METHOD FOR THREE-DIMENSIONAL MODELLING OF A MIXED FLOW PUMP USING PHOENICS

by D. Radosavljevic

Technical Investigation Department
Lloyd's Register

Computer and operating system used:	*Compaq XP-1000 running under NT4*
Date of issue of PHOENICS version used:	*PHOENICS 3.2 of July 1999*

ABSTRACT

The paper describes a technique applied to model the flow through the last stage of a multi-stage mixed-flow pump. A significant amount of cracking was observed on the pump bowl after only a limited time in operation. As part of the overall investigation into the stress levels within the bowl walls the hydrodynamic loads within the bowl needed to be calculated.

The pump consisted of an impeller with six rotor vanes and a bowl with seven stator vanes. Due to geometric complexities of both the impeller and the bowl, in particular, a simplified approach to modelling was adopted and coded within the PHOENICS Ground routine. This approach involved modelling only one impeller passage under steady conditions and saving the exit flow field data into a separate file. These results were then used as input to the simulation of the bowl flow. Again one passage was modelled. However, the bowl flow field was modelled as transient, allowing for one full pass of the impeller passage in front of the bowl inlet plane. The advancement of the impeller passage was modelled through Ground, with the impeller exit flow-field results 'cycling' along the bowl inlet at each time step. The time step was synchronized with the rotational speed of the impeller. Re-calculation of the inlet velocity components was performed at each time step in order to preserve the correct momentum inflow.

The model allowed the transient hydrodynamic pressure field within the bowl passage to be calculated. The hydrodynamic load results were then transferred to a finite element analysis package where they were superimposed onto other components of the total load on the pump.

CONTENTS LIST

INTRODUCTION

The Technical Investigation Department (TID) of Lloyd's Register (LR) was requested to assist in the investigation of cracking and corrosion of a number of in-service pumps related to a major water supply project located in North Africa. The type of pump under investigation was defined as a well pump of the vertical submersible turbine type. A schematic of the pump is shown in **Figure 1.**

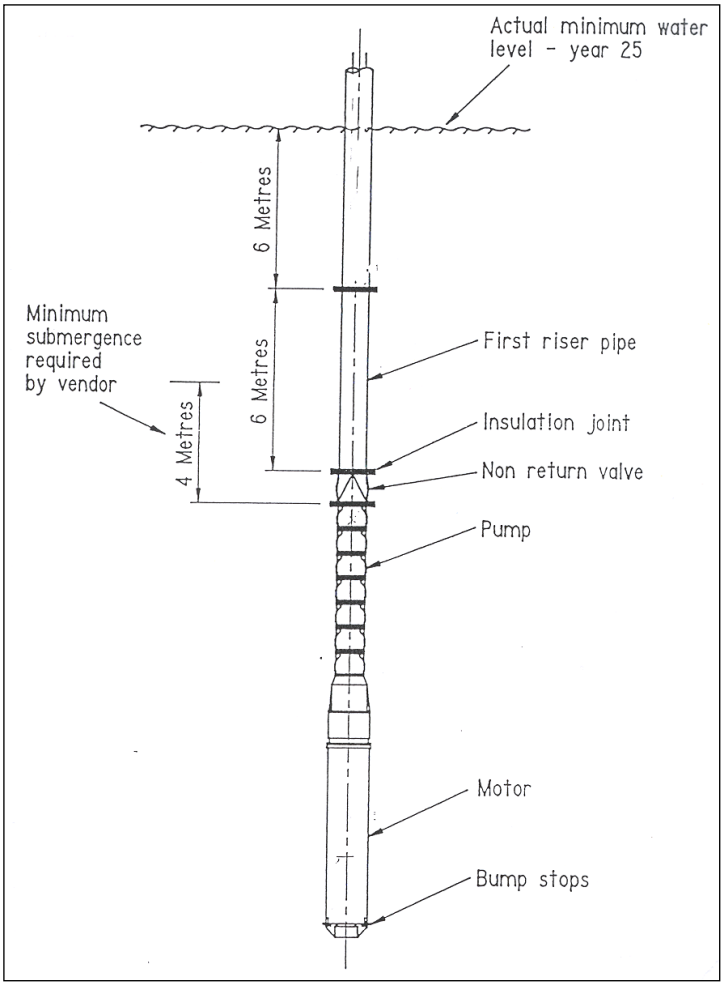

Figure 1 Schematic of pump

The pump was specified to meet a range of duties for 25 years in relation to the envisaged drawdown schedule. During the course of approximately 3 years of operation, pump performance problems were encountered in some wells. Upon withdrawal from the well, the pumps were observed to exhibit severe cracking and corrosion, in particular in the region of the upper pump bowl, as shown typically in **Figure 2.** Cracking has also been observed in a corroded impeller.

Figure 2 Corroded/Cracked Pump Bowl

In view of the perceived significance of hydrodynamically-induced stress levels, TID was requested to carry out certain hydrodynamic and stress analysis studies that were followed by related metallographic studies.

Computational fluid dynamic (CFD) analysis was used to determine hydrodynamic loads which were then input into a finite element analysis package to calculate in-service theoretical stress levels. The hydrodynamic loads were to include the effects of rotation of the impeller relative to the bowl, and related transient processes, but not to include possible water hammer effects.

OBJECTIVE OF WORK

The objectives of the CFD part of the overall work may be summarised as follows:

- Set up a realistic geometric model of one impeller flow passage based on the full-scale measurement data of the actual blades;

- Calculate the flow field through the blade passage of a mixed-flow pump impeller using PHOENICS. Prepare results to be used as input to the bowl model;

- Set up a realistic geometric model of one bowl flow passage based on the full-scale measurement data of the sectioned bowl;

- Calculate the transient flow and pressure field based on the input from the impeller calculation;

- Provide hydrodynamic loading data within impeller and bowl passages for inclusion into the finite element stress analyses.

DESCRIPTION OF PHENOMENON SIMULATED

The flow inside the final stage of a multi-stage mixed-flow pump has been investigated. The stage is composed of an impeller having 6 blades and a diffuser having 7 vanes. At nominal operating condition, the flow through the pump is 45 kg/s with an impeller rotating at 2900 rpm.

The flow in hydraulic machines, such as turbines and pumps, is inherently unsteady due to the relative motion of an impeller blade passing in front of diffuser vanes (**Figure 3**). Furthermore, the flow is fully turbulent, highly three-dimensional and spatially non-uniform. Although the unsteady stator/rotor interaction mechanisms can be found only by calculating the coupled stator-rotor flow solution the computational resources required for the numerical simulation of this phenomenon are large and often exceed available resources. To overcome this problem each stage of a hydraulic machine is often treated separately and steady simulations are performed for each component. Simulations of the impeller associated with a vaneless diffuser are used for designing the impeller form. The diffuser is designed by assuming that the flow entering it is completely mixed (circumferentially uniform inflow).

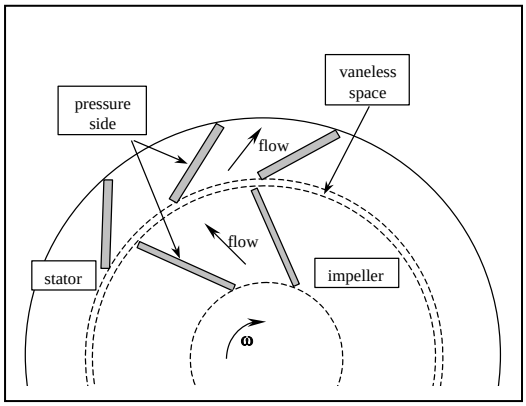

Figure 3 Schematic of pump operation

An additional difficulty associated with multi-stage simulations is that, in general, each stage has a different number of blades. For practical machines, if N_R and N_S are respectively the number of blades of the rotor and of the stator stages, neither N_R/N_S nor N_S/N_R has an integer value. (If, for example, the stator/rotor pitch ratio is 2/1, simulations could be performed on one rotor channel and two stator channels by applying simple periodic boundary conditions on the lateral boundaries.)

As stated in the objectives of this study it was important to estimate whether the stator-rotor interactions could generate unsteady pressure forces that could result in excessive loads and observed mechanical damage to the pump.

The flow leaving an impeller channel is fully three-dimensional and non-uniform between the hub and the shroud as well as in the circumferential direction. Therefore, in view of the complexity of the outflow, it appeared doubtful that separate steady models could be used to construct the flow entering the downstream blade row, making the uncoupled approach insufficiently accurate for detailed analysis.

The classical solution to this problem is to use several grids that move relative to each other. Typically, stationary grids are employed for partitioning the stator flow domain, and grids rotating with the blades are used in the rotor stage. Specialized boundary treatments are then employed at the grid interfaces

to transfer information between them. The sliding grid option in PHOENICS is intended to allow this kind of simulation. However, due to some practical limitations it was not possible to employ this option.

Taking into account various limitations and difficulties associated with different methods it was decided to adopt the 'middle' approach. The impeller and bowl would be modelled separately, with the classical steady approach applied to the impeller. But the bowl would be modelled as transient in order to assess stator/rotor interaction. Accuracy would be enhanced by fully preserving the non-uniformity of the impeller exit flow field and using it as input to the separate rotor model.

PHOENICS SETTINGS

SATELLITE

Impeller

As regards pump construction the geometry of the impeller and the blades was based on a set of drawings generated from sectioning and measuring the actual impeller (**Figure 4**). The resulting geometry of the blades and the flow passage are shown in **Figure 5**. The computational grid used to define the vanes is also visible on the figure.

Figure 4 Sectioning of the Impeller

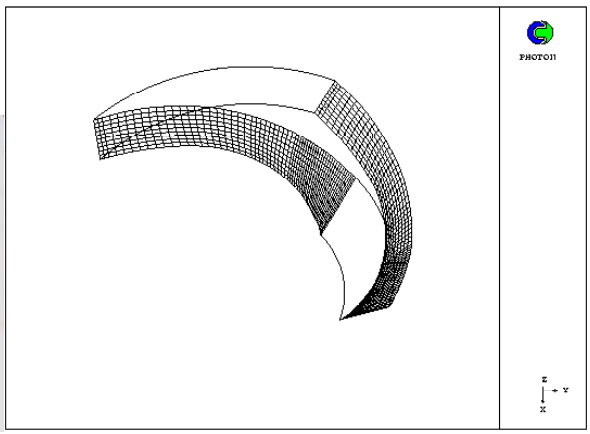

Figure 5 Computational domain

Due to the shape of the blades, a Body Fitted Grid was used to define the computational domain. With 28 cells in the x-direction, 10 cells in the y-direction and 60 cells in the z-direction, the total number of computational cells was 16,800. Only one blade passage was modelled in order to reduce the size of the computational domain. Defined in this way, the computational domain represented 1/6 of the impeller flow volume. Cyclic boundary conditions were set for the exit passage, linking the vaneless space at the exit of the current blade passage with the adjacent ones.

The inlet mass flow was set to 1/6 of 45 kg/s, while the angular speed of the impeller of 2,900 rpm was used to calculate the radial velocity distribution at the blade passage inlet. The incoming fluid was taken to be water, and therefore treated as incompressible, in line with conventional approaches.

A pressure boundary condition at the impeller outlet was specified, in line with the expected pressure increase within the impeller. The hydrostatic part of the pressure was removed from the computations to allow better capture of the dynamic component.

The resulting hydrodynamic model solved the full set of Navier-Stokes equations and included a two-equation k-ε turbulence model. Rotational forces were taken into account by adding an additional term to the momentum equations through the use of a ROTA patch.

Friction along the blades, the shroud and the hub was taken into account by applying the standard 'wall function' boundary condition which specifies zero velocity at the wall surface and introduces special sources for k and ε.

The full listing of the impeller Q1 is given as Appendix 1.

Bowl

The geometry of the bowl passage was similarly based on a set of drawings generated from sectioning an unused bowl (**Figure 6**). The resulting geometry of the blades and the flow passage are shown in **Figure 7**. The computational grid used to define the vanes is also visible on the figure. The outlet boundary from the impeller, which also represents an inlet boundary to the bowl computational domain, is outlined in an orange colour.

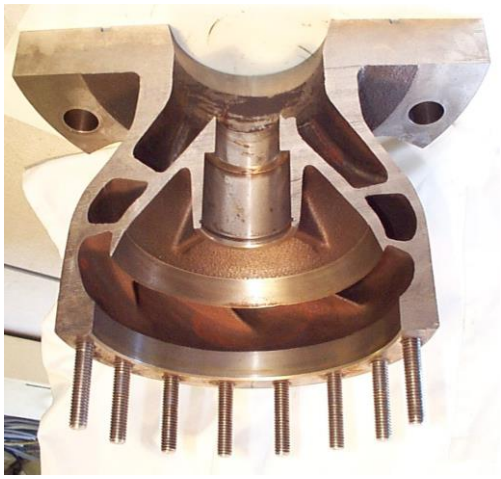

Figure 6 Sectioning of the Bowl

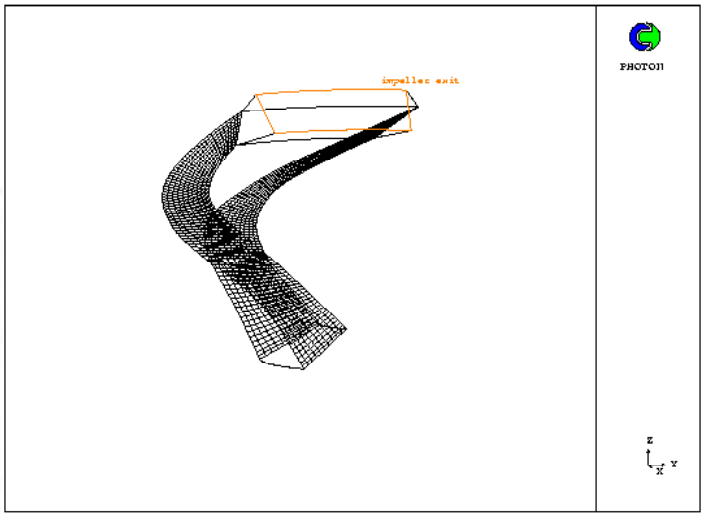

Figure 7 Computational domain

As for the impeller, a Body Fitted Grid was used to define the computational domain. With 24 cells in the x-direction, 10 cells in the y-direction and 92 cells in the z-direction, the total number of computational cells was 22,080. Again only one blade passage was modelled in order to reduce the size of the computational domain. Defined in this way, the computational domain represented 1/7 of the bowl flow volume. Cyclic boundary conditions were set for the inlet and exit passage, linking the current blade passage with the adjacent ones.

The hydrodynamic model used for simulation was identical to the one used for the impeller, with one important difference. During the rotation of the impeller, the inlet flow field to any one bowl passage varies with time. As described earlier this is a highly transient process, so the model for the bowl flow was set as transient. Inlet boundary conditions were then synchronized to vary in accordance with impeller movement. This allowed a full 'cycle' of one impeller flow passage passing in front of one bowl passage to be captured. The GROUND routine was used to define this process, as described below.

Logical variable LG(11) was used as a flag to signal the use of GROUND generated code for reading of impeller flow results and setting of inlet boundary conditions.

The full listing of the bowl Q1 is given as Appendix 2.

GROUND

Impeller calculations did not require GROUND coding except for writing values of velocity components in the exit plane. This was achieved with a logical flag in Q1 that activated coding in Group 19.

The unsteady inlet conditions for the bowl require the specification of appropriate inflow boundary conditions. These settings are made in Section 12 of Group 13 in GROUND. The technique applied was similar to the use of the face sliding block technique. At the time t=0 the impeller and the bowl mesh are aligned (**Figure 8**), starting at IX=1. This means that opposite to the IX=1 cell of the bowl is the IX=1 cell of the impeller, opposite to the IX=2 cell of the bowl is the IX=2 cell of the impeller and so on. Due to the difference in length of the circumferential sections the number of cells in the bowl passage (NX_{st}=24) was different from the number of cells in the impeller passage (NX_{im}=28). However, the numbers were chosen in a way that achieved identical cell size at the interface (7 bowl sections x 24 = 6 impeller sections x 28).

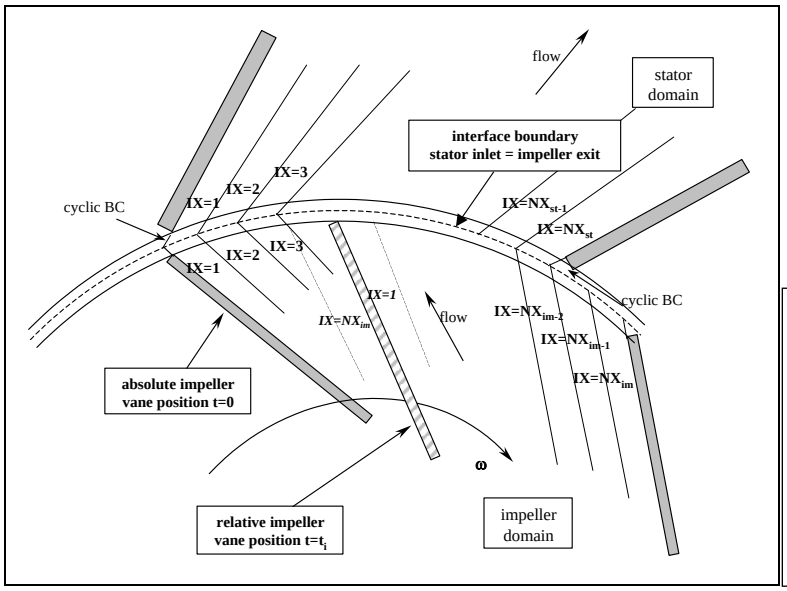

Figure 8 Links at impeller and bowl interface plane

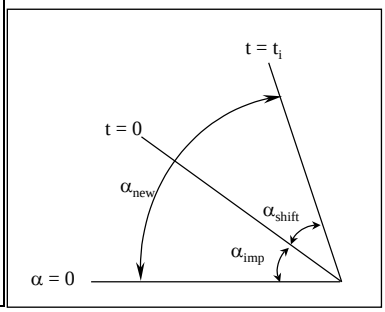

Figure 9 Relative grid shift

At each time step the impeller mesh is shifted by one place and the momentum and mass flux at the block interface are read from the file. The time step size was synchronized with a grid shift of exactly one cell. A total of 28 time steps were needed (=NX_{im}) to cover one full passage of the impeller channel.

The main difficulty that needs to be overcome in this approach results from the fact that as the impeller mesh 'rotates', the Cartesian velocity components that were written into the results file become incorrect. **Figure 9** indicates the problem. At time t=0 each cell centre of the impeller mesh has an angle $\alpha=\alpha_{imp}$. At time $t=t_i$ the same cell should shift to a new position, in line with the rotational speed, to make an angle $\alpha=\alpha_{new}$. The difference represents an angular shift, α_{shift} that has to be calculated for each cell. This angular shift is equal to the angle between two adjacent cells ($\alpha_{slice}= 2\pi / 7/ 24$) times the current time step **i**.

Once each new angle is calculated a correction has to be applied in order to re-calculate the new Cartesian velocity components and ensure their values correspond to the instantaneous vector position in space. As rotation is performed in the x-y plane only, x and y vector components need to be re-calculated.

One final modification to the newly calculated velocity components is required to account for the fact that the bowl flow field is now defined in an absolute frame of reference, while the impeller results are valid in a rotational frame. The circumferential velocity vector is calculated at each point and its components are superimposed on the existing velocity components.

Parts of the existing GXBFC routine were adapted and used to calculate the vector normal to each cell, the corresponding vector components and the mass flow.

The full commented listing of the relevant GROUND coding is given as Appendix 3.

PRESENTATION OF RESULTS

Typical Output Appearance

Some typical results of the simulations are shown in **Figures 10 to 18**.

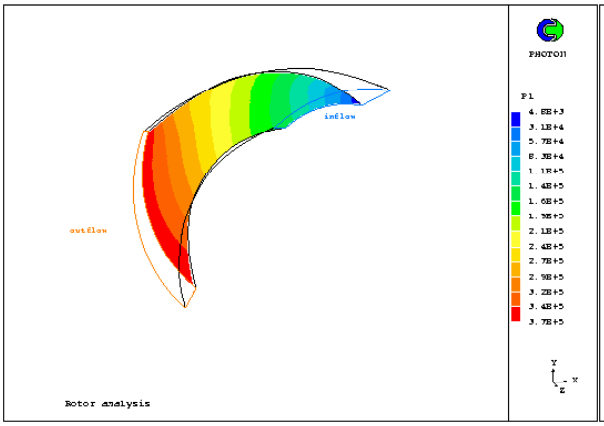

Figure 10 Pressure distribution over the hub surface

Figure 11 Pressure distribution over the shroud surface

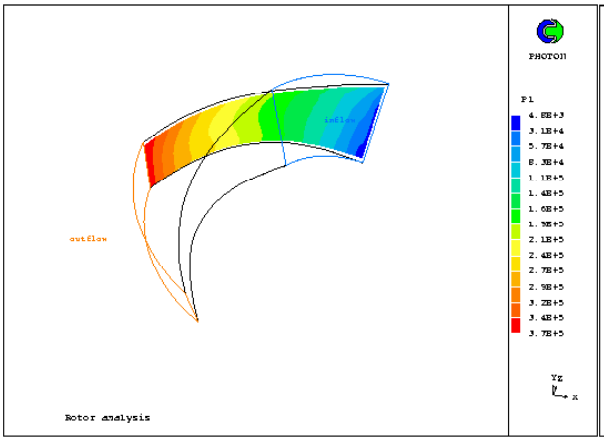

Figure 12 Pressure distribution over the low-pressure vane surface

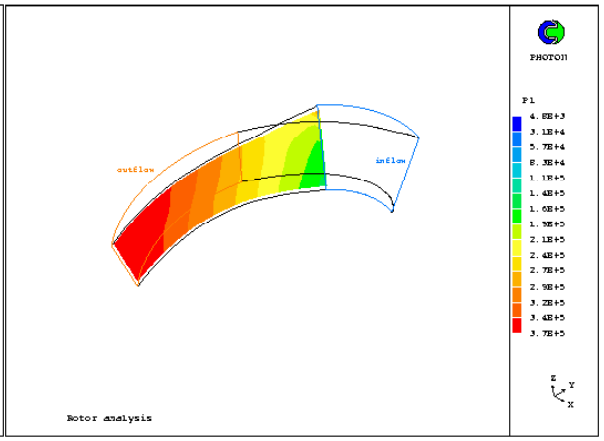

Figure 13 Pressure distribution over the high-pressure vane surface

Figures 10 to 13 show pressure contours on the four 'sides' of the impeller flow passage. **Figures 10** and **11** show the pressure distribution at the bottom and top (hub and shroud) surface within the passage. **Figure 12** shows the pressure contours on the suction side of the vane, while **Figure 13** shows the pressure contours on the pressure side of the vane.

Figures 14 to 17 show a similar set of pressure contours on the four 'sides' of the bowl flow passage, at one particular time step. Although the pressure distribution varies from one time step to another, in

line with the instantaneous position of the impeller vanes, the results shown on these figures can be regarded as typical.

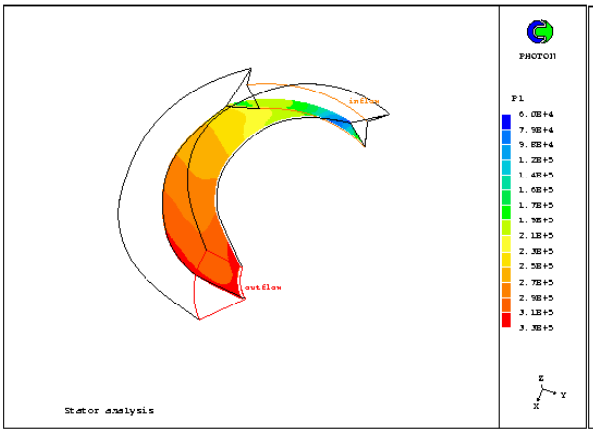

Figure 14 Pressure distribution over the inner surface – bowl flow passage

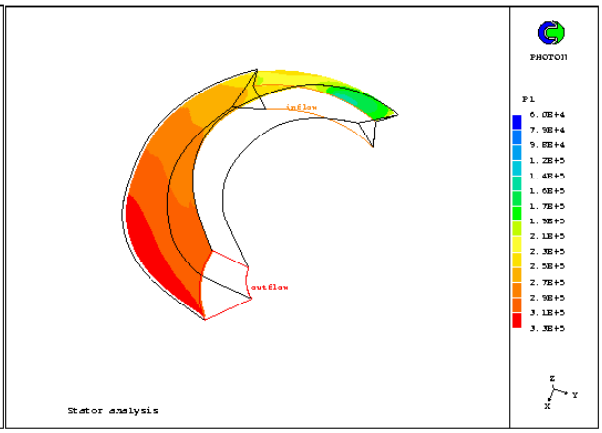

Figure 15 Pressure distribution over the outer surface – bowl flow passage

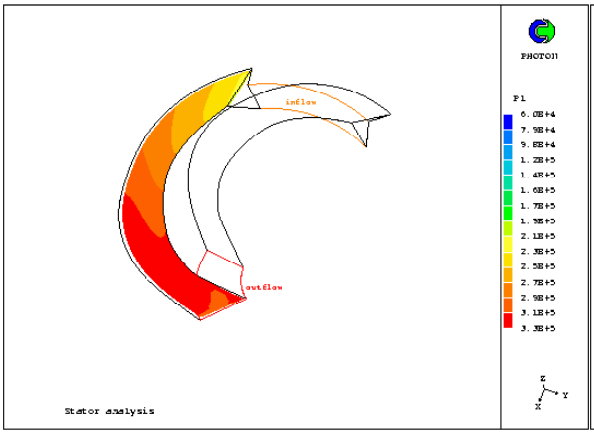

Figure 16 Pressure distribution over the high-pressure vane surface – bowl flow passage

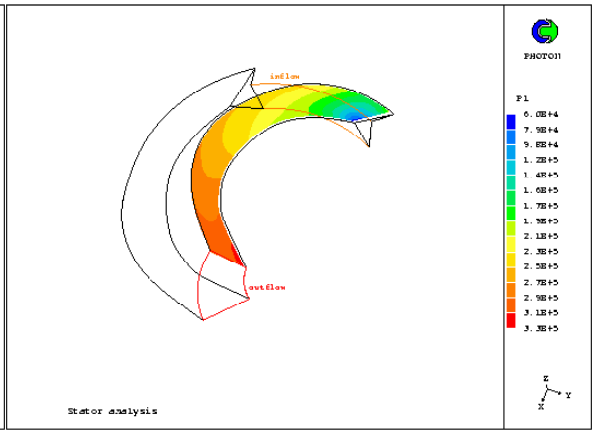

Figure 17 Pressure distribution over the low-pressure vane surface – bowl flow passage

Figure 18 shows the pressure field in the bowl inlet plane at several time steps. Variations in the pressure field show a strong spatial non-uniformity. In addition it is possible to visualize the movement of the impeller blade across the opening by following the boundary between the low and high-pressure zones.

Convergence

The impeller calculations achieved full convergence within 500 sweeps. There were no problems encountered although relaxation had to be applied.

It was found that for the bowl calculations convergence was greatly enhanced by running the case first as steady. Results were obtained for the ‘aligned’ grid position (t=0, see **Figure 8**). These results were then used as a restart option for transient calculations to provide an initial velocity field.

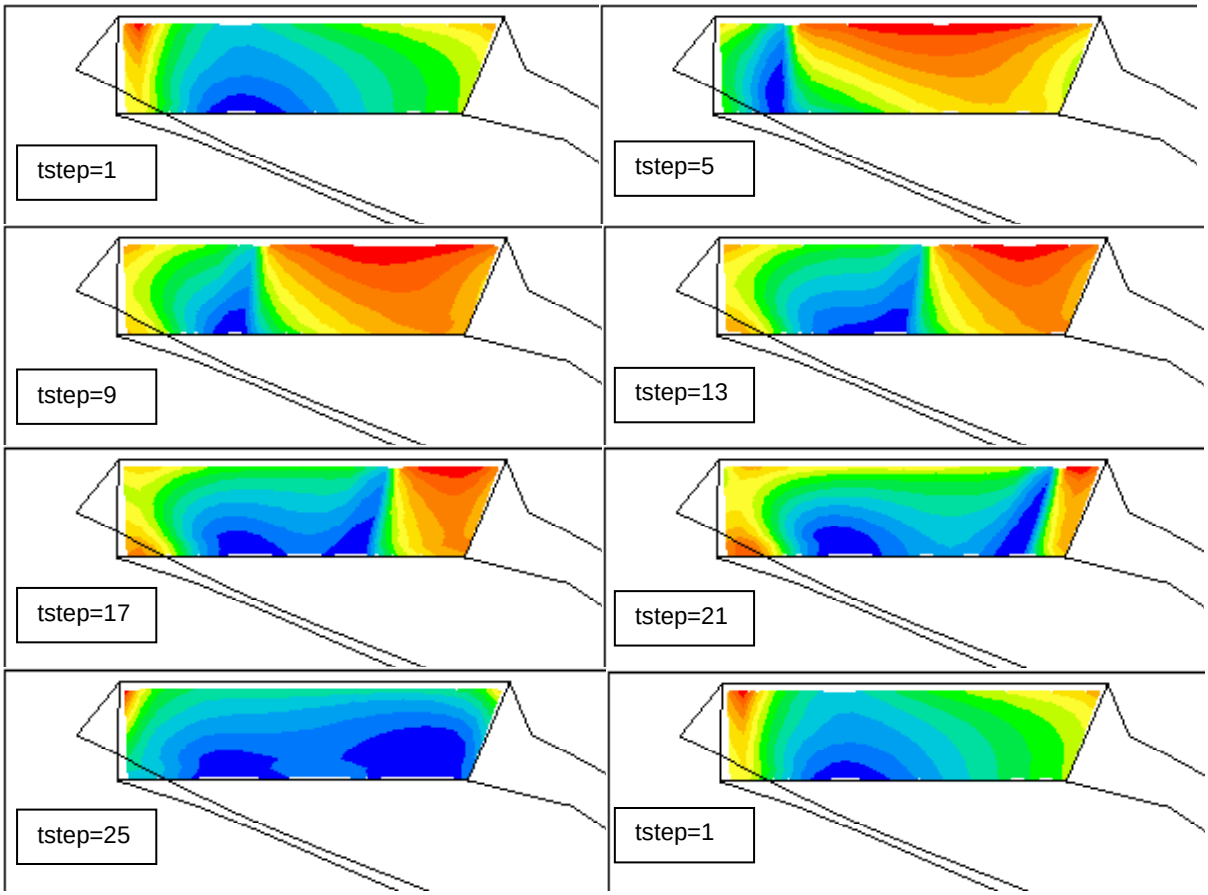

Figure 18 Transient pressure field at the interface plane showing the movement of the impeller blade

Computer storage and time

The General Collocated Velocity (GCV) option in PHOENICS demands increased computer storage and significantly slows down computations compared to the standard BFC option. The size of the F-array required for these calculations was 21 million. The run-time for impeller calculation was typically 20 minutes, while the transient run for the bowl took approximately 10 hours CPU.

DISCUSSION OF RESULTS

Assessment of numerical accuracy

Measurement data were not available to be able to assess the numerical accuracy of this approach. However some comparisons with the known data are possible. The calculated pressure increase within the impeller had an average value of 3.6 bar. The calculations for the impeller were carried out as steady so this pressure increase remains constant. The calculated pressure increase within the bowl varied depending on the relative position of the impeller blades at the inlet to the bowl diffuser. The pressure at the bowl exit did not show large variations and the pressure increase achieved throughout the bowl passage was on average between 2 and 3 bar. The peak pressure value at the inlet was calculated at an instant when there were no impeller blades opposite it. As there are seven stator blades and six impeller blades there is an instant in time when there are no impeller blades in front of the bowl passage inlet. Combined with the pressure increase within the impeller the whole device achieved a pressure increase close to 6 bar, which is in line with the overall pressure increase of 40 bar across 7 stages, or 5.71 bar per stage.

In addition to using the local pressures to calculate the hydrodynamic load acting on the internal surfaces of the impeller, they were used to calculate the torque required to balance the system. Subroutine GXDRAG was used to calculate the overall force generated within the impeller passage, which was then translated into the torque required to power the pump. This corresponded closely to the actual power of the motor used to drive the pump during operation.

Assessment of physical realism

The pressure contours (**Figures 10** to **17**) clearly show an increase in pressure along the passage, as would be expected from engineering considerations of this type of flow situation. These contours represent the pressure relative to the reference hydrostatic value; therefore to obtain the absolute value, the hydrostatic pressure corresponding to the pump operating depth should be added.

The transient change in the pressure field shown in **Figure 18** allows visualization of the impeller blade movement through observation of the high/low pressure boundary as it moves across the opening. This boundary creates a 'shadow' effect of the pressure/suction side of the impeller blade. It is possible to track blade movement from left to right following the increase in time step. It should be noted that these plots were generated with the plane scaling of pressure values in order to provide the sharpest contrast, so they should be primarily observed from the qualitative point of view.

Some assumptions in the applied methodology could have influenced the accuracy of the calculations. The impeller flow field was calculated in isolation, without taking into account interaction with the bowl flow field. The assumption of cyclic conditions in the vaneless space ahead of the bowl inlet is not strictly true due to the difference in size between the bowl inlet and the impeller exit. However, these assumptions were not thought to have influenced either the physical realism of the method or the achieved accuracy within the scope of the study.

CONCLUSIONS

The CFD study has fulfilled its main objective, namely to provide estimates of the hydrodynamic loading within the pump for input into a finite element stress analysis package. Despite non-uniformity in the pressure distribution along the passages, there is generally a smooth transition from the lower to the higher-pressure values, without pronounced local peaks. This implies that the total hydrodynamic load experienced by the pump flow surfaces follows a smooth increase along the flow passages and is unlikely to be the cause of the damage observed on the pump walls. The results also showed qualitative behaviour in keeping with experience and the calculated torque corresponded closely to the power used for pump operation.

Due to certain restrictions that proved critical for this study it was not possible to use the PHOENICS built-in sliding grid option. However the flexibility of the GROUND structure allowed an alternative method to be developed and coded. This approach could be useful even in cases when it is possible to apply the sliding grid option. A reduction in the grid requirement and a much shorter turnaround time might prove useful for the initial stages of the problem set-up.

RECOMMENDATIONS

This study revealed some practical limitations in the available PHOENICS options designed for the simulation of rotating machinery. With the fast advancement in hardware capability it seems that resource limitation for full equipment simulation is disappearing fast. It would be of great use then to be able to easily apply a multi-block, sliding mesh option to simulate complete geometry of a rotating machine. In the meantime, the method described in this paper could be used as an alternative.

LITERATURE REFERENCES

1. J. R. Maguire , *Water Supply Pump Bowl and Impeller Pump Investigation - Summarising report*, 99/TID/5675-3, 1999. (Confidential Report).

2. F.A. Muggli, D. Wiss, K. Eisele, Z. Zhang, M.A. Casey and P. Galpin, *Unsteady flow in the vaned diffuser of a medium specific speed pump*, ASME Paper, 96-GT-157, 1996.

3. Y.P. Marx, *Investigation of stator-rotor interaction phenomena in a centrifugal pump*, In 8th International Symposium on Unsteady Aerodynamics and Aeroelasticity of Turbomachines, Stockholm, 1997.

4. J.P. Gostelow, *Cascade Aerodynamics*, Pergamon Press, 1984.

Appendix I Q1 for rotor analysis

```
TALK=F;RUN( 1, 1)
 ************************************************************
CPVNAM=GENERAL
 ************************************************************
IRUNN    =       1 ;LIBREF =      14
 ************************************************************
  Group 1. Run Title
TEXT(Rotor analysis                              )
 ************************************************************
  Group 2. Transience
STEADY   =    T
 ************************************************************
  Groups 3, 4, 5  Grid Information
    * Overall number of cells, RSET(M,NX,NY,NZ,tolerance)
RSET(M,28,10,60)
    * Set overall domain extent:
    *        xulast  yvlast  zwlast
           name
XSI= 1.000000E+00; YSI= 1.000000E+00; ZSI= 1.000000E+00
RSET(D,CHAM    )
 ************************************************************
  Group 6. Body-Fitted coordinates
BFC=T
READCO(XYZ)
NONORT   =    T
    * X-cyclic boundaries switched
 ************************************************************
  Group 7. Variables: STOREd,SOLVEd,NAMEd
ONEPHS   =    T
   * Non-default variable names
    * Solved variables list
SOLVE(P1  ,U1  ,V1  ,W1  )
    * Stored variables list
STORE(UCRT,VCRT,WCRT,EPKE)
    * Additional solver options
TURMOD(KEMODL)
GCV      =    T
 ************************************************************
  Group 8. Terms & Devices
 ************************************************************
  Group 9. Properties
RHO1     = 9.990000E+02
ENUL     = 1.000000E-06 ;
 ************************************************************
  Group 10.Inter-Phase Transfer Processes
 ************************************************************
  Group 11.Initialise Var/Porosity Fields

ANGVEL   = 3.036870E+02

REAL(VEL,W1IN,V1IN,U1IN)
VEL=3.87
W1IN=-1.*VEL*0.406
V1IN=-1.*VEL*0.913
U1IN=-1*ANGVEL*0.044

FIINIT(P1  ) =  1.000000E+04 ;FIINIT(U1  ) =  U1IN
FIINIT(W1  ) =  U1IN ;FIINIT(V1  ) = 1.
FIINIT(EPKE) =  1.000000E+00
   No PATCHes used for this Group
INIADD   =    F

 ************************************************************
```

```
  Group 12. Convection and diffusion adjustments
   No PATCHes used for this Group
 ************************************************************
  Group 13. Boundary & Special Sources
     ** Radii of hub and shroud at inlet...
REAL(RHUB,RSHRO,HDY)
RHUB=0.0348;RSHRO=0.05368;HDY=(RSHRO-RHUB)/(2*NY)

DO II=1,NY
 INLET (BFC:II:,LOW ,1,NX,II,II,1,1,#1,#1)
 VALUE (BFC:II:,P1  , RHO1*VEL)
 VALUE (BFC:II:,U1  , GRND1       )
 VALUE (BFC:II:,V1  , GRND1       )
 VALUE (BFC:II:,W1  , GRND1       )
 VALUE (BFC:II:,WCRT,W1IN)
 VALUE (BFC:II:,VCRT,V1IN)
 VALUE (BFC:II:,UCRT,-1.*ANGVEL*(RHUB+(II*2-1)*HDY))
    ** Based on 5% intensity
 VALUE (BFC:II:,KE  , 3.75E-02)
 VALUE (BFC:II:,EP  , 1.85E-02)
ENDDO

PATCH (OUTLET  ,HIGH,1,NX,1,NY,NZ,NZ,#1,#1)
COVAL (OUTLET  ,P1  , 1.000000E+00, 3.600000E+05)

    ** Source for rotational forces **
PATCH (ROTA    ,PHASEM,1,NX,1,NY,1,NZ,#1,#1)
COVAL (ROTA    ,U1  , FIXFLU      , GRND1       )
COVAL (ROTA    ,V1  , FIXFLU      , GRND1       )
COVAL (ROTA    ,W1  , FIXFLU      , GRND1       )

  ** Friction on channel walls
PATCH (WFUN1   ,NWALL ,1,NX,NY,NY,1,NZ-2,#1,#1)
COVAL (WFUN1   ,U1  , GRND2       , 0.000000E+00)
COVAL (WFUN1   ,W1  , GRND2       , 0.000000E+00)
COVAL (WFUN1   ,KE  , GRND2       , GRND2       )
COVAL (WFUN1   ,EP  , GRND2       , GRND2       )

PATCH (WFUN2   ,SWALL ,1,NX,1,1,1,NZ-2,#1,#1)
COVAL (WFUN2   ,U1  , GRND2       , 0.000000E+00)
COVAL (WFUN2   ,W1  , GRND2       , 0.000000E+00)
COVAL (WFUN2   ,KE  , GRND2       , GRND2       )
COVAL (WFUN2   ,EP  , GRND2       , GRND2       )

PATCH (WFUN3   ,WWALL ,1,1,1,NY,1,NZ-2,#1,#1)
COVAL (WFUN3   ,V1  , GRND2       , 0.000000E+00)
COVAL (WFUN3   ,W1  , GRND2       , 0.000000E+00)
COVAL (WFUN3   ,KE  , GRND2       , GRND2       )
COVAL (WFUN3   ,EP  , GRND2       , GRND2       )

PATCH (WFUN4   ,EWALL ,nx,nx,1,ny,1,nz-2,#1,#1)
COVAL (WFUN4   ,V1  , GRND2       , 0.000000E+00)
COVAL (WFUN4   ,W1  , GRND2       , 0.000000E+00)
COVAL (WFUN4   ,KE  , GRND2       , GRND2       )
COVAL (WFUN4   ,EP  , GRND2       , GRND2       )

BFCA    = 9.990000E+02

  ** Cyclic conditions at the exit
XCYIZ(NZ-1,NZ,T)
 ************************************************************
  Group 14. Downstream Pressure For PARAB
 ************************************************************
  Group 15. Terminate Sweeps
 LSWEEP  =   600
```

```
 RESFAC  = 1.000000E-03
 ************************************************************
  Group 16. Terminate Iterations
 ************************************************************
  Group 17. Relaxation
 RELAX(U1  ,FALSDT, 1.000000E-04)
 RELAX(V1  ,FALSDT, 1.000000E-04)
 RELAX(W1  ,FALSDT, 1.000000E-04)
 RELAX(KE  ,LINRLX, 5.000000E-01)
 RELAX(EP  ,LINRLX, 5.000000E-01)
 KELIN   =       3
 ************************************************************
  Group 18. Limits
 VARMAX(U1  ) = 1.000000E+06 ;VARMIN(U1  ) =-1.000000E+06
 VARMAX(V1  ) = 1.000000E+06 ;VARMIN(V1  ) =-1.000000E+06
 VARMAX(W1  ) = 1.000000E+06 ;VARMIN(W1  ) =-1.000000E+06
 ************************************************************
  Group 19. EARTH Calls To GROUND Station
 GENK    =    T
   ** Axis of rotation
 ROTAZB  = 1.000000E+00
 ************************************************************
  Group 20. Preliminary Printout
 ECHO    =    T
 ************************************************************
  Group 21. Print-out of Variables
 ************************************************************
  Group 22. Monitor Print-Out
 IXMON   =       3 ;IYMON  =       5 ;IZMON  =       6
 NPRMNT  =       1
 TSTSWP  =      -1
 ************************************************************
  Group 23.Field Print-Out & Plot Control
   No PATCHes used for this Group
 ************************************************************
  Group 24. Dumps For Restarts
 ********
   *** Write exit data for input to bowl (LG(10)=T)
LG(10)=T
STOP
```

Appendix II Q1 for bowl analysis

```
TALK=F;RUN( 1, 1)
  ************************************************************
  Group 1. Run Title
 TEXT(Stator analysis                          )
 ************************************************************
  Group 2. Transience
STEADY  = F
  ** Time step corresponds to the shift of one cell
  ** No. of time steps equal NX-impeller
RSET(U,0.0,3.44827e-3,28)
  ************************************************************
  Groups 3, 4, 5  Grid Information
    * Overall number of cells, RSET(M,NX,NY,NZ,tolerance)
 RSET(M,24,10,92)
  ************************************************************
  Group 6. Body-Fitted coordinates
BFC=T
READCO(XYZ)
NONORT  =    T
    * X-cyclic boundaries switched
 ************************************************************
   Group 7. Variables: STOREd,SOLVEd,NAMEd
ONEPHS  =    T
SOLVE(P1  ,U1  ,V1  ,W1  )
     * Stored variables list
STORE(UCRT,VCRT,WCRT,EPKE)
     * Additional solver options
SOLUTN(P1  ,Y,Y,Y,N,N,N)
TURMOD(KEMODL)
GCV     =    T
  ************************************************************
   Group 8. Terms & Devices
  ************************************************************
   Group 9. Properties
RHO1    = 9.990000E+02
ENUL    = 1.000000E-06 ;
  ************************************************************
   Group 10.Inter-Phase Transfer Processes
  ************************************************************
   Group 11.Initialise Var/Porosity Fields

RESTRT(ALL)
   No PATCHes used for this Group

  ************************************************************
   Group 12. Convection and diffusion adjustments
    No PATCHes used for this Group
  ************************************************************
   Group 13. Boundary & Special Sources

     *** Set inflow through Ground - based on impeller exit
 INLET (BFCIN,LOW ,1,nx,1,ny,1,1,1,lstep)
 VALUE (BFCIN,P1  , grnd)
 VALUE (BFCIN,U1  , GRND1       )
 VALUE (BFCIN,V1  , GRND1       )
 VALUE (BFCIN,W1  , GRND1       )
 VALUE (BFCIN,WCRT,grnd)
 VALUE (BFCIN,VCRT,grnd)
 VALUE (BFCIN,UCRT,grnd)
 VALUE (BFCIN,KE  , grnd)
 VALUE (BFCIN,EP  , grnd)

 PATCH (OUTLET  ,HIGH,1,nx,1,ny,nz,nz,1,lstep)
```

```
 COVAL (OUTLET  ,P1  , 1.000000E+00, 3.00000E+05)
 COVAL(OUTLET,U1,0.,0.)
 COVAL(OUTLET,V1,0.,0.)
 COVAL(OUTLET,W1,0.,0.)
 COVAL (OUTLET  ,KE  , 0.000000E+00, SAME        )
 COVAL (OUTLET  ,EP  , 0.000000E+00, SAME        )

  ** Friction along bowl channel walls
 PATCH (WFUN1   ,NWALL ,1,nx,ny,ny,1,nz-2,1,lstep)
 COVAL (WFUN1   ,U1  , GRND2       , 0.000000E+00)
 COVAL (WFUN1   ,W1  , GRND2       , 0.000000E+00)
 COVAL (WFUN1   ,KE  , GRND2       , GRND2       )
 COVAL (WFUN1   ,EP  , GRND2       , GRND2       )

 PATCH (WFUN2   ,SWALL ,1,nx,1,1,1,nz-2,1,lstep)
 COVAL (WFUN2   ,U1  , GRND2       , 0.000000E+00)
 COVAL (WFUN2   ,W1  , GRND2       , 0.000000E+00)
 COVAL (WFUN2   ,KE  , GRND2       , GRND2       )
 COVAL (WFUN2   ,EP  , GRND2       , GRND2       )

 PATCH (WFUN3   ,WWALL ,1,1,1,ny,11,nz-2,1,lstep)
 COVAL (WFUN3   ,V1  , GRND2       , 0.000000E+00)
 COVAL (WFUN3   ,W1  , GRND2       , 0.000000E+00)
 COVAL (WFUN3   ,KE  , GRND2       , GRND2       )
 COVAL (WFUN3   ,EP  , GRND2       , GRND2       )

 PATCH (WFUN4   ,EWALL ,nx,nx,1,ny,11,nz-2,1,lstep)
 COVAL (WFUN4   ,V1  , GRND2       , 0.000000E+00)
 COVAL (WFUN4   ,W1  , GRND2       , 0.000000E+00)
 COVAL (WFUN4   ,KE  , GRND2       , GRND2       )
 COVAL (WFUN4   ,EP  , GRND2       , GRND2       )


BFCA    = 9.990000E+02
   *** Cyclic conditions at the inlet and exit to the channel
XCYIZ(1,10,T)
XCYIZ(nz-1,nz,T)
  ************************************************************
   Group 14. Downstream Pressure For PARAB
  ************************************************************
   Group 15. Terminate Sweeps
LSWEEP  = 500
RESFAC  = 1.000000E-03
  ************************************************************
   Group 16. Terminate Iterations
  ************************************************************
   Group 17. Relaxation
RELAX(KE  ,LINRLX, 5.000000E-01)
RELAX(EP  ,LINRLX, 5.000000E-01)
KELIN   =       3
  ************************************************************
   Group 18. Limits
VARMAX(U1  ) = 1.000000E+06 ;VARMIN(U1  ) =-1.000000E+06
VARMAX(V1  ) = 1.000000E+06 ;VARMIN(V1  ) =-1.000000E+06
VARMAX(W1  ) = 1.000000E+06 ;VARMIN(W1  ) =-1.000000E+06
  ************************************************************
   Group 19. EARTH Calls To GROUND Station
GENK    =    T

  ************************************************************
   Group 20. Preliminary Printout
ECHO    =    T
  ************************************************************
   Group 21. Print-out of Variables
  ************************************************************
```

```
   Group 22. Monitor Print-Out
IXMON   =       3 ;IYMON  =       5 ;IZMON  =       6
NPRMNT  =       1
TSTSWP  =      -1
  ************************************************************
   Group 23.Field Print-Out & Plot Control
    No PATCHes used for this Group
  ************************************************************
   Group 24. Dumps For Restarts
  *** Read impeller exit plane data as input to bowl (LG(11)=T)
LG(11)=T
IDISPA=1
CSG1='S'
STOP
```

Appendix III GROUND coding for pump analysis

```
C.... FILE NAME GROUND.FTN--------------------------------230597
```

NOTE: Only Groups that contain relevant coding are shown!

```
C
C 2   User dimensions own arrays here, for example:
C     DIMENSION GUH(10,10),GUC(10,10),GUX(10,10),GUZ(10)
      DIMENSION EV(3),A(3),B(3),CC(3),D(3),IJKA(3),IJKB(3),IJKC(3),
     1          IJKD(3)
      SAVE IJKA,IJKB,IJKC,IJKD
C
C 3   User places his data statements here, for example:
C*************************************************************
C
C--- GROUP 1. Run title and other preliminaries
C
    1 GO TO (1001,1002,1003),ISC
C
 1001 CONTINUE
C
C   * -----------GROUP 1  SECTION  3 ----------------------------
C---- Use this group to create storage via MAKE, GXMAKE etc which it is
C     essential to dump to PHI (or PHIDA) for restarts
C     User may here change message transmitted to the VDU screen
      IF(.NOT.NULLPR.AND.IDVCGR.EQ.0)
     1  CALL WRYT40('GROUND file is GROUND.F   of:    230597 ')
C
C … Initialize arrays for vector normal components (modified from GXBFC)
C
      CALL SUB4(IJKA(2),0,IJKB(2),0,IJKC(2),1,IJKD(2),1)
      CALL SUB4(IJKA(3),0,IJKB(3),1,IJKC(3),1,IJKD(3),0)
C
      RETURN
C*************************************************************
C
C--- GROUP 13. Boundary conditions and special sources
C                                       Index for Coefficient - CO
C                                       Index for Value       - VAL
   13 CONTINUE
      GO TO (130,131,132,133,134,135,136,137,138,139,1310,
     11311,1312,1313,1314,1315,1316,1317,1318,1319,1320,1321),ISC
1311 CONTINUE
C------------------- SECTION 12 ------------------- value = GRND
c
      IF(IZ.EQ.1.AND.LG(11)) THEN
C
C... Angle for each x-slice for one bowl flow channel (2Pi/no of blades/NX)
C
             ALPHASLICE=8.9760E-01/NX
C
C... Set NX for impeller grid
             INX=28
C
C... Rotational speed
             OMEGA=3.036870E+02
C
      CALL SUB2(L0W1,L0F(LBNAME('WC1')),L0V1,L0F(LBNAME('VC1')))
      CALL SUB2(L0KE,L0F(KE),L0EP,L0F(EP))
      CALL SUB2(L0U1,L0F(LBNAME('UC1')),L0VAL,L0F(VAL))
      L0P1=L0F(P1)
```

```
C
C... Initial calculations - adopted from GXBFC
C
                      MIT = MOD(INTTYP+1,2)
                CALL SUB4(IJKA(1),MIT,IJKB(1),MIT,IJKC(1),MIT,IJKD(1),
     1                     MIT)
                FLIO = FLOAT(2*MIT-1)
                ITD2 = INTTYP/2
                CALL SUB3(MITX,MOD(4-ITD2,3)+1,MITY,MOD(5-ITD2,3)+1,
     1                     MITZ,MOD(6-ITD2,3)+1)
       IZ1 = IZSTEP
c
c ... Open file for reading input data
c
        OPEN(UNIT=10, FILE='interface_data', STATUS='UNKNOWN', ERR=1981)

C...Swap input values for each time step advancing 1 cell/time step

  107 READ(10,170)idum1,idum2,uc1in,vc1in,wc1in,rkein,repin
c
c...Find IX position corresponding to the current time step
c
              IF(IDUM1.NE.((INX+1)-ISTEP)) GOTO 107
c
      DO 1235 IX=1,NX

C
c...Find relative position of the current shifted impeller cell against
C... its position at impeller exit
C
       ALPHASHIFT=ALPHASLICE*(ISTEP-INX)
c
         DO 1235 IY=1,NY
            ICELL=IY+(IX-1)*NY
c
c...Calculate inlet velocity vector
C
       VELLOC=SQRT(VC1IN*VC1IN+UC1IN*UC1IN)
C
C... Calcualte absolute vector position
C
       ALPHAIMP=ACOS(UC1IN/VELLOC)

C … Calculate new cartesian components to take rotational shift into account
C
       ALPHANEW=ALPHAIMP+ALPHASHIFT
       UC1NEW=VELLOC*COS(ALPHANEW)
       VC1NEW=VELLOC*SIN(ALPHANEW)
C
C...Velocity location in the cell centre - middle of the diagonal
C
      CALL GETPT(IX+IJKA(MITX),IY+IJKA(MITY),
     1                         IZ1+IJKA(MITZ),A1,A2,A3)
       CALL GETPT(IX+IJKC(MITX),IY+IJKC(MITY),
     1                         IZ1+IJKC(MITZ),CC1,CC2,CC3)
       XDIA=0.5*(A1+CC1)
       YDIA=0.5*(A2+CC2)
c
c...find radius
c
       VELRAD=SQRT(XDIA*XDIA+YDIA*YDIA)
c
c...find angle
c
       RADANGLE=ATAN(YDIA/XDIA)
```

```
c
c...find U and V components for rotational frame
c
       ROTU=VELRAD*OMEGA*SIN(RADANGLE)
       ROTV=VELRAD*OMEGA*COS(RADANGLE)
c
c...velocities in absolute frame
c
       UC1ABFR=UC1NEW+ROTU
       VC1ABFR=VC1NEW-ROTV
c
c ... Set values for velocities
c
       if(indvar.eq.LBNAME('UC1')) f(l0val+icell)=UC1ABFR
       if(indvar.eq.LBNAME('VC1')) f(l0val+icell)=VC1ABFR
       if(indvar.eq.LBNAME('WC1')) f(l0val+icell)=WC1IN
c...
c..   Set values for k-e
c...
       if(indvar.eq.EP)f(l0val+icell)=REPIN
       if(indvar.eq.KE)f(l0val+icell)=RKEIN
c
c... Mass inflow
c... Calculate vector
c
       IF(INDVAR.EQ.P1) THEN
C
C...Unit normals (adopted from GXBFC)
C... Get remaining two corners of the cell
C
       CALL GETPT(IX+IJKB(MITX),IY+IJKB(MITY),
     1                          IZ1+IJKB(MITZ),B1,B2,B3)
       CALL GETPT(IX+IJKD(MITX),IY+IJKD(MITY),
     1                          IZ1+IJKD(MITZ),D1,D2,D3)
                    CALL SUB4R(A(1),A1,B(1),B1,CC(1),CC1,D(1),D1)
                    CALL SUB4R(A(2),A2,B(2),B2,CC(2),CC2,D(2),D2)
                    CALL SUB4R(A(3),A3,B(3),B3,CC(3),CC3,D(3),D3)
C
C... Construct the unit vector normal to the cell face, store in EV
                    CALL NORML(A,B,CC,D,EV)
C... Mass flow is density * (vector velocity).(normal unit vector)
         F(L0VAL+ICELL)= FLIO*RHO1* (UC1ABFR*EV(1)+VC1ABFR*EV(2)+
     1                          WC1IN*EV(3))
       ENDIF
C
       IF(.NOT.EOF(10)) THEN
         READ(10,170)idum1,idum2,uc1in,vc1in,wc1in,rkein,repin
       ELSE
         REWIND 10
         READ(10,170)idum1,idum2,uc1in,vc1in,wc1in,rkein,repin
         INX=0
       ENDIF
1235   CONTINUE
c
       CLOSE(10)
c
       ENDIF
C
      RETURN
 1312 CONTINUE
C------------------- SECTION 13 ------------------- value = GRND1
      RETURN
C--- GROUP 19. Special calls to GROUND from EARTH
C
   19 GO TO (191,192,193,194,195,196,197,198,199,1910,1911),ISC
```

```
196 CONTINUE
C   * ------------------- SECTION 6 ---- Finish of iz slab.
       IF(ISWEEP.EQ.LSWEEP-1.AND.IZ.EQ.NZ.AND.LG(10)) THEN
             CALL SUB2(L0U1,L0F(LBNAME('UCRT')),L0V1,L0F(LBNAME('VCRT')))
             L0W1=L0F(LBNAME('WCRT'))
             CALL SUB2(L0KE,L0F(KE),L0EP,L0F(EP))

c
c ... Open file for writing exit data
c
        OPEN(UNIT=10, FILE='interface_data', STATUS='UNKNOWN', ERR=1981)
c
c ... Write titles
c
C        WRITE(10,*)'IX   IY   UCRT   VCRT   WCRT   KE   EP'
c ... Write data
c
      DO 1234 IX=1,NX
         DO 1234 IY=1,NY
            ICELL=IY+(IX-1)*NY
C
      WRITE(10,170)IX,IY,F(L0U1+ICELL),F(L0V1+ICELL),F(L0W1+ICELL),
     +     F(L0KE+ICELL), F(L0EP+ICELL)
1234   CONTINUE
c ... Close file
c
        CLOSE(10)
      ENDIF

      RETURN
  170  FORMAT(2I5,5(1P,E12.4))
 1981 write(*,*)'Problem with opening file for writing'
  197 CONTINUE
C   * ------------------- SECTION 7 ---- Finish of sweep.
      RETURN
  198 CONTINUE
C   * ------------------- SECTION 8 ---- Finish of time step.
       RETURN
C*************************************************************
```