines the velocity potential for flow about a body that is alone in an infinite fluid. In this case there is no free surface, and therefore the solution is obtained in the first iteration. The only necessary input files are `body.dat` and `guess_singularities.dat`. If the file `constants.dat` is not supplied, then the free-stream speed is assumed to be one.

The flow can also be determined about an arbitrary number of bodies, simply by representing all bodies at once within the file `body.dat`.

In cases where lateral symmetry about $y = 0$ can be assumed, it is computationally expedient to use the singularity type `three_d_source_with_image` and to represent only one half of the body and free surface. This results in a system of equations of half the size that would otherwise be the case, and a consequential reduction in computational effort by a factor of up to 8.

One method for determining if the body representation is accurate is to solve for the flow about the body in an infinite fluid, i.e. without a free surface. By examining the component of velocity normal to the body's surface at locations other than where the boundary condition is enforced, one may measure the level of overall satisfaction of the Neumann boundary condition, and hence the likely accuracy of the result. Once the body representation is found to be satisfactory, the user may then proceed to include the free surface and to solve for the resulting free-surface shape.

The free surface should be represented by a rectangular grid, with the radiation condition being applied at two of the most upstream rows; we recommend rows 1 and 3. Since the program finds the potential by inversion of a system of linear equations, in order for the number of unknowns to match the number of equations, it is necessary that the number of singularities equal the sum of the number of body nodes, surface nodes, and radiation condition nodes. Additional singularities are required, and are typically placed one row upstream of the free surface and one row downstream.

A certain amount of experimentation may be required by the user in selection of singularity locations to get the most accurate solution. A reasonable guideline is that singularities should be located above the free surface by a distance approximately three times the local free-surface grid size. Singularities are best placed approximately normal to the body and free surface from collocation nodes.

Guidelines for the location of free-surface nodes, body nodes, singularities and points at which the radiation condition is to be satisfied are given in [1].

References

[1] Scullen, D.C. *Accurate computation of nonlinear free-surface flows.* PhD Thesis, The University of Adelaide (1998) `http://www.maths.adelaide.edu.au/Applied/staff/dscullen.html`